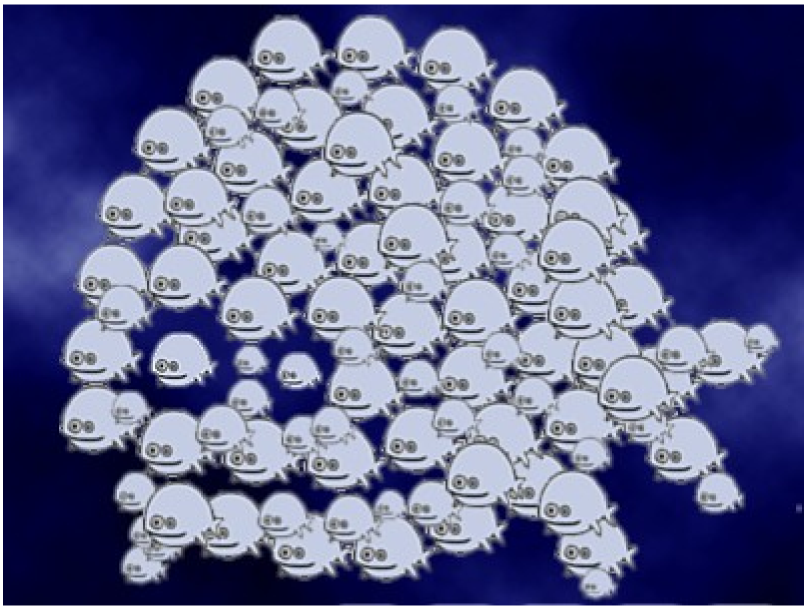


Networking FreeDOS

Contents

Introduction

Ulrich Hansen

In this document I describe how to network a computer running DOS. The system used for installing and testing was FreeDOS. Most things are supposed to work with other versions of DOS as well.

The first chapter is about the history of DOS networking, which mostly happened in the years 1981 to 1995. In the second chapter you find an overview over the drivers, protocols and applications that are still around in the 21st century.

The third chapter explains how to add TCP/IP capabilities to a DOS machine. It is a detailed guide through the installation and configuration of drivers and other required software.

It took me some time to get a clue about DOS networking. This guide shall make it easier for others.

Motivation

I began DOS networking because I wanted to connect my old laptop to the internet. I also wanted to transfer data over the network to my other machines with Windows XP and Debian GNU/Linux.

I would have had alternatives:

- I could have installed another operating system that is able to run on old hardware and easier to network (f.i. GNU/Linux, Windows for Workgroups 3.11 or Windows 9x).

- I could have used floppies to exchange data with my other computers or connected them by a parallel cable (f.i. with MS DOS 6.22 Interlnk, Open/DR DOS Filelink, Laplink, Winlink or FileMaven).

So I admit it: I just did it out of curiosity, for fun and to learn how things work.

System

All solutions described in this document were tested on a Compaq Contura Aero 486SX33 laptop with 20 MB RAM.

Compaq Contura Aero 4/33C

I tried these PCMCIA **network interface cards (NIC)**:

- The 3Com Megahertz 3C574 10/100 LAN PC Card,
- the 3Com EtherLink III 3C589 10Mbit Card and
- the D-Link DFE-670TXD Fast Ethernet PC Card.

The PCMCIA cards are configured by SystemSofts CardSoft Software, which is available for the specified laptop as Compaq softpack **SP1045.EXE**.

It also can be downloaded from driverguide as "SystemSoft Version 3.1, **CARDSOFT.ZIP**".

The **network** is a typical 100Base-T Ethernet LAN that is connected to the internet through a DSL router.

The FreeDOS version is 1.0.

Remarks

I have only been able to try, install and document the programs mentioned here, because many other people were willing to share information and code on the internet. Thanks!

Some parts of this document were inspired by Michael Bernardi's "DOS Networking HOWTO". The links there and also his collections of FAQs and applications were of great value.

Although I tried my best to check the facts, I may be wrong sometimes. Please don't hesitate to correct me. Please also forgive (or correct) my errors in spelling and grammar - my mother tongue is German, not English.

Disclaimer: I wrote this document as a hobbyist and just for the fun of understanding how things work. I am not related to any of the manufacturers, developers or companies mentioned. This text reflects my own experiences with DOS networking. I can not take responsibility for others. My solutions and suggestions may or may not work in your environment.

So please be careful: Installing software to your system or changing system files can damage your machine, disturb your workflow and waste your time.

Copyright

The screenshot of "LAN Manager 2.1" in chapter LAN Manager was taken by Michal Necasek for his "History of OS/2". He gave me his friendly permission to use this screenshot and to publish it under the GNU Free Documentation License.

The foto of the software package "Workgroup Add-On for MS-DOS" in the same chapter has been made by Dirk Makowski for his "Winhistory", a huge collection of items and screenshots of historic software. He also allowed to use his foto as preview picture in this document and publish it under the GNU FDL.

Thanks! The rest of this document, text and pictures, was created by myself.

1. History of DOS networking

This first chapter provides information about the history of networking DOS computers. You don't need to read it, if you just want to install a few drivers and applications. But it may help to avoid confusion.

Read more about the history of TCP/IP networking on DOS computers in Steven Baker's article "Net Worth - Desktop TCP/IP At Middle Age".

Why MS DOS came without network

The IBM Personal Computer (PC) was introduced on August 12th, 1981 as an answer to the Apple II, that challenged IBM's market with office software like VisiCalc and WordStar. The operating system (OS) for the PC had to be done in an extremely short period of time, after Digital Research (DR) had blown the opportunity to license their "Control Program for Micros" (CP/M) to IBM.

The young software company Microsoft, originally only supposed to provide language interpreters and compilers as BASIC and Fortran, stepped in. Microsoft agreed to come up with an OS prototype, similar to CP/M, in just three months. They bought and adapted Seattle Computer's "Quick and Dirty OS" (QDOS) to meet IBM's expectations. Version 1.0 of the "Microsoft Disk Operating System" (MS DOS, also sold as "PC DOS" by IBM) had 4.000 lines of code. Networking was none of its capabilities.

Networking DOS wasn't Microsoft's top priority. Other tasks were far more important: The first DOS did not even know how to handle hard disks. Microsoft was also busy working on the applications that should become the corner stones of its success: "Multiplan"/"Excel" and "Word".

Networking DOS wasn't even planned. This feature was postponed to it's successor OS/2, which was announced by Microsoft and IBM in 1985 (and released in 1987). In the meantime Microsoft provided XENIX, a UNIX variant which was available for the IBM PC since 1983.

This left a time frame for others. There were two approaches for DOS networking, which can be called the "PC centric" and the "UNIX centric" approach.

- The "PC centric" approach means a way to network a group of PCs in order to share files and resources (f.i. printers).
- The "UNIX centric" approach means to develop DOS versions of the standard UNIX network programs and to use them to access or to provide UNIX-like services in LAN or internet (which, in the end, came down to file and resource sharing as well).

"PC centric" approach

Novell NetWare

Novell NetWare

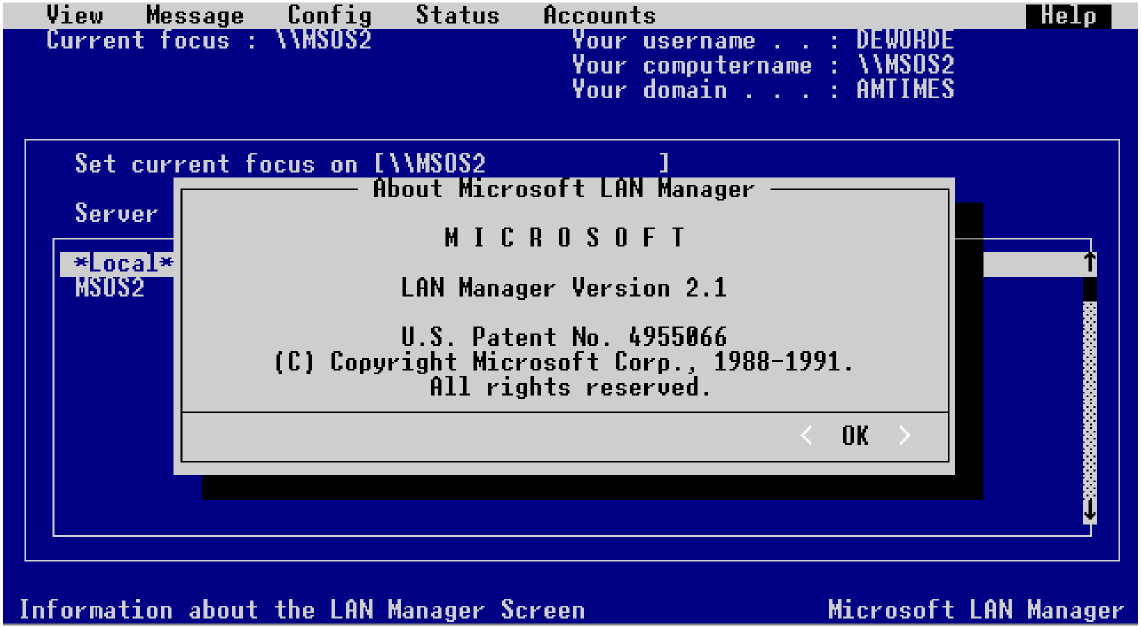
In 1985 Novell, a former hardware manufacture of CP/M systems, released its product "NetWare 86" (v 1.5) for the PC. A version for the AT followed with "NetWare 286" (v 2.0) in 1986.

Novell networked computers according to the client/server model: Clients running MS DOS and some memory resident (TSR) Novell software were able to "log in" to a server that ran "Novell NetWare Server". Once connected they could "map" a volume on the server to a driveletter in DOS and then use it like a local drive. NetWare also enabled the clients to use printers connected to the server.

Client and server communicated over Novell's "Internet Packet Exchange / Sequenced Packet Exchange" (IPX/SPX) protocol. The driver architecture was called "Open Datalink Interface" (ODI).

NetWare established a dominant position in the market until the middle of the nineties.

LAN Manager



```
 View   Message   Config   Status   Accounts                          Help
 Current focus : \\MSOS2               Your username . . : DEWORDE
                                       Your computername : \\MSOS2
                                       Your domain . . . : AMTIMES

    Set current focus on [\\MSOS2              ]
                      About Microsoft LAN Manager
    Server
                          M I C R O S O F T
   *Local*
    MSOS2                LAN Manager Version 2.1

                         U.S. Patent No. 4955066
                 (C) Copyright Microsoft Corp., 1988-1991.
                           All rights reserved.

                                                      < OK >

 Information about the LAN Manager Screen             Microsoft LAN Manager
```

© Screenshot by Michal Necasek, 2001. You may use it under the terms of the GNU Free Documentation License.
Source: http://pages.prodigy.net/michaln/history/os2ee/

Microsoft tried to break Novell's dominance by its own networking software for DOS. The "Microsoft LAN Manager" 1.0 was released in 1987.

Microsoft's networking software came with an own network driver model called "Network Driver Interface Specification" (NDIS). Just like Novell NetWare Microsoft's LAN Manager implemented a **client/server** architecture:

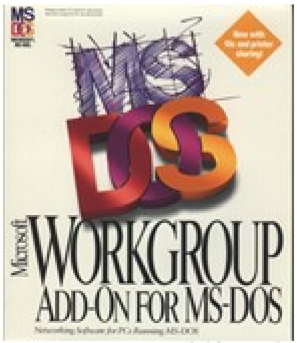
- The server ran OS/2 as operating system and LAN Manager as server software.

- The client ran DOS, OS/2 or Macintosh as OS and used a client software like "LAN Manager client" or "MS Client" for the connection to the server.

With the "Server Message Block" (SMB) protocol merged into LAN Manager since 1991, Microsoft networks became able to do **peer-to-peer networking** as well. This was used by Windows for Workgroups 3.11, released in 1993. The mechanism is known to Windows users as "Windows share", "network neighbourhood" or "Workgroup".

In October 1993 Microsoft released a tool called "Microsoft Workgroup Add-On for DOS" which allowed to have the same peer-to-peer networking possibilities in DOS. While the tool itself is not sold any longer, most of its functionally is still available by usage of "MS Client" and a special update called "`WG1049`" (see MS Client section).

As a solution to network DOS PCs LAN Manager couldn't compete against Novell NetWare. Beneath the disadvantage of being second in the market, this may also have been caused by the fact that you had to install the OS/2 operating system on the server. OS/2 was unpopular among customers. Since 1993 "Microsoft Windows NT Advanced Server (v3.1)" was used instead on the server side.



winhistory.de

DOS Workgroup Add-On

In August 1995 the introduction (and positive reception) of "Windows 95" put an end to DOS networking activities. This Windows version had an own TCP/IP stack already built-in and came with SMB peer-to-peer networking - so many third-party solutions were unnecessary.

"UNIX centric" approach

Experiments to get TCP/IP running on the PC began shortly after IBM introduced it. The first people who worked on it were Dave Clark, Jerry Saltzer and freshman student John Romkey at the Lab for Computer Science of the Massachusetts Institute of Technology (MIT). In 1981 they started a research project with the purpose "to see if TCP/IP could run on something as small as an IBM PC", as Romkey recalls.

PC-IP

The outcome was named "PC-IP", a small TCP/IP implementation that was linked into a few applications like finger, whois and netwatch. PC-IP was released with its source into the public domain. Drew Perkins from Carnegie Mellon University (CMU) and Dan Lanciani from Harvard University improved the code further. PC-IP still can be found in the web.

PC/TCP

Icons of PC/TCP applications

The success of PC-IP inspired Romkey and friends to take the CMU version of the code and develop it into a proprietary product: "PC/TCP", which they sold through their company "FTP Software, Inc.", founded in 1986.

The name was chosen after the popular 'File Transfer Protocol': PC/TCP included one of the first applications for FTP services on the PC.

Packet Drivers

Along with PC/TCP, FTP Software Inc. developed the "Packet Driver Specification", the first multiprotocol driver specification for PC network interface cards. On 12 Dec 1988 this specification was released as an open standard, for anyone to implement.

Packet drivers proved useful to hardware vendors and software developers alike. Most of the public-domain or shareware TCP/IP applications have been written to interface with packet drivers only.

Many packet drivers were written or managed by Russ Nelson at Clarkson University, who became known as the "Packet Driver King" - a story, he describes at his website. Nelson still distributes these drivers as free software through the site of his company "Crynwr" (which was named after the welsh word for "Quaker", his religious belief).

One cause for the success of FTP Software's PC/TCP was this open packet driver interface, which made it easy to develop drivers and applications. When other driver interfaces like ODI and NDIS appeared on the scene, PC/TCP included converters: With a tool called **ODIPKT.COM** ODI drivers became usable as packet drivers, a tool called **DIS_PKT.GUP** did the same for NDIS drivers.

PC/TCP also came with an external TCP/IP kernel called **ETHDRV.EXE** that allowed other programmers to call network functions within their applications without programming them themselves. All these features and an application suite that allowed DOS computers to access or provide TCP/IP services such as news, e-mail, ftp, telnet or network storage (PC/TCP even included a NFS client called InterDrive) made FTP Software Inc. market leader for DOS TCP/IP software.

PC/TCP was installed on over 10 million DOS machines worldwide. In the middle of the nineties FTP Software Inc. had more than 700 employees (see here for more info).

Software by universities and hobbyists

PC-IP and its successor PC/TCP may have been the first or most successful TCP/IP kernels for DOS but they were not the only ones.

KA9Q

Second after the early PC-IP kernel was the ""KA9Q Network Operating System" (NOS)" by Phil Karn in 1985. Karn, an engineer from Baltimore, Maryland had created it for CP/M two years before and then ported it to DOS. It also was one of the first TCP/IP applications for DOS that used FTP Software's packet driver specification.

KA9Q simultaneously acted as an Internet client, a server and an IP packet router. The program was specialized for amateur packet radio, but also could be used f.i. as ftp- or webserver. KA9Q attracted many contributors. As Karn describes: "It was the Linux of its day".

In 2002 KA9Q became free software (GNU GPL). Its descendants, JNOS and EZ-NOS (see also here and here) are still actively maintained.

NCSA Telnet

Based on the Packet Driver Specification several universities wrote TCP/IP applications and utilities for DOS PCs. In 1986 the "National Center for Supercomputing Applications" (NCSA) at the University of lllinois released "Telnet" - a client for UNIX telnet services which could act as FTP and rcp server as well. It was accompanied by TCP/IP tools like finger, whois and lpr.

NCSA Telnet came with its own TCP/IP kernel, which was already built-in into the applications. The software was developed until 1995 (v2.308) and released with source into the public domain.

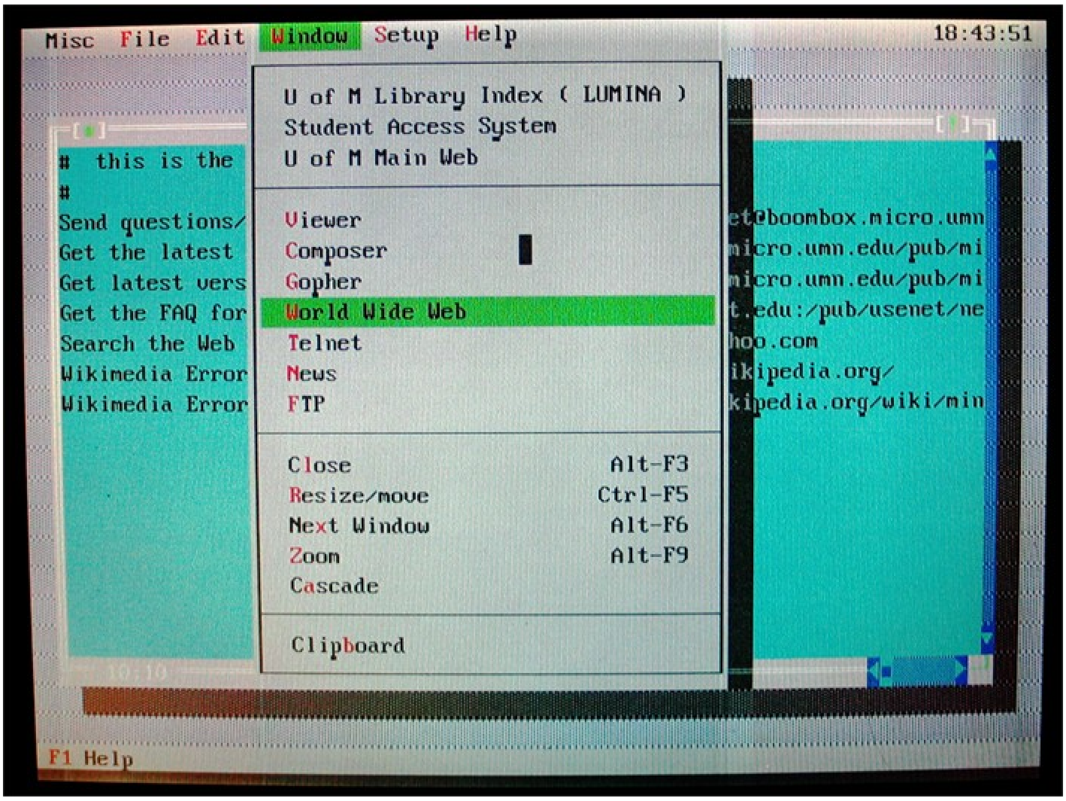
CUTCP

The Clarkson University in Potsdam, New York took version 2.2 of NCSA Telnet and modified it. They called their version CUTE or CUTCP. Eventually Rutgers University, New Jersey took over the maintenance and published the latest release in July 1993. In difference to NCSA Telnet, CUTCP supported the IBM-3270 emulation and was technically improved. It also became public domain.

Others

Other developments at universities that should be mentioned are "Kermit", a terminal emulation for DOS written at the Columbia University, New York and the University of Minnesota's "Minnesota InterNet Users Essential Tool" ("Minuet"), a packet driver based suite of TCP/IP applications.

Minuet provided a mouse driven graphical interface that integrated client applications as E-mail, Gopher, telnet, Usenet News, a web browser and FTP.



Minuet is a good example for the unfortunate design of DOS software licenses in this era. In difference to the GNU General Public License (GPL) that became a standard in the GNU/Linux world, licenses for DOS software written at Universities and by hobbyists were mostly homebrewed. The Minuet license f. i. allowed free usage for the University of Minnesota faculty, staff and students. All others were expected to pay 50$ after 15 days evaluation. Distribution was allowed for non-commercial reasons only.

While these shareware licenses may have worked fifteen years ago, today they are more like a death certificate for the software: The software is not sold or officially distributed any longer. While you may still find someone to pay the shareware fee of 50$ to, it is unlikely he will give you any support or updates. So the cost may exceed what the software is worth today, after its market disappeared and the original developers lost interest. More important: Without source it is of no use for other programmers - so development is put to an end by the license.

And to disallow any "commercial distribution" has the effect, that the places, where f.i. Minuet can be found, are mostly changing, unreliable and random.

Thanks to some developers, essential networking software for DOS with their corresponding sourcecode are in the public domain or Free Software. This enables users and developers alike to still write and use TCP/IP applications for DOS. These important pieces of software include:

- Free versions of packet driver converters for today's more common ODI and NDIS drivers: `ODIPKT.COM` written by Dan Lanciani (Harvard University) allows to use ODI drivers as packet drivers, `DIS_PKT9.DOS` by Prof. Joe R. Doupnik (Utah State University) and Dan Lanciani does the same with NDIS drivers.
- "WatTCP", a popular free TCP/IP kernel library written 1990 - 1992 by Erick Engelke from Canadian Waterloo University and its successor Watt-32 by Gisle Vanem (Norway). These libraries enable programmers to include TCP/IP functions into their applications without having to reinvent the wheel.

- "Trumpet TCP driver (NTCPDRV)" from 1992, a free external TCP/IP kernel created by Australian programmer Peter Tattam (also well known for his Trumpet Winsock).

We will learn more about this software a bit further in this document.

Proprietary software

Beneath FTP Software Inc.'s PC/TCP a lot of other commercial applications could be found on the market for DOS TCP/IP networking software ten or fifteen years ago.

Up to 30 TCP/IP packages for DOS were available in the middle of the nineties, including IBM "TCP/IP for DOS", NetManage "Chameleon", Sun "PC-NFS", Novell "LAN WorkPlace", Frontier Technologies "SuperTCP" or Artisoft "Lantastic".

A comparison of their features can be found here. Read also the corresponding article in the "Network Computing" magazine.

The end of DOS networking

We have to keep in mind, that all this development of TCP/IP software for DOS happened in a relatively short period of time. The IBM PC didn't have any networking capabilities when it was released in 1981. Whoever worked with UNIX machines at this time regarded the PC as toy. But networking became a serious option, when the stronger PC AT (i286) came out in 1984 and in 1986, when Compaq released the first PC with an i386 processor.

This window of opportunity began to close with the introduction of Windows NT in 1993 and was shut by Windows 95 in August 1995, which had an own TCP/IP kernel already built-in and included the Point-to-Point Protocol (PPP), a dialer and SMB peer-to-peer networking. This was a blow for commercial networking software, including Novell NetWare.

FTP Software Inc. was bought out 1998 by former competitor NetManage for a price that was only slightly higher than the amount of money in FTP software's bank account (see here for more info). Today it's gone, of course.

In 2007 the market of DOS networking software is abandoned. Most commercial solutions are not available any more. Their manufacturers changed owners or markets or simply ceased to exist. And whoever worked at the "UNIX centric" approach 10 years ago may today perhaps be found working on GNU/Linux solutions.

Picking up the pieces

Networking DOS in the 21st century means mostly to take what's left of the era of DOS networking from the middle of the eighties until the middle of the nineties. Some Novell and Microsoft tools can still be legally downloaded and used. And TCP/IP solutions by universities and hobbyists that were released into the public domain, as shareware and even as postcardware, are still around, although most of it cannot be found any more at their original locations.

But it is more than that. A few projects are still alive: The graphical DOS browser Arachne is actively developed again by a group of programmers after it was released under the GNU GPL. Also ssh2dos, Watt-32 and the KA9Q successors EZ-NOS2 and JNOS2 are still maintained and developed. Even new DOS TCP/IP software has been written recently, although there seems to be some reluctance to release the source and give these creations a future as free software projects.

2. DOS networking today

In this chapter we will learn more about the network drivers, protocols and applications that can be used with a DOS PC in the 21st century.

Drivers

Ethernet is today's dominant network hardware technology. For this type of network adapters generally three sorts of drivers can be used under DOS (and you should be able to find at least one of them for your card):

- **Packet drivers**, invented by FTP Software Inc.,
- Open Datalink Interface (**ODI**) drivers, developed by Novell and Apple and the
- Network Driver Interface Specification (**NDIS**) drivers, developed by Microsoft and 3Com.

All three are *multiprotocol* network drivers, what means that they are able to support multiple protocols over the same card. Earlier drivers did support only a single protocol. Multiprotocol drivers communicate directly with the network interface card and provide a published interface specification, to which applications can be written.

For a good general introduction into this topic see "Implementing Multi-Protocol Network Drivers in a DOS Environment" by the University of Georgia, Athens, Georgia (U.S.).

Protocols

The following protocols are supported by these three drivers:

- Transmission Control Protocol / Internet Protocol (**TCP/IP**) used for instance by UNIX, GNU/Linux, Windows Vista, OS X and the Internet,
- Internet Packet Exchange / Sequenced Packet Exchange (**IPX/SPX**) protocols, used for instance by Novell NetWare,
- Network Basic Input Output System (NetBIOS) Extended User Interface **(NetBEUI**) protocol used for instance by OS/2, Windows 9x, ME and 2000.

TCP/IP is the standard for basic internet services as http, smtp or ftp and it also became the default protocol for connecting Local Area Networks (LAN).

NetBEUI was the default protocol for LANs in Microsoft systems until Windows 2000. It was replaced by NetBIOS over TCP/IP (NBT) and then by TCP/IP. The application level network protocol SMB for instance runs directly atop of TCP since Windows XP.

Novell's IPX was used in Novell NetWare, which has been the default networking solution for personal computers running DOS or Windows 3.x. Since 1998 NetWare is able to run on TCP/IP, more recent versions use it per default.

Conclusion: TCP/IP is the "lingua franca" of modern networks. It is still possible to use other protocols, but support may come to an end.

Applications

In general we can distinguish the following three ways of networking that still can be used with DOS.

- There are still ODI driver based **Novell NetWare** programs around for accessing a NetWare network and using TCP/IP services.
- NDIS driver based **Microsoft "LAN Manager Client" or "MS Client 3.0"** are supposed to integrate DOS machines into a Windows / SMB workgroup.
- Packet driver based "UNIX-like" **TCP/IP applications** can be used to access or provide network services like http, smtp, ftp, ssh or NFS.

We will learn more about all of them in the following chapters.

While the above categories should cover most programs, there are a few exceptions. One of them is "Invisible LAN", a NetWare-like DOS application that even comes with an own protocol called "TransBIOS".

Another special type of applications are network bootdisks, which are able to work with a broad range of hardware and use multiple protocols. Bootdisks are mostly used for backup, restore and repair. They can be extremely helpful.

Examples are:

- Bart's Network Boot Disk
- Drivesnapshot
- Netbootdisk

The legal situation is unfortunate: Some packages seem to include files from MS Client, while Microsoft's license does not allow such distribution.

Novell "NetWare DOS Client" and "TCP/IP Client"

Novell, former market leader for local networks, hasn't forgotten it's DOS users. The company still provides software for DOS, that can be downloaded and used for free:

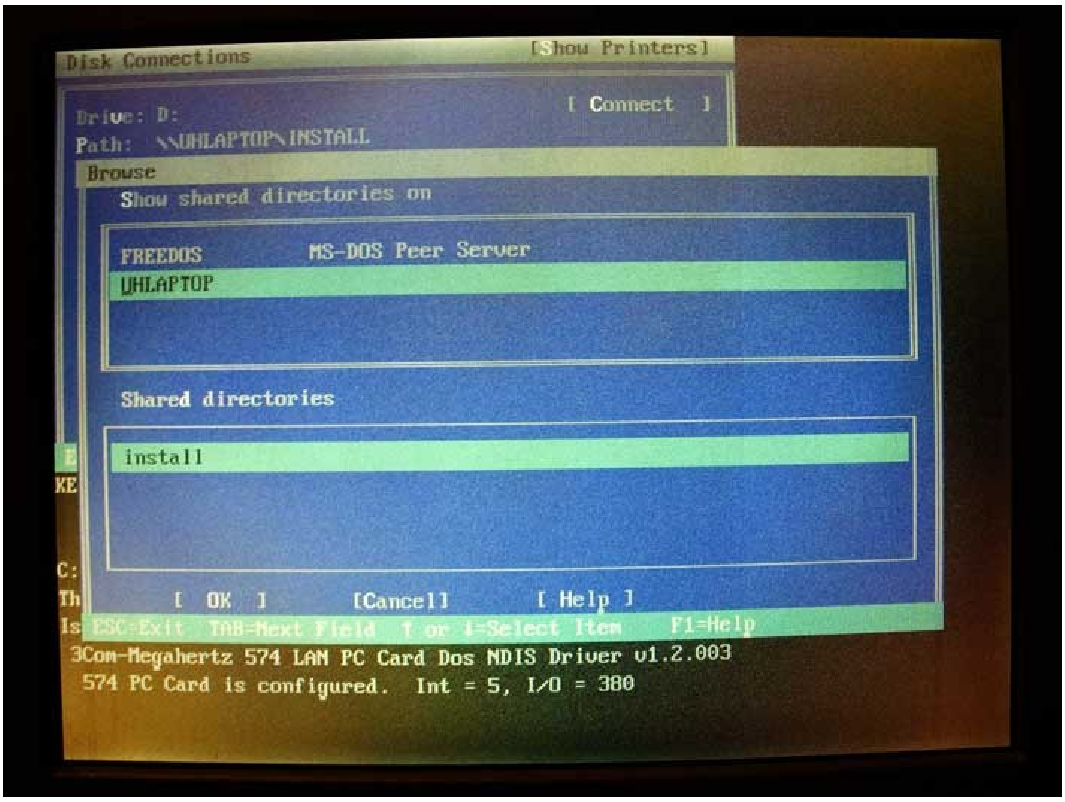
- The NetWare ODI Client 16 Driver for MS-DOS (1996) and the NetWare Client 32-Driver (1998) allow DOS users to connect to a Novell NetWare Server.
- Novell's TCP/IP kernel "**`TCP16.EXE`**" (1996) can still be used to run TCP/IP applications directly over ODI (see chapter "Other TCP/IP Kernel").

On the sites above Novell includes a warning that this software is neither maintained nor supported any longer.

Useful information about Novell's DOS clients can still be found at the newsgroup "novell.support.os.client.dos-win3x" which was active until 2004.

More recent information and useful links about using NetWare with DOS can be found at "DOS Solutions".

Microsoft "LAN Manager Client" and "MS Client 3.0"



```
Disk Connections
Drive: D:
Path: \\UHLAPTOP\INSTALL
Browse
Show shared directories on
FREEDOS              MS-DOS Peer Server
UHLAPTOP
Shared directories
install
3Com-Megahertz 574 LAN PC Card Dos NDIS Driver v1.2.003
574 PC Card is configured.  Int = 5, I/O = 380
```

Microsoft's programs "LAN Manager" and "MS Client 3.0" are still popular among DOS users. They can be downloaded from Microsoft's ftp-site and are free for "internal use", but not for distribution.

- MS LAN Manager Client (1998) can be found at: ftp://ftp.microsoft.com/bussys/Clients/LANMAN/
- MS Client 3.0 (1995) can be found at: ftp://ftp.microsoft.com/bussys/Clients/MSCLIENT/

There are numerous webpages about the installation and usage especially of MS Client 3.0, so I won't reinvent the wheel here.

- An excellent review, installation report and comparison of both Microsoft products has been undertaken by Dutch programmer Jacco de Leeuw.
- Ken from British website "DOS Solutions" gives a detailed overview over many resources related to MS Client .
- U.S. programmer Jonathan Young has created a nice looking MS Client installation walkthrough with many screenshots.
- Installation guides in German language have been written by Gerhard Zeiser, by Gerd Röthig, by Michael Vogl and by Stefan Mayrhofer.

The popularity of the thirteen years old MS Client 3.0 may be caused by the fact that it is possible (but not officially supported) to use an update called "`WG1049.EXE`" to add server functionality to the client. In effect this adds features comparable to those of the "Workgroup Add-On for DOS", which is not longer sold by Microsoft.

This "hack" seems to give people hope to be able to integrate their DOS machine into a modern Windows XP or GNU/Linux SMB Workgroup. There may be limits, though.

Steven Baker critically remarks, that, while the core TCP/IP protocol remained stable over the years, Microsoft's SMB protocols changed from upgrade to upgrade and exist "in a dozen different dialects". So compatibility is an issue. Other problems can be caused by the authentication methods that are used with Windows or Samba.

So in the end you may be better off by using TCP/IP applications, than by endlessly trying to get SMB to work.

Apart from that it can be considered a good thing that Microsoft still allows to legally download and use this software. Some files from the "MS Client" package are also essential for the use of TCP/IP applications over today's common NDIS drivers.

TCP/IP applications

We heard in the "History of DOS Networking" above how popular TCP/IP applications for DOS were in the eighties and nineties of the last century. This era ended when Microsoft introduced Windows 95: It provided TCP/IP and other networking functions under an easy, user friendly interface.

With their customers many commercial network applications for DOS vanished. Public domain software, freeware and shareware programs were preserved by the upcoming internet. Some are still developed today. Even new software has been written.

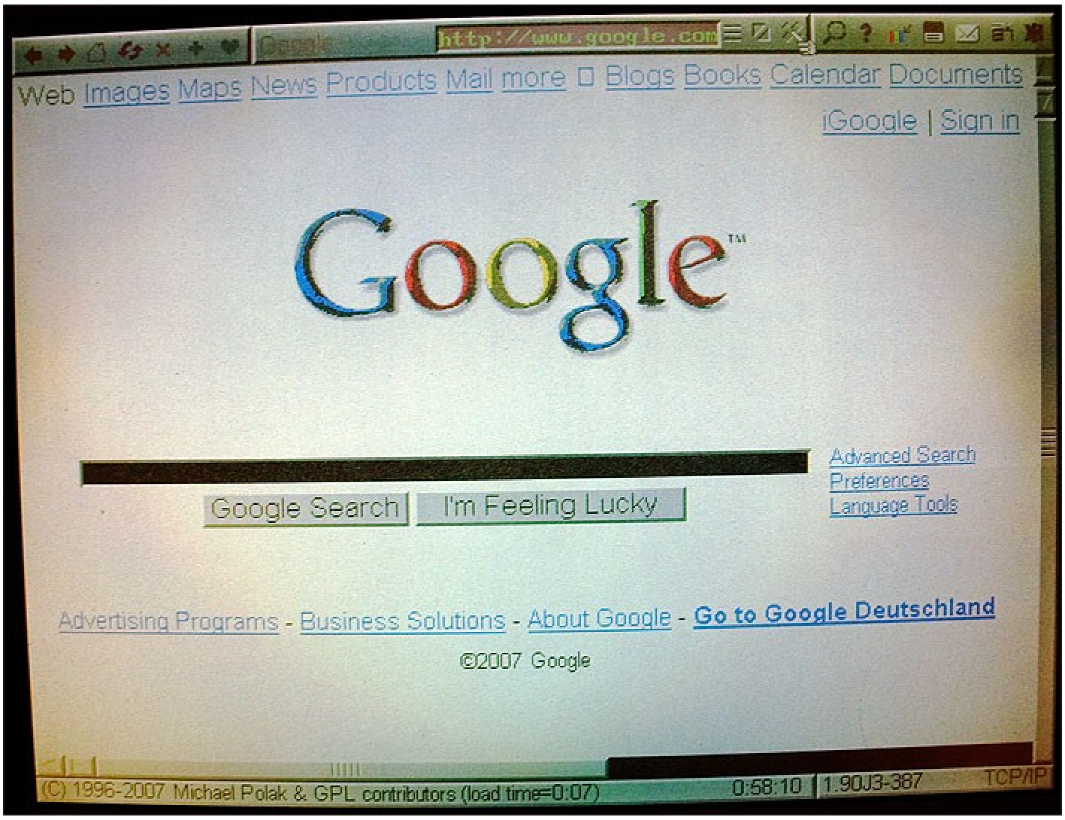
DOS TCP/IP applications include classic networking tools as ping, lpr or finger as well as SSH and NFS clients, FTP servers, webservers, mail and news software or web browsers. There is even a complete peer-to-peer networking solution called "Network Enhanced Operating System" (NeOS) which was developed 1995-2000 as part of European Union's "EUREKA" project. Now it is postcardware.

Michael Bernardi has collected a list of more than hundred TCP/IP applications available for DOS. There also other lists here, here and here, which may contain additional information.

A recent and constantly updated resource for DOS networking applications is the British "DOS Solutions". Links to other DOS resources are collected in FreeDOS technote 157. I also found the nostalgia site oldskool.org pretty helpful.

A lot of information can still retrieved from usenet newsgroup comp.protocols.tcp-ip.ibmpc, especially from their FAQ, which is posted in parts one, two and three.

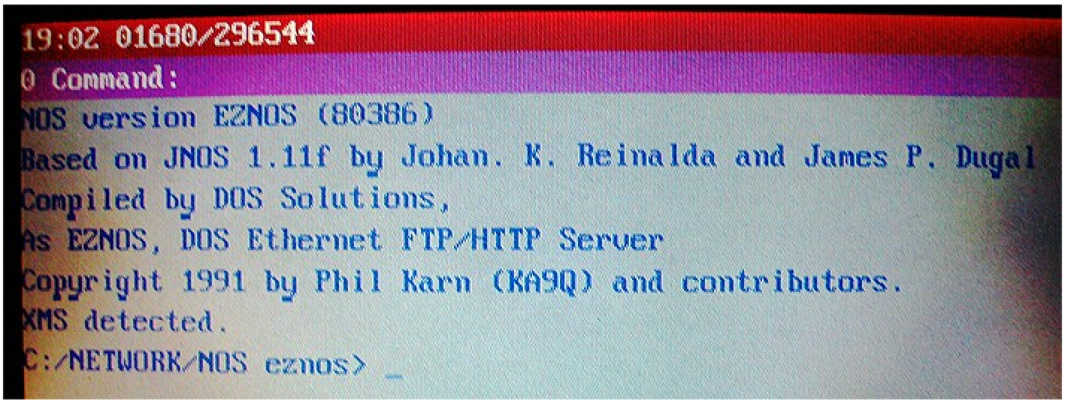
It is not possible to cover all TCP/IP applications in this document. But here are some remarkable examples:


Google
http://www.google.com
Web Images Maps News Products Mail more Blogs Books Calendar Documents
iGoogle | Sign in
Google™
Google Search
I'm Feeling Lucky
Advanced Search
Preferences
Language Tools
Advertising Programs - Business Solutions - About Google - Go to Google Deutschland
©2007 Google
(C) 1996-2007 Michael Polak & GPL contributors (load time=0:07)
0:58:10
1.90J3-387
TCP/IP

"**Arachne**" is a graphical web browser for DOS. It was created in 1996 by Czech programmer Michael Polak and his company xChaos / Arachne Labs. In 2002 Michael Polak decided to make Arachne free software. The new license is the GNU GPL.

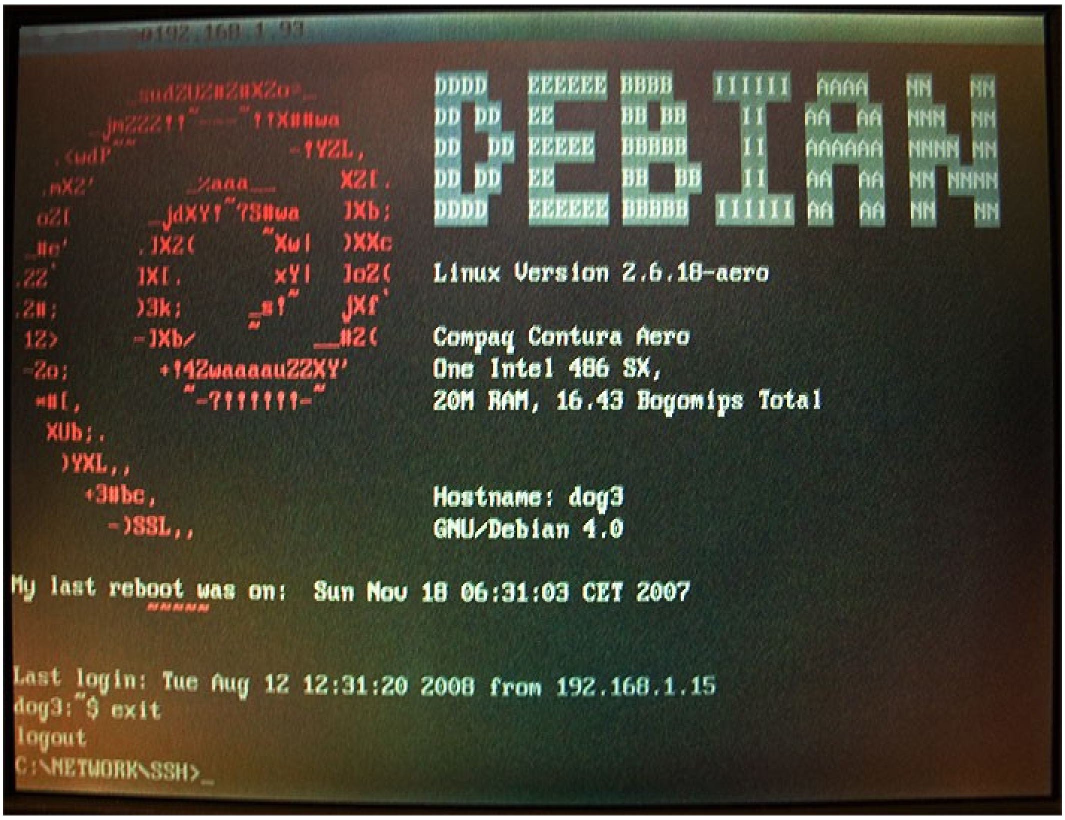
The browser supports various picture formats. Tables and frames are shown correctly, it renders HTML 4 and CSS 1.0 and the latest version (1.90J3) even understands UTF-8. Other Arachne services include FTP, NNTP, IRC-Chat, RSS, POP3 and SMTP. Look here for a complete history of the software.

Arachne is still actively developed - programmers are invited to join a mailing list. The latest version can be found at the site of Glenn McCorkle. Read more about it at Wikipedia.



```
19:02 01680/296544
0 Command:
NOS version EZNOS (80386)
Based on JNOS 1.11f by Johan. K. Reinalda and James P. Dugal
Compiled by DOS Solutions,
As EZNOS, DOS Ethernet FTP/HTTP Server
Copyright 1991 by Phil Karn (KA9Q) and contributors.
XMS detected.
C:/NETWORK/NOS eznos> _
```

"**EZ-NOS 2**" is one of the descendants of Phil Karn's KA9Q. It is currently developed by DOS Solutions. The software suite includes a webserver, a FTP-server and a bootp server as well as an email client. As all KA9Q descendants EZ-NOS 2 is licensed with the GNU GPL. The source can be downloaded as ez_src.zip.



```
Linux Version 2.6.18-aero

Compaq Contura Aero
One Intel 486 SX,
20M RAM, 16.43 Bogomips Total


Hostname: dog3
GNU/Debian 4.0

My last reboot was on:  Sun Nov 18 06:31:03 CET 2007


Last login: Tue Aug 12 12:31:20 2008 from 192.168.1.15
dog3:~$ exit
logout
C:\NETWORK\SSH>_
```

"**SSH2DOS**" is a SSH, SFTP and SCP client for DOS. These services more and more replace the classic telnet- and FTP-services, which are regarded as less secure. (The screenshot shows a closed SSH session to my Debian server.)

SSH2DOS was created by Hungarian developer Nagy Daniel in 2000 and can be downloaded at Sourceforge.net. It is released under the GNU GPL. It uses code of the PuTTY SSH client as well as the WatTCP kernel library.

In the next chapter of this documentation we will learn how to add TCP/IP functions to a DOS machine, so we can install and use TCP/IP applications like those mentioned above.

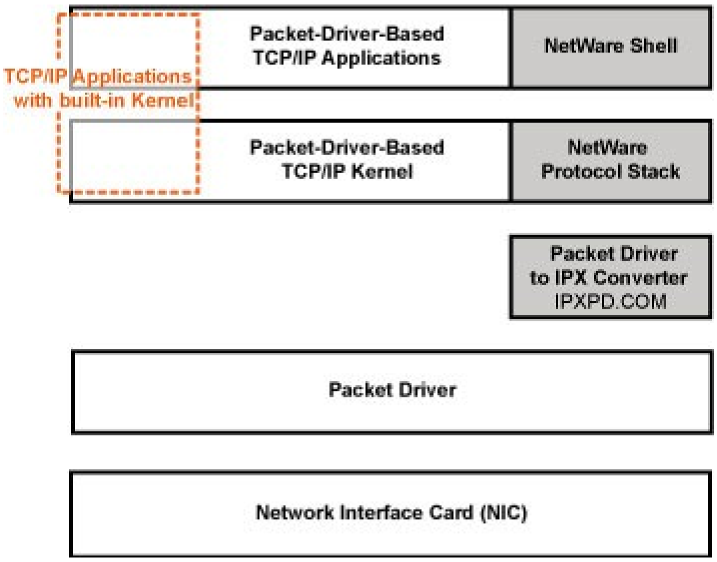
3. Practical Guide: Adding TCP/IP to DOS

The majority of TCP/IP applications for DOS uses the packet driver interface. So to be able to run them, we need to install the appropriate drivers first.

To install a generic packet driver is the simplest way. But it isn't always possible, as modern network cards often only come with ODI- and NDIS-drivers. Anyway: ODI- or NDIS-drivers can also be used as packet drivers through special converters ("shims"). This is just a bit more complicated.

The following chapters describe where to obtain drivers and converters and how to configure them. Afterwards we will take a look at the TCP/IP kernels available.

Packet drivers



Packet drivers work as an interface between the hardware (the network interface card) and the TCP/IP kernel (which is also called "protocol stack"). One sort of TCP/IP kernels work as an external program that can be called and used by various network applications. Other TCP/IP kernels are already built into network applications, a web browser for instance or a ftp-client. We will learn more about these kernels a bit further in this document.

As already mentioned, packet drivers are multiprotocol drivers so TCP/IP isn't the only kernel that can work atop of it. As you can see in the figure above, it is also possible to run Novell NetWare over of a packet driver: Specialized drivers like **IPXPD.COM** or **PDIPX.COM** support IPX over the packet driver interface. The NetBEUI protocol can't be used on top of a packet driver though, as the packet driver interface is too different from NDIS.

The first place to look for a packet driver should be the installation medium that came with your card. Packet drivers often have the letters "**PD**" in their names, so the packet driver of a 3com 3C589 PC-Card is called "**3C589PD.COM**" and the driver of the D-Link

DFE-670TXD PC-Card is called "**DFE670PD.COM**". Look for a directory "**PKTDRV**" on the CD or floppy that came with your card.

If there is no such driver on your installation medium or you don't have any, try searching the web. For ISA and PCI network cards there is a chance a packet driver can be found at Russell Nelson's Crynwr project - a resource of public domain packet drivers. PCMCIA drivers seem to be rare there, though.

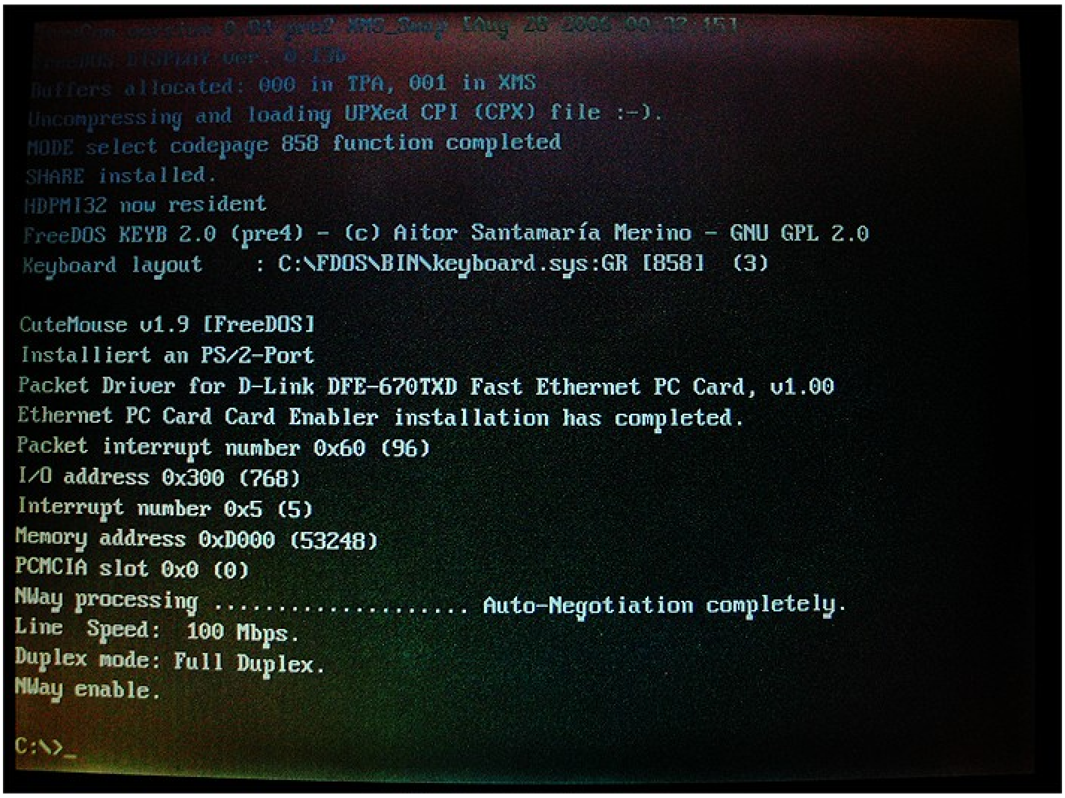
Georg Potthast provides a collection of PCI card packet drivers and a tool called **NICSCAN.EXE** to determine the chipset of PCI network cards. He made the experience that packet drivers are often the same for a large number of models by the same manufacturer, so he recommends not to try finding a packet driver specific for your model number.

Installing a packet driver

To install a packet driver, add a line like this to your **AUTOEXEC.BAT** (example for the 3c589 PCMCIA card):

```
LH 3C589PD.COM 0x60 5 0x300
```

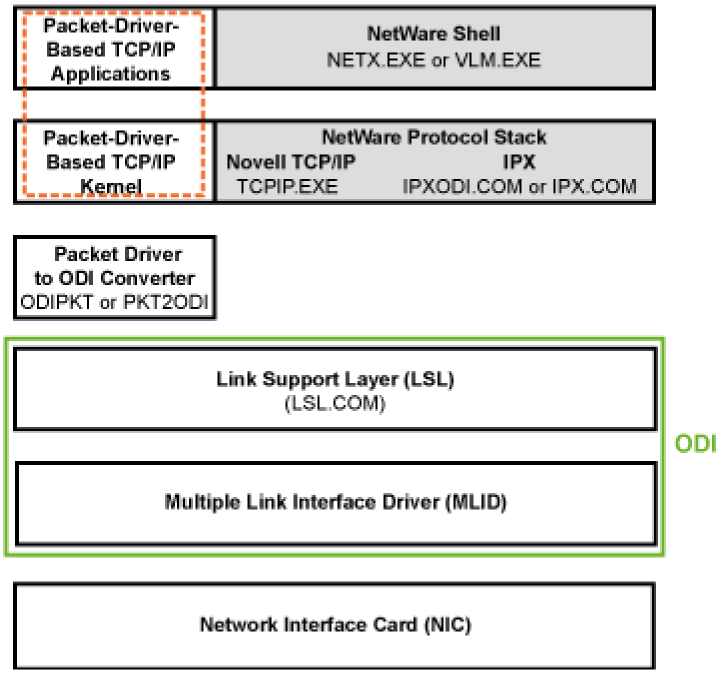
In the example above the driver is loaded into high memory by using the command "**LH**". The first option ("**0x60**") sets the software interrupt (vector) used by the driver. The most frequently used packet driver software interrupt number is 0x60. The second option ("**5**") sets the IRQ, the third option ("**0x300**") sets the I/O port. Some drivers only need the vector and find the other values by themselves. Most packet drivers can be unloaded after use with the option "**-u**".



```
FreeCom version 0.84-pre2 XMS_Swap [Aug 28 2006 00:32:15]
FreeDOS DISPLAY ver. 0.13b
Buffers allocated: 000 in TPA, 001 in XMS
Uncompressing and loading UPXed CPI (CPX) file :-).
MODE select codepage 858 function completed
SHARE installed.
HDPMI32 now resident
FreeDOS KEYB 2.0 (pre4) - (c) Aitor Santamaría Merino - GNU GPL 2.0
Keyboard layout     : C:\FDOS\BIN\keyboard.sys:GR [858]  (3)

CuteMouse v1.9 [FreeDOS]
Installiert an PS/2-Port
Packet Driver for D-Link DFE-670TXD Fast Ethernet PC Card, v1.00
Ethernet PC Card Card Enabler installation has completed.
Packet interrupt number 0x60 (96)
I/O address 0x300 (768)
Interrupt number 0x5 (5)
Memory address 0xD000 (53248)
PCMCIA slot 0x0 (0)
NWay processing .................... Auto-Negotiation completely.
Line  Speed:  100 Mbps.
Duplex mode: Full Duplex.
NWay enable.

C:\>_
```

That's all. After successfully installing a packet driver, you can skip the next two steps (ODI / NDIS drivers) and proceed with the TCP/IP kernel.

ODI drivers


Packet-Driver-Based TCP/IP Applications
NetWare Shell
NETX.EXE or VLM.EXE
Packet-Driver-Based TCP/IP Kernel
NetWare Protocol Stack
Novell TCP/IP
TCPIP.EXE
IPX
IPXODI.COM or IPX.COM
Packet Driver to ODI Converter
ODIPKT or PKT2ODI
Link Support Layer (LSL)
(LSL.COM)
Multiple Link Interface Driver (MLID)
ODI
Network Interface Card (NIC)

"Open Datalink Interface" (ODI) drivers were originally needed for Novell NetWare. But that doesn't mean they can not be used for packet-driver-based TCP/IP applications. In fact, they were used for this since they were invented.

An ODI driver is a good choice, if you cannot get a packet driver for your NIC. Because NetWare was a popular system during the eighties and nineties of the last century, there is quite a good chance that the manufacturer of your network card included such an ODI driver.

Look for a directory called "**netware**", "**nwclient**", "**ODI**" or "**VLM**" on the installation floppy or CD-ROM of your card. The drivers are executables, their names look like "**3C574.COM**" or "**DFE670.COM**".

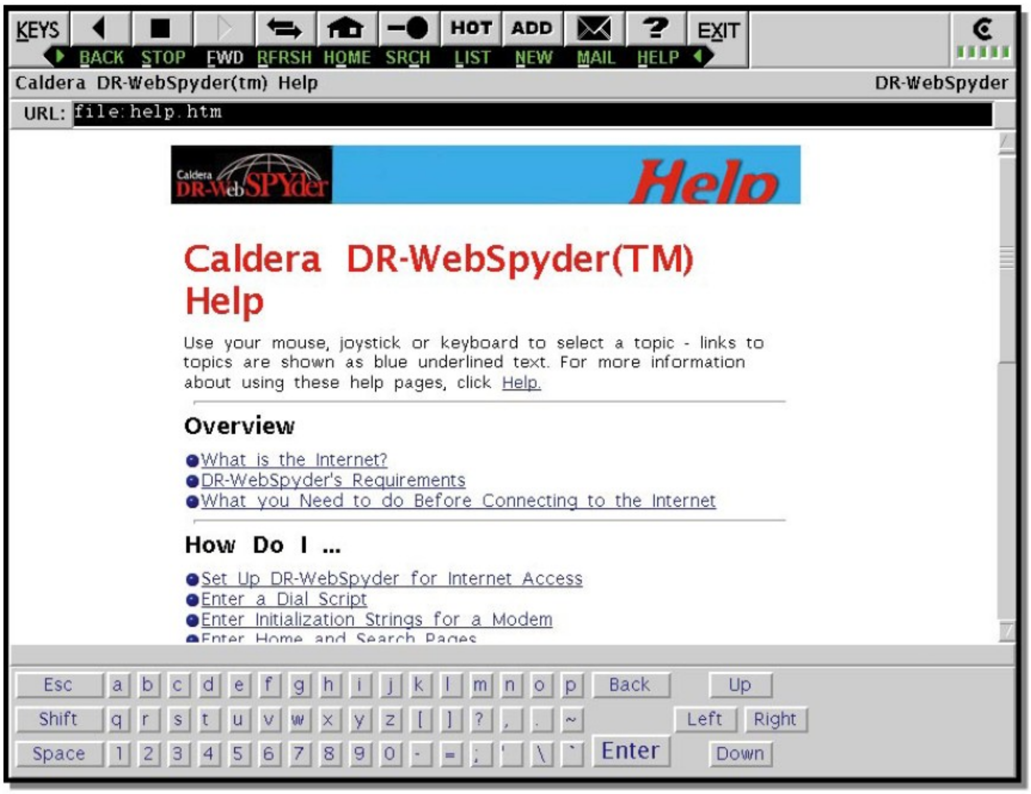
In the language of the ODI specification these drivers of the network card are called **"Multiple Link Interface Driver" (MLID)**.

As you see in the figure above, the MLID communicates directly with the hardware of the network interface card. The MLID receives packets for different protocol stacks (kernels) in the system and passes these packets to a second piece of software, the **"Link Support Layer" (LSL)**. The LSL then determines which protocol stack is to receive the packet. Both, MLID and LSL form the ODI architecture.

So we need an additional file to get ODI working: The LSL, which is a file called "**LSL.COM**". This file is part of the "DOS NetWare Client", which can be downloaded at Novell. Novell's download site explicitly remarks "LICENSE: FREE" for it. After download and extraction, the file "**LSL.COM**" can be found in the folder "**NIOS**".

As we want to use ODI with packet driver based TCP/IP applications, we have to download another piece of software, a **"Packet Driver to ODI Converter"**. Converters are also called "wrappers" or a "shim". We have three options.

- `ODIPKT.COM` v3.1 by Dan Lanciani (http://www.danlan.com) at Harvard University. The software is public domain; its license allows free redistribution of binary and source and the modification of the source (assembler).


KEYS
BACK
STOP
FWD
RFRSH
HOME
SRCH
HOT
LIST
ADD
NEW
MAIL
HELP
EXIT
Caldera DR-WebSPYder
Help
Esc a b c d e f g h i j k l m n o p Back Up
Shift q r s t u v w x y z [ ] ? , . ~ Left Right
Space 1 2 3 4 5 6 7 8 9 0 - = ; ' \ ` Enter Down

Caldera DR-WebSpyder(tm) Help DR-WebSpyder

URL: file:help.htm

Caldera DR-WebSpyder(TM) Help

Use your mouse, joystick or keyboard to select a topic - links to topics are shown as blue underlined text. For more information about using these help pages, click Help.

Overview

- What is the Internet?
- DR-WebSpyder's Requirements
- What you Need to do Before Connecting to the Internet

How Do I ...

- Set Up DR-WebSpyder for Internet Access
- Enter a Dial Script
- Enter Initialization Strings for a Modem
- Enter Home and Search Pages

- `PKT2ODI.EXE` by Caldera. This file is part of the DR "WebSpyder" software, a graphical DOS browser, that Caldera released in 1998. WebSpyder was licensed from xChaos/Arachne and can be downloaded here. It's license allows to evaluate and redistribute it for non-commercial purposes.

- If you are already running full NetWare, you can use `IPXPKT.COM` to run a packetdriver over `IPX.COM`. `IPXPKT.COM` is free software by Crynwr and is a part of their packet driver collection `PKTD11.ZIP`. This is a special case and will not be further investigated in this document.

With the MLID, the LSL and the converter we have most of the files needed. The only one missing is the configuration file "`NET.CFG`". An example file with settings supposed for your card may be found on the CD or floppy that came with it and should be located in the same directory as the ODI driver.

Configuration of NET.CFG

Now, lets edit our main configuration file `NET.CFG`. If it doesn't exist, create it. You have to choose a directory - default locations of `NET.CFG` seem to be `C:\NWCLIENT` or `C:\NET`. Anyway, the location should be in the same directory, where `LSL.COM` and the driver can be found.

An example **`NET.CFG`** which came with the 3Com 3C574 PC Card shows a lot of configuration settings, that are needed for Novell NetWare and other software. For using packet driver based TCP/IP applications the following four lines in **`NET.CFG`** are enough:

```
--- NET.CFG ---

Link Support
  buffers 8 1600

Link Driver 3C574
 Frame Ethernet_II
```

Explanation of the parameters

```
link support
```

The line "**`Link Support`**" configures the LSL. Options are: "**`buffers`**", "**`max boards`**", "**`max stacks`**" and "**`mempool`**". We only need to set:

```
buffers 8 1600
```

This determines the number and size of the receive buffers of the LSL. The default communication number for TCP/IP is 8. The author of ODIPKT, Dan Lanciani recommends to use a buffer size of 1600 bytes for ODIPKT.

```
Link Driver 3C574
```

The line "**`Link Driver`**" configures the MLID (the driver of your network card). First of all, the name of the driver is specified. Above you see the example of a 3com 3C574 PC Card. Change this to the name of your own NIC.

Then we configure the "**`frame`**" or "**`envelope type`**" (both declarations work and mean the same thing) used by this driver:

```
Frame Ethernet_II
```

It is possible to define more than one frame type here: We already heard that ODI is a multiprotocol driver, so the MLID is able to use more than one frame and protocol with the same hardware network board. For this purpose it defines *logical boards* for each defined frame. Possible frames are for instance "**`ETHERNET_II`**" (IP protocol), "**`ETHERNET_802.2`**" or "**`ETHERNET_802.3`**" (both IPX/SPX protocol).

For the packet driver converter, that we want to start later, it is mandatory to define at least the Ethernet II frame here. You also have to inform ODIPKT in a command line parameter to use the board with the **`ETHERNET_II`** frame (see below).

For a full documentation of the parameters in **NET.CFG** see the "NetWare Client for DOS and MS Windows Technical Reference".

Installing the ODI drivers

After we have configured **NET.CFG** we may start the ODI drivers in **AUTOEXEC.BAT** at boot.

The first thing we have to start is the Link Support Layer:

```
LH C:\NETWORK\NWCLIENT\LSL.COM /C C:\NETWORK\NET\NET.CFG
```

Please change the paths to the ones you use on your system. The "**/C**" option tells the LSL where to find **NET.CFG**. This option is only necessary, if **NET.CFG** is not in the same directory as **LSL.COM**.

Now we start the MLID:

```
LH C:\NETWORK\PCMCIA\3C574\ODI\3C574.COM
```

As a last step we have to install the Packet Driver to ODI Converter. Here we can choose between ODIPKT or PKT2ODI.

Alternative 1: ODIPKT

To use the free converter **ODIPKT.COM** by Dan Lanciani, we start it with a command like this in **AUTOEXEC.BAT**:

```
LH C:\NETWORK\NWCLIENT\ODIPKT.COM 0 96
```

The first number of the parameters above ("**0**") determines the board that uses the **ETHERNET_II** frame. The following example assumes you defined different frames (also known as "envelope types") in **NET.CFG**:

```
--- NET.CFG ---

Link Driver 3C574
  FRAME ETHERNET_II
  FRAME ETHERNET_802.2
  FRAME ETHERNET_802.3
  FRAME ETHERNET_SNAP
```

Then you have to tell ODIPKT the index number of the logical board that supports the Ethernet II frame. Just count the frames in NET.CFG: Ethernet II is the first one, the frame 802.2 is the second, the frame 802.3 is the third, snap is fourth.

To make things a bit tricky, ODIPKT starts counting with "`0`" (like programmers do). That means, according to the example aove:

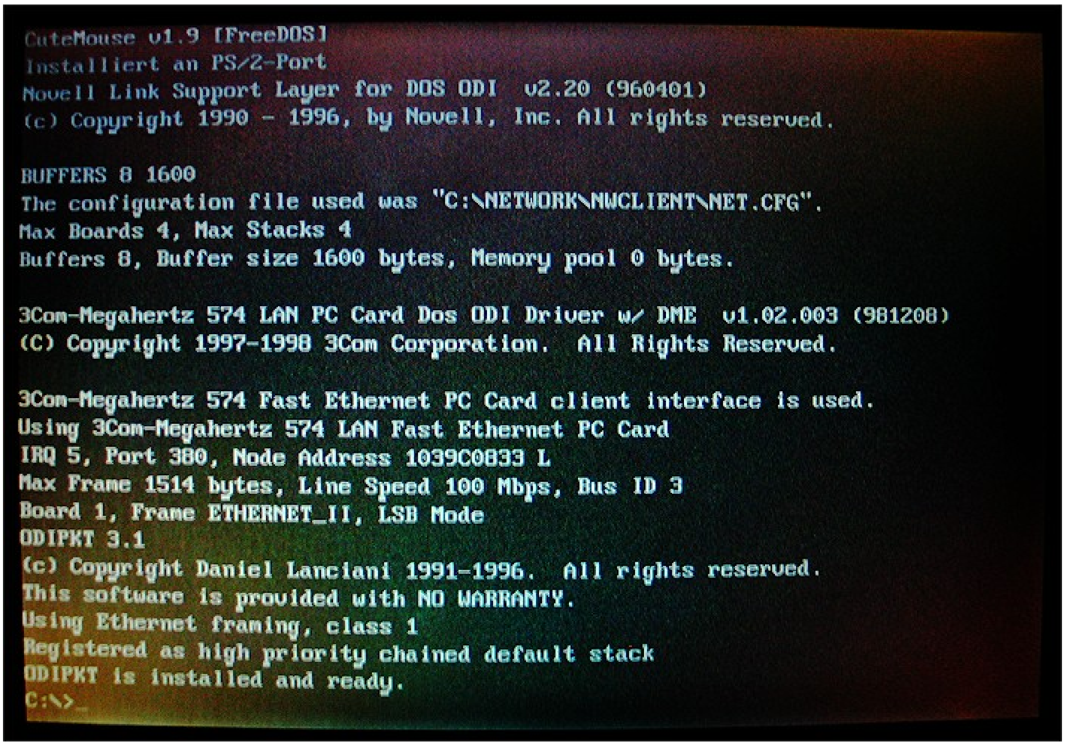
```
ODIPKT.COM 0 | board with ETHERNET_II
ODIPKT.COM 1 | ETHERNET_802.2 (won't work)
ODIPKT.COM 2 | ETHERNET_802.3 (won't work)
ODIPKT.COM 3 | ETHERNET_SNAP (won't work)
```

The only option that works for ODIPKT is the number that defines the logical board with the frame ETHERNET II, which is "`0`" in our example.

Now we look at the second parameter ("`96`") that we use to start ODIPKT. This parameter defines the software interrupt (vector) used by the driver. As we already heard when configuring a packet driver, the most frequently used packet driver software interrupt number is 0x60, which is the hexadecimal value 60 (the "0x" just tells us it is hex format). Unfortunately ODIPKT doesn't understand hex values, so you have to translate this parameter to a decimal number - which is 96.

If you want to use other values and you are not used to hex, you may try a hex-dec calculator or see the following list:

```
0x60 = 96
0x61 = 97
0x62 = 98
 ...
0x69 = 105
0x6A = 106
 ...
0x7D = 125
0x7E = 126
```



```
CuteMouse v1.9 [FreeDOS]
Installiert an PS/2-Port
Novell Link Support Layer for DOS ODI  v2.20 (960401)
(c) Copyright 1990 - 1996, by Novell, Inc. All rights reserved.

BUFFERS 8 1600
The configuration file used was "C:\NETWORK\NWCLIENT\NET.CFG".
Max Boards 4, Max Stacks 4
Buffers 8, Buffer size 1600 bytes, Memory pool 0 bytes.

3Com-Megahertz 574 LAN PC Card Dos ODI Driver w/ DME  v1.02.003 (981208)
(C) Copyright 1997-1998 3Com Corporation.  All Rights Reserved.

3Com-Megahertz 574 Fast Ethernet PC Card client interface is used.
Using 3Com-Megahertz 574 LAN Fast Ethernet PC Card
IRQ 5, Port 380, Node Address 1039C0833 L
Max Frame 1514 bytes, Line Speed 100 Mbps, Bus ID 3
Board 1, Frame ETHERNET_II, LSB Mode
ODIPKT 3.1
(c) Copyright Daniel Lanciani 1991-1996.  All rights reserved.
This software is provided with NO WARRANTY.
Using Ethernet framing, class 1
Registered as high priority chained default stack
ODIPKT is installed and ready.
C:\>_
```

Now the packet driver is installed and ready. You may use an application to test if it works, for instance the command "ping" from the WATTCP package. Good luck!

Alternative 2: PKT2ODI

As an alternative we may use Calderas **PKT2ODI.EXE** as our Packet Driver to ODI Converter.

Please be aware, that the license for Novell's WebSpyder (which applies to **PKT2ODI.EXE**) only allows to use the software for the purpose of evaluating it. For commercial users the evaluation period is restricted to 90 days. For non-commercial users evaluation is allowed for "a reasonable period" of time. (Just how reasonable can it be to evaluate this software in 2007?)

Anyway, **PKT2ODI.EXE** will be started in **AUTOEXEC.BAT** with a command like this:

```
LH C:\NETWORK\NWCLIENT\PKT2ODI.EXE /B:1 /I:69
```

The "**/B**" option tells **PKT2ODI.EXE** which logical board it should use. Like explained above we have to choose the board that uses the Ethernet II frame. In difference to ODIPKT, PKT2ODI starts counting with "**1**", so if the **NET.CFG** looks like this example:

```
--- NET.CFG ---


Link Driver 3C574
  FRAME ETHERNET_II
  FRAME ETHERNET_802.2
  FRAME ETHERNET_802.3
  FRAME ETHERNET_SNAP
```

we have to use the parameter

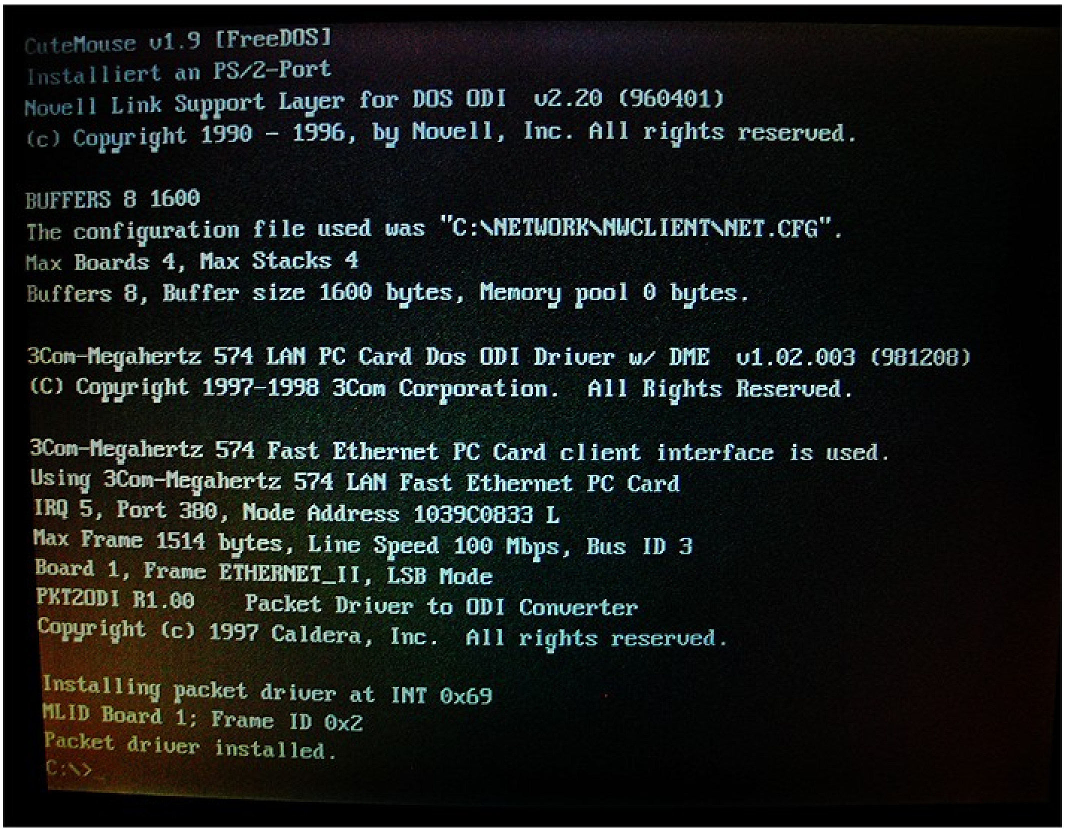
```
PKT2ODI.EXE /B:1
```

to define the use of the first board (the one with **ETHERNET_II**). Remember that we have to use the board with Ethernet II, otherwise the converter won't be able to communicate with the driver and complain about not finding the MLID.

The second parameter, the "**/I**" sets the software interrupt (vector) used by the driver. We already found out that this should be 0x60, but unfortunately this won't work with PKT2ODI. The program doesn't accept interrupt vectors 0x60 to 0x68. We may use vector 0x69 with the parameter:

```
PKT2ODI.EXE /I:69
```

Using an unusual interrupt vector like 0x69 can be a problem for some TCP/IP applications. They must be configured to use this vector. Some programs, like the webserver SIOUX, may not work correctly, if the vector is not 0x60.

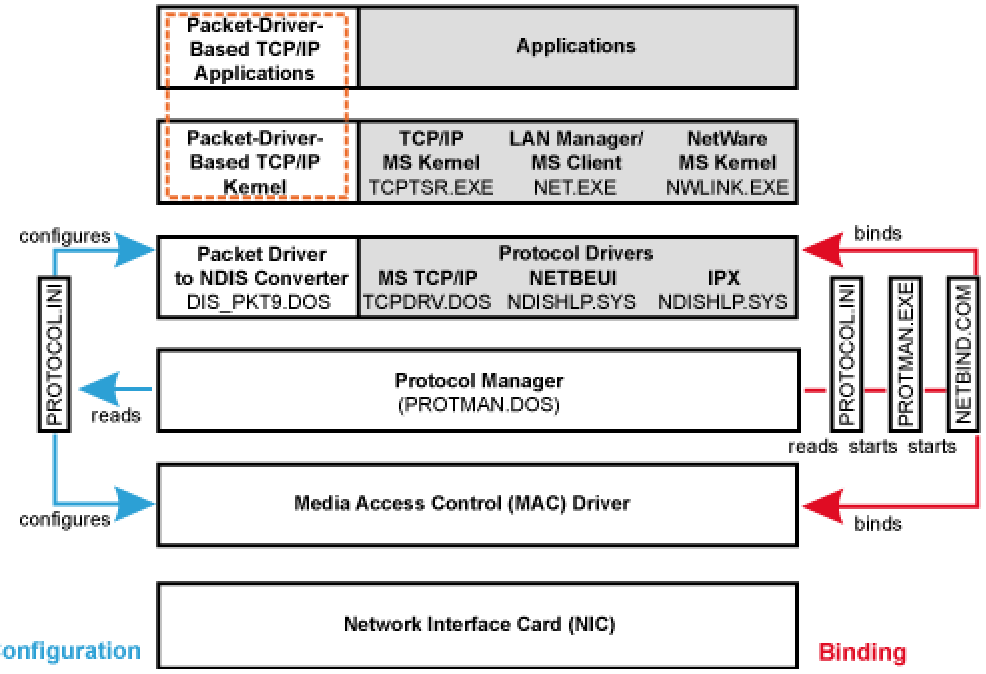


```
CuteMouse v1.9 [FreeDOS]
Installiert an PS/2-Port
Novell Link Support Layer for DOS ODI  v2.20 (960401)
(c) Copyright 1990 - 1996, by Novell, Inc. All rights reserved.

BUFFERS 8 1600
The configuration file used was "C:\NETWORK\NWCLIENT\NET.CFG".
Max Boards 4, Max Stacks 4
Buffers 8, Buffer size 1600 bytes, Memory pool 0 bytes.

3Com-Megahertz 574 LAN PC Card Dos ODI Driver w/ DME  v1.02.003 (981208)
(C) Copyright 1997-1998 3Com Corporation.  All Rights Reserved.

3Com-Megahertz 574 Fast Ethernet PC Card client interface is used.
Using 3Com-Megahertz 574 LAN Fast Ethernet PC Card
IRQ 5, Port 380, Node Address 1039C0833 L
Max Frame 1514 bytes, Line Speed 100 Mbps, Bus ID 3
Board 1, Frame ETHERNET_II, LSB Mode
PKT2ODI R1.00    Packet Driver to ODI Converter
Copyright (c) 1997 Caldera, Inc.  All rights reserved.

Installing packet driver at INT 0x69
MLID Board 1; Frame ID 0x2
Packet driver installed.
C:\>_
```

That's all. After starting the LSL, the MLID and the converter the packet driver interface should now be up and running. Proceed with the chapter "TCP/IP kernel".

You can also use a NDIS driver for packet-driver-based TCP/IP applications. This will be explained in the next chapter.

NDIS drivers


Packet-Driver-Based TCP/IP Applications
Applications
Packet-Driver-Based TCP/IP Kernel
TCP/IP MS Kernel TCPTSR.EXE
LAN Manager/ MS Client NET.EXE
NetWare MS Kernel NWLINK.EXE
configures
Packet Driver to NDIS Converter DIS_PKT9.DOS
Protocol Drivers
MS TCP/IP TCPDRV.DOS
NETBEUI NDISHLP.SYS
IPX NDISHLP.SYS
binds
PROTOCOL.INI
reads
Protocol Manager (PROTMAN.DOS)
PROTOCOL.INI
PROTMAN.EXE
NETBIND.COM
reads starts starts
configures
Media Access Control (MAC) Driver
binds
Configuration
Network Interface Card (NIC)
Binding

Drivers based on the "Network Driver Interface Specification" (NDIS) are the default in the Windows world. The most recent version is NDIS 6 which is used in Windows Vista. Windows 98 used NDIS 5, Windows 95 came with NDIS 3.1. In DOS you have to use the real-mode driver NDIS2.

It is most likely a NDIS driver exists for your network card. Look for a directory called "`NDIS2`" in your cards installation files. If you can't find a driver, take a look at this site. The ending of the driver name has to be `*.DOS`, so for instance the driver of the 3com 3C574 PC-Card is called "`EL3C574.DOS`". In the language of the NDIS architecture these drivers are called **"Media Access Control" (MAC)** drivers.

The MAC driver is only one component of the NDIS architecture as you can see in the figure above. According to the NDIS 2.1 documentation NDIS seems to work like that:

- Commands in `FDCONFIG.SYS` (FreeDOS) or `CONFIG.SYS` (other DOS) load the Protocol Manager driver `PROTMAN.DOS`, the MAC driver and the protocol driver. This can be done by lines for each of these drivers or by starting the "Installable File System Helper" driver, which is loaded by "`DEVICEHIGH=IFSHLP.SYS`" and starts the Protocol Manager, the MAC and the protocol drivers according to `PROTOCOL.INI`.
- The Protocol Manager reads the configuration from `PROTOCOL.INI` and makes them available to the MAC driver and protocol driver which load after him.
- The MAC driver and the protocol driver load and configure themselve according to that information. They identify themselves to the Protocol Manager.
- Before MAC driver and protocol driver can communicate, they have to be bound together, so they can access each others "entry points" (communication ports). This bind process is controlled by the Protocol Manager based on information in `PROTOCOL.INI`.
- The binding sequence can be triggered by `NET.EXE` or `NETBIND.COM` (which can be done in `AUTOEXEC.BAT`).
- The Protocol Manager (`PROTMAN.DOS`) then starts the memory resident (Terminate and Stay Resident - TSR) program `PROTMAN.EXE` to execute the bind command and to control the correct ordering of drivers.

- After binding has occurred, the MAC and the protocol driver communicate directly to each other. The Protocol Manager is not involved in this communication process any longer, so `NETBIND.COM` frees most of the memory previously reserved by the Protocol Manager.

The configuration of NDIS under DOS has changed over time with different versions of the package. Some hints about the differences can be found here. In this document, I am assuming that we use the NDIS files shipped with MS Client 3.0.

As this section is all about getting packet driver based TCP/IP applications up and running with DOS I will not go into details about MS Client here (as I mentioned above, this is already explained elsewhere).

So how can we use the NDIS2 (MAC) driver that came with our network interface card for TCP/IP networking? As we have learned from the description above, we need some more files. These are:

- The Protocol Manager ("`PROTMAN.DOS`" and "`PROTMAN.EXE`")
- The bind utility ("`NETBIND.COM`")

These files are part of the MS Client 3.0 which can be downloaded here:

ftp://ftp.microsoft.com/bussys/Clients/MSCLIENT/DSK3-1.EXE
ftp://ftp.microsoft.com/bussys/Clients/MSCLIENT/DSK3-2.EXE

The three files mentioned are part of "`DSK3-1.EXE`". Use the utility "`EXPAND.EXE`" included on the first disk to uncompress these files.

- Copy `DSK3-1.EXE` to a directory like `C:\MSCLIENT1`. Avoid a long path if you do this in Windows. 16-bit software won't execute, if the path is too long.
- Run `DSK3-1.EXE` to unpack its content.
- Decompress `PROTMAN.DO_` and `PROTMAN.EX_`.

```
expand -r protman.do_

expand -r protman.ex_
```

- `NETBIND.COM` is already uncompressed. Now you can take and use the three files.

Read the `LICENSE.TXT` file before you start using the files. The software is free for internal use. Neither the distribution nor a modification is allowed.

The last piece we need is the **"Packet Driver to NDIS Converter"**. Get the widely used "`DIS_PKT9.DOS`" (version 9) or alternatively the slightly newer (version 11) "`DIS_PKT.DOS`". There are no differences in usage and handling.

Both were written by Prof. Joe R. Doupnik (Utah State University) and Dan Lanciani (Harvard University). The copyright holds the now historic FTP Software, Inc., which luckily released this driver as public domain, free for use, distribution, change and with sourcecode.

Configuration of NDIS drivers

As I already mentioned, this is just a minimal installation of the MS Client 3.0. The only purpose is to use packet-driver-based TCP/IP applications. For this purpose:

Copy the four files

```
PROTMAN.DOS
PROTMAN.EXE
NETBIND.COM
DIS_PKT.DOS
```

into a directory, for instance **C:\NET**.

Copy also the **MAC driver** of your network interface card, for instance:

```
EL3C574.DOS
```

into this directory.

Now we have to create the configuration file needed for the NDIS architecture: Create a file with the name **PROTOCOL.INI** inside the **C:\NET** directory. For our minimal configuration it just needs the following lines:

```
--- PROTOCOL.INI ---

[protman]
DriverName=PROTMAN$

[EL3C574]
DriverName=EL3C574$

[PKTDRV]
drivername=PKTDRV$
bindings=EL3C574
intvec=0x60
chainvec=0x68
```

Explanation of the parameters

Like other *.ini files, **PROTOCOL.INI** is structured into section names in square brackets and item names with values assigned to them.

```
[protman]
DriverName=PROTMAN$
```

The first section **[protman]** defines the Protocol Manager. Like described above, the Protocol Manager is loaded as driver **PROTMAN.DOS** which is called "**PROTMAN$**" here. Section and line are mandatory.

```
[EL3C574]
DriverName=EL3C574$
```

The section `[EL3C574]` defines the network interface card. By default the section itself is named after the card - this is useful, if you have more than one network card and use different sections for different cards.

So the section name is the first value to change here. It would be a good idea to name it after your own network card. But you may as well rename the section to "netcard", "NIC" or even "baked_beans" - whatever, as long as you also change all the other lines in `PROTOCOL.INI` that point to that section.

The next line of this section is "`DriverName=EL3C574$`". It defines the driver for the network interface card. In our example this is `EL3C574.DOS`, which is called "`EL3C574$`" here. Most certainly you will use a different NIC than the one mentioned in the example. So you will have to change this line. You find the correct name of the driver in a text-file called "`PROTOCOL.INI`" that should be part of the NDIS driver files supplied with your card.

It is also possible to add more lines to this section, so you can define special settings for your network card. Consult the "`PROTOCOL.INI`" supplied with your NDIS driver for more information. But in many cases the line with the drivername should be enough.

```
[PKTDRV]
drivername=PKTDRV$
bindings=EL3C574
intvec=0x60
chainvec=0x68
```

The next section "`[PKTDRV]`" defines our Packet Driver to NDIS Converter, which is "`DIS_PKT.DOS`" or "`DIS_PKT9.DOS`". Both are called by the name "`PKTDRV$`".

It is important that the converter will bind to the driver of the network card. This is done by the next line "`bindings=EL3C574`". Please note that this name includes no "`$`" letter - it refers to the name of the **section** that defines the driver, not to the driver itself. So if you had called this section "`[baked_beans]`" as I have suggested above, you would have to write "`bindings=baked_beans`" here ;-).

The item "`intvec`" specifies the software interrupt vector used by the packet driver interface. As already explained, this should be the hexadecimal value 0x60 per default.

The "`chainvec`" item also defines an available software interrupt. Its function is still a mystery to me. According to packet driver inventor FTP, adding a chain vector interrupt may improve packet processing speed and reliability. Users saw "a 10-fold increase in performance". To avoid EMM386 errors, some recommend to set an interrupt that increases the vector by decimal 8. So if the `intvec` is 0x60 (that is decimal 96), then the `chainvec` should be decimal 104 (96+8) which is hexadecimal 0x68. This is also described by one of the authors of `DIS_PKT.DOS`, Dan Lanciani. Please write if you find out more.

These are all settings in **`PROTOCOL.INI`** needed for our purpose. A last hint: If you consider changing some of these values at each boot, for instance by choices presented by a DOS boot menu, you may find Horst Schaeffer's freeware "Inifile" helpful.

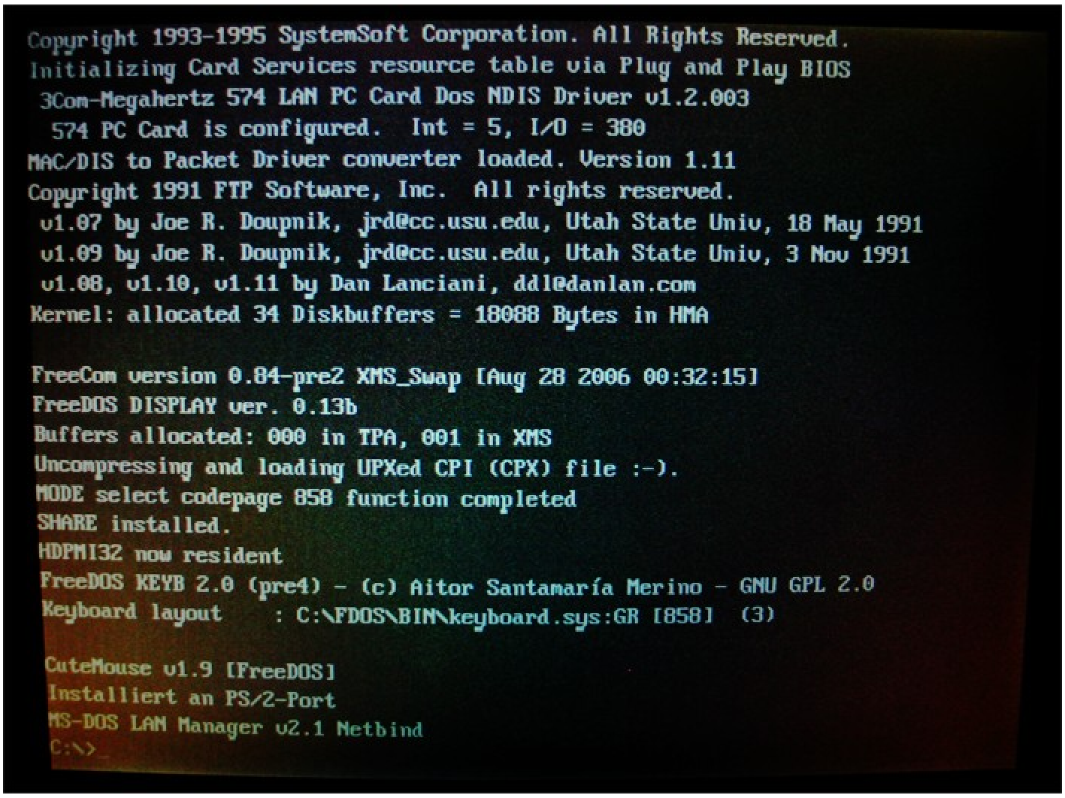
The last thing left to do now is to load the drivers at boot. So change your system files to contain the following lines:

```
--- FDCONFIG.SYS (FreeDOS) ---
--- or CONFIG.SYS (MS DOS/other DOS) ---

DEVICEHIGH=C:\NET\PROTMAN.DOS /I:C:\NET
DEVICEHIGH=C:\NET\EL3C574.DOS
DEVICEHIGH=C:\NET\DIS_PKT.DOS
```

The "**`/I`**" parameter tells the Protocol Manager the location of **`PROTOCOL.INI`**. This is not needed if both are in the same directory.

```
--- AUTOEXEC.BAT ---

C:\NET\NETBIND.COM
```

Note that **`NETBIND.COM`** can not be loaded high and should just be executed from **`AUTOEXEC.BAT`**. If you try to load it like a driver, it will abort with the message "**`run-time error R6009 - not enough space for environment`**".



```
Copyright 1993-1995 SystemSoft Corporation. All Rights Reserved.
Initializing Card Services resource table via Plug and Play BIOS
 3Com-Megahertz 574 LAN PC Card Dos NDIS Driver v1.2.003
  574 PC Card is configured.  Int = 5, I/O = 380
MAC/DIS to Packet Driver converter loaded. Version 1.11
Copyright 1991 FTP Software, Inc.  All rights reserved.
 v1.07 by Joe R. Doupnik, jrd@cc.usu.edu, Utah State Univ, 18 May 1991
 v1.09 by Joe R. Doupnik, jrd@cc.usu.edu, Utah State Univ, 3 Nov 1991
 v1.08, v1.10, v1.11 by Dan Lanciani, ddl@danlan.com
Kernel: allocated 34 Diskbuffers = 18088 Bytes in HMA

FreeCom version 0.84-pre2 XMS_Swap [Aug 28 2006 00:32:15]
FreeDOS DISPLAY ver. 0.13b
Buffers allocated: 000 in TPA, 001 in XMS
Uncompressing and loading UPXed CPI (CPX) file :-).
MODE select codepage 858 function completed
SHARE installed.
HDPMI32 now resident
FreeDOS KEYB 2.0 (pre4) - (c) Aitor Santamaría Merino - GNU GPL 2.0
Keyboard layout    : C:\FDOS\BIN\keyboard.sys:GR [858]  (3)

CuteMouse v1.9 [FreeDOS]
Installiert an PS/2-Port
MS-DOS LAN Manager v2.1 Netbind
C:\>_
```

So that's all. The best thing to do now is to reboot the system and use a simple TCP/IP application like ping to test if everything is working.

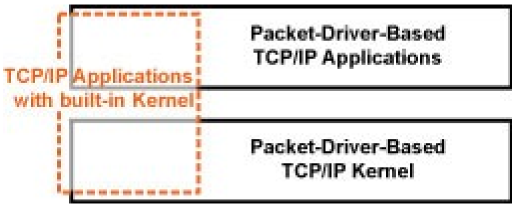
And this is exactly what we do next: We loaded the network drivers. So now we start TCP/IP.

TCP/IP Kernel (Stack)

The TCP/IP kernel is also called "stack" or "protocol stack", because it consists of two layers: IP and TCP. IP runs on top of the hardware driver and TCP on top of IP, providing an interface to network applications.

Modern operating systems have their own TCP/IP kernel already built-in: Microsoft systems come with "Winsock" since Windows 95. In GNU/Linux the TCP/IP stack is part of the Linux kernel.

For DOS different types of TCP/IP kernels are available.



As we see in the figure above, they can either work as external programs or they are already built into the applications.

- An *external* TCP/IP kernel is typically started by **`AUTOEXEC.BAT`** or by a batchfile. It stays memory resident, so it can answer ping requests for instance.
- Other network applications come with TCP/IP kernel functions already *built-in* and do not need an external TSR kernel. An application with built-in TCP kernel has the advantage that it doesn't need another driver which consumes precious DOS memory. On the other hand TCP/IP functions are limited to those of the network application and are only provided as long as it runs.

Every TCP/IP kernel must be configured with general information about the network, as IP address, netmask, nameservers and gateway, or it must be told to get this information via a BOOTP or DHCP server.

Examples for external TCP/IP kernels for DOS are the free Trumpet TCP kernel **`NTCPDRV.EXE`**, the TCP/IP kernel of Novell NetWare **`TCPIP.EXE`**, Microsoft MS Client's **`TCPTSR.EXE`** and FTP Software Inc.'s **`ETHDRV.EXE`**.

Examples for TCP/IP kernels that are already built-in into DOS applications are **WatTCP**, which is f.i. already built-in the graphical web browser "Arachne", the **KA9Q** kernel which is part of that program, the **NCSA** Telnet kernel which is built into the applications included in this suite, the **CUTCP** kernel and the University of Minnesota stack, which is part of "**Minuet**".

According to Jeffrey L. Hayes from the retrocomputing website http://www.old-skool.org more than half of the DOS networking applications available use the WatTCP kernel.

We will first take a closer look at WatTCP and its successor Watt-32. Then we will examine NTCPDRV which is the only free external TCP/IP kernel available. Then we briefly look at other external TCP/IP kernels.

WatTCP / Watt-32

Waterloo TCP (WatTCP) was written 1990 - 1992 at the University of Waterloo in Ontario (Canada). The author is Erick Engelke, then a developer in the Faculty of Engineering, today the university's Manager of Networks and System Integration. The core TCP engine of WatTCP was based on Geoffrey H. Cooper's TinyTCP.

WatTCP was originally released under a homebrewed public domain license - binary and source were free for use, distribution was allowed only for non-commercial reasons. In 2004 Erick Engelke copyrighted the libraries under the terms of the GNU LGPL. WatTCP can be found at Erick Engelke's homepage or as **WAT1104.ZIP** at various sites in the internet. While the software is free, the WatTCP Programmer's Reference is sold as PDF for 55$.

WatTCP was ported to 32-bit by Gisle Vanem at Bergen, Norway in 1999. The port is called Watt-32. It supports 32-bit protected-mode as well as 16-bit real-mode. It comes with uncompiled applications, compiled versions can be downloaded here. There is a developer forum which is still active.

WatTCP and Watt-32 are not external TCP/IP kernels. Both are just sets of *libraries* designated to programmers - they can use these libraries to implement TCP/IP functions into their applications. The WatTCP package includes such applications with built-in WatTCP kernel like "ping", "finger", "whois" or "lpr".

Some important DOS network applications use the WatTCP libraries like Arachne or ssh2dos. A list of these apps is available at "DOS Solutions".

Usage

So you don't have to download anything to use the WatTCP / Watt-32 kernel. It is already in the applications. The only thing needed is a configuration file, which is called "**WATTCP.CFG**" and is basically the same in both versions. It is located in the applications directory per default. The general settings:

```
--- WATTCP.CFG ---

# These are example values:
my_ip = 192.168.1.10
netmask = 255.255.255.0
nameserver = 192.168.1.1
nameserver = 129.97.128.196
nameserver = 129.97.128.1
gateway = 192.168.1.1

# Uncomment, if your receive your configuration via DHCP
# my_ip = dhcp
```

Additional settings may be needed by the applications used. A deeper introduction into the use of WatTCP programs can be found here.

TCPDRV / NTCPDRV

The Trumpet TCP driver is an *external* TCP/IP kernel. It works as a TSR program that runs on top of a packet driver.

TCPDRV was created in 1992 by Peter Tattam, a programmer in the Psychology department of the University of Tasmania at Hobart, Australia. According to a biography on his website, he developed the shareware "Trumpet Newsreader" short time before and wrote the TCP driver just because nothing else was openly available for Turbo Pascal. Development eventually lead to the popular Windows software "Trumpet Winsock", which Tattam sold through his own company "Trumpet International of Australia", founded in 1994.

TCPDRV 2.01 was released as "experimental version". In 1993 a version 3.01 followed, which was called NTCPDRV. Improvements included a more efficient memory usage and bug fixes. Both versions and the textfile **TCP201.SPE** (the specification for programmers) were made publicly available from the Trumpet website:

http://www.trumpet.com.au/dosapps/

Licensing issues on this site are kept short and simple: "These DOS applications are provided free without support." Thanks to the popularity of the trumpet software, it can be downloaded from various mirrors. A commercial version of the TCP driver is available from Peter Tattams new company "Tattam Software Enterprises".

Usage

Download **NTCPDRV.ZIP** and extract it. If you need a documentation, download the older version **TCP201.ZIP** as well, which also includes several applications.

The TCP/IP kernel has to be configured with the settings of your network. This is either done by commandline parameters or by setting DOS environment variables.

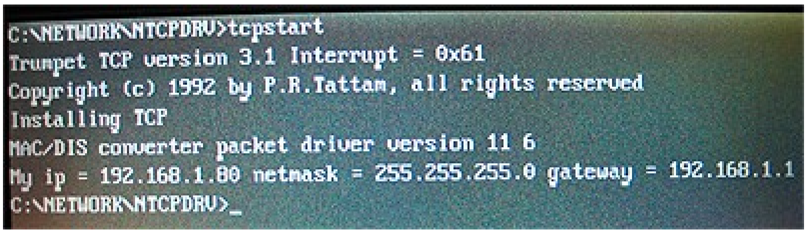
So start **NTCPDRV.EXE** with a command like this (one line):

```
NTCPDRV -ip=192.168.1.80 -netmask=255.255.255.0
-gateway=192.168.1.1 -dns-host=192.168.1.1
```

or configure it first by setting environment variables: Add those lines to **AUTOEXEC.BAT** or to a batchfile that will be started before **NTCPDRV.EXE**.

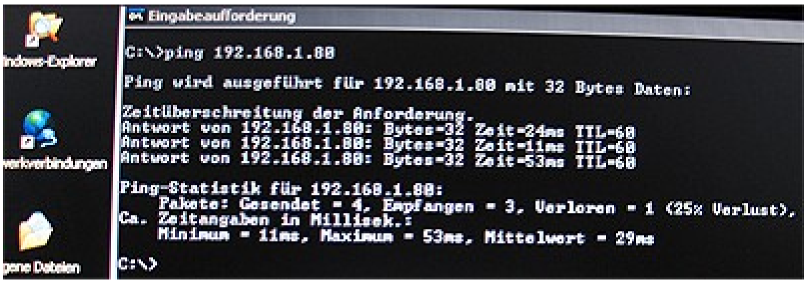
```
--- AUTOEXEC.BAT ---

set ip=192.168.1.80
set netmask=255.255.255.0
set gateway=192.168.1.1
set dns=192.168.1.1
```

As shown in the picture below, the kernel automatically searchs for a usable interrupt vector, after it is started. You may also use the parameter "**`-vec=61`**" to specify for instance the vector 0x61 that's provided by the packet driver for applications. See the documentation for more possibilities.



```
C:\NETWORK\NTCPDRV>tcpstart
Trumpet TCP version 3.1 Interrupt = 0x61
Copyright (c) 1992 by P.R.Tattam, all rights reserved
Installing TCP
MAC/DIS converter packet driver version 11 6
My ip = 192.168.1.80 netmask = 255.255.255.0 gateway = 192.168.1.1
C:\NETWORK\NTCPDRV>_
```

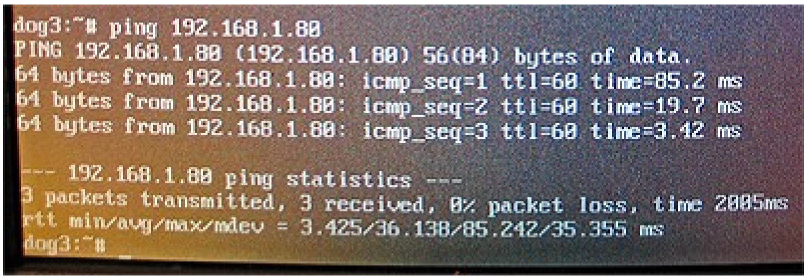
After the Trumpet TCP/IP kernel has been started, you are able to run several TCP/IP applications which require a TCP/IP kernel (for instance the Trumpet Newsreader, the DOS webservers Sioux or Webserv). Your machine will also be reachable from the network now, just try a ping request.



```
C:\>ping 192.168.1.80

Ping wird ausgeführt für 192.168.1.80 mit 32 Bytes Daten:

Zeitüberschreitung der Anforderung.
Antwort von 192.168.1.80: Bytes=32 Zeit=24ms TTL=60
Antwort von 192.168.1.80: Bytes=32 Zeit=11ms TTL=60
Antwort von 192.168.1.80: Bytes=32 Zeit=53ms TTL=60

Ping-Statistik für 192.168.1.80:
    Pakete: Gesendet = 4, Empfangen = 3, Verloren = 1 (25% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 11ms, Maximum = 53ms, Mittelwert = 29ms

C:\>
```

In the picture above, we ping the machine with FreeDOS and the running NTPCDRV TSR from another computer with Windows XP. It is not the fastest performance and the first packet is lost.

Then we ping it from a Debian GNU/Linux machine:



```
dog3:~# ping 192.168.1.80
PING 192.168.1.80 (192.168.1.80) 56(84) bytes of data.
64 bytes from 192.168.1.80: icmp_seq=1 ttl=60 time=85.2 ms
64 bytes from 192.168.1.80: icmp_seq=2 ttl=60 time=19.7 ms
64 bytes from 192.168.1.80: icmp_seq=3 ttl=60 time=3.42 ms

--- 192.168.1.80 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 3.425/36.138/85.242/35.355 ms
dog3:~#
```

The first packet needs more time again, the third is received in reasonable time. At least we see: The network is up and the Trumpet TCP driver is answering our calls.

Other TCP/IP Kernels

Some other external TCP/IP kernels can also still be legally used and downloaded. Especially the ones from Novell and Microsoft may be of use.

- Novell distributes its popular 16-bit TCP/IP protocol stack "`TCP16.EXE`", which runs directly atop of the ODI driver. A few programs like Josh's "Tiny" remote control software and Tsoft's "NFS Clients for DOS" still use this kernel.
- Microsoft's "MS Client 3.0" includes its own TCP/IP kernel "`TCPTSR.EXE`". It also can be used by external programs. It may be useful though, to check memory requirements first. MS Client's TCP/IP seems to consume a lot of conventional DOS memory.

Please look up Novell's or Microsoft's websites and archives for more information about installation and configuration.

Conclusion

The first chapters gave an overview over DOS networking. The last chapter explained, how to install and configure TCP/IP drivers and kernels. Now it's up to you to choose the network applications, you wish to run under DOS.

It took me months, to find all these things out, and again several weeks to write it down. It's quite a good feeling to have this accomplished now. But it is still not complete:

- I did not explain PPP, as I do not own a modem anymore.
- I also didn't experiment with DOS TCP/IP networking applications running under Windows or GNU/Linux or in an emulator.

So there's still something to do, something to find out. That's fine. Like one of my camera operators once said to me: "Work isn't a bear. It doesn't disappear in the woods. It's still around tomorrow."

Ulrich Hansen, 12.12.2007.

GNU Free Documentation License

GNU Free Documentation License
Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
H. Include an unaltered copy of this License.
I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and

```
   publisher of the Modified Version as given on the Title Page.  If
   there is no section Entitled "History" in the Document, create one
   stating the title, year, authors, and publisher of the Document as
   given on its Title Page, then add an item describing the Modified
   Version as stated in the previous sentence.
J. Preserve the network location, if any, given in the Document for
   public access to a Transparent copy of the Document, and likewise
   the network locations given in the Document for previous versions
   it was based on.  These may be placed in the "History" section.
   You may omit a network location for a work that was published at
   least four years before the Document itself, or if the original
   publisher of the version it refers to gives permission.
K. For any section Entitled "Acknowledgements" or "Dedications",
   Preserve the Title of the section, and preserve in the section all
   the substance and tone of each of the contributor acknowledgements
   and/or dedications given therein.
L. Preserve all the Invariant Sections of the Document,
   unaltered in their text and in their titles.  Section numbers
   or the equivalent are not considered part of the section titles.
M. Delete any section Entitled "Endorsements".  Such a section
   may not be included in the Modified Version.
N. Do not retitle any existing section to be Entitled "Endorsements"
   or to conflict in title with any Invariant Section.
O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or
appendices that qualify as Secondary Sections and contain no material
copied from the Document, you may at your option designate some or all
of these sections as invariant.  To do this, add their titles to the
list of Invariant Sections in the Modified Version's license notice.
These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains
nothing but endorsements of your Modified Version by various
parties--for example, statements of peer review or that the text has
been approved by an organization as the authoritative definition of a
standard.

You may add a passage of up to five words as a Front-Cover Text, and a
passage of up to 25 words as a Back-Cover Text, to the end of the list
of Cover Texts in the Modified Version.  Only one passage of
Front-Cover Text and one of Back-Cover Text may be added by (or
through arrangements made by) any one entity.  If the Document already
includes a cover text for the same cover, previously added by you or
by arrangement made by the same entity you are acting on behalf of,
you may not add another; but you may replace the old one, on explicit
permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License
give permission to use their names for publicity for or to assert or
imply endorsement of any Modified Version.


5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this
License, under the terms defined in section 4 above for modified
versions, provided that you include in the combination all of the
Invariant Sections of all of the original documents, unmodified, and
list them all as Invariant Sections of your combined work in its
license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and
multiple identical Invariant Sections may be replaced with a single
copy.  If there are multiple Invariant Sections with the same name but
different contents, make the title of each such section unique by
adding at the end of it, in parentheses, the name of the original
author or publisher of that section if known, or else a unique number.
Make the same adjustment to the section titles in the list of
Invariant Sections in the license notice of the combined work.
```

```
In the combination, you must combine any sections Entitled "History"
in the various original documents, forming one section Entitled
"History"; likewise combine any sections Entitled "Acknowledgements",
and any sections Entitled "Dedications".  You must delete all sections
Entitled "Endorsements".


6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents
released under this License, and replace the individual copies of this
License in the various documents with a single copy that is included in
the collection, provided that you follow the rules of this License for
verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute
it individually under this License, provided you insert a copy of this
License into the extracted document, and follow this License in all
other respects regarding verbatim copying of that document.


7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate
and independent documents or works, in or on a volume of a storage or
distribution medium, is called an "aggregate" if the copyright
resulting from the compilation is not used to limit the legal rights
of the compilation's users beyond what the individual works permit.
When the Document is included in an aggregate, this License does not
apply to the other works in the aggregate which are not themselves
derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these
copies of the Document, then if the Document is less than one half of
the entire aggregate, the Document's Cover Texts may be placed on
covers that bracket the Document within the aggregate, or the
electronic equivalent of covers if the Document is in electronic form.
Otherwise they must appear on printed covers that bracket the whole
aggregate.


8. TRANSLATION

Translation is considered a kind of modification, so you may
distribute translations of the Document under the terms of section 4.
Replacing Invariant Sections with translations requires special
permission from their copyright holders, but you may include
translations of some or all Invariant Sections in addition to the
original versions of these Invariant Sections.  You may include a
translation of this License, and all the license notices in the
Document, and any Warranty Disclaimers, provided that you also include
the original English version of this License and the original versions
of those notices and disclaimers.  In case of a disagreement between
the translation and the original version of this License or a notice
or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements",
"Dedications", or "History", the requirement (section 4) to Preserve
its Title (section 1) will typically require changing the actual
title.


9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except
as expressly provided for under this License.  Any other attempt to
copy, modify, sublicense or distribute the Document is void, and will
automatically terminate your rights under this License.  However,
parties who have received copies, or rights, from you under this
License will not have their licenses terminated so long as such
parties remain in full compliance.
```

```
10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions
of the GNU Free Documentation License from time to time.  Such new
versions will be similar in spirit to the present version, but may
differ in detail to address new problems or concerns.  See
http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number.
If the Document specifies that a particular numbered version of this
License "or any later version" applies to it, you have the option of
following the terms and conditions either of that specified version or
of any later version that has been published (not as a draft) by the
Free Software Foundation.  If the Document does not specify a version
number of this License, you may choose any version ever published (not
as a draft) by the Free Software Foundation.
```