traceroute

COLLABORATORS			
	TITLE : traceroute		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		December 8, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

traceroute

1.1 traceroute.guide

```
    traceroute - trace the route ip packets follow going to a host
    -Amiga AmiTCP/IP port-

 1.  Introduction                -        the purpose of this tool

 2.  Usage                       -        a brief overview of usage

 3.  about the Amiga port        -        some notes about the port

 4.  Copyright                   -        included materials
```

1.2 traceroute.guide/TR_INTRO

```
    Introduction
    ============

 traceroute is a networking utility written by Van Jacobson originally
for BSD Unix machines to find out the route ip packets follow,
including round-trip times for each packet, when going to a host of
your choice, which can be of great interest during troubleshooting a
network.

 For detailed information about the internal side, I recommend you to
read the introduction in the source file src/traceroute.c.
```

1.3 traceroute.guide/TR_USAGE

```
    Usage
    =====

 The basic usage is just

  > traceroute <host>
```

. The complete route to <host> will be printed to stdout, if it exists.
As an example, 'traceroute ftp.uni-paderborn.de' would have the
following output as a result (if no problems occur):

```
  traceroute to ftp.uni-paderborn.de (131.234.2.32), 30 hops max, 38 byte packets
   1  cisco.uni-trier.de (136.199.8.33)  8 ms   9 ms   7 ms
   2  LanWanGate.uni-trier.de (136.199.232.48)   7 ms   8 ms   7 ms
   3  cisco.Uni-Paderborn.DE (188.1.132.67)   188 ms   263 ms   272 ms
   4  athene.uni-paderborn.de (131.234.2.32)   172 ms   152 ms   163 ms
```

'30 hops max' means 'only the first 30 hosts the packets will be routed
via will be reported, the rest will be cut off'. Since there are always
3 packets transmitted, you'll get the time these packets needed from
your host to the host specified in the line, and then back, in 10^-6 second
units.

Since some kind of error might occur, I've added this (it never has
occured to, up to now :-)) example:

```
  traceroute to ftp.uni-regensburg.de (132.199.1.202), 30 hops max, 38 byte  ↩
     packets
   1  cisco.uni-trier.de (136.199.8.33)  8 ms   7 ms   5 ms
   2  LanWanGate.uni-trier.de (136.199.232.48)   5 ms   8 ms   6 ms
   3  * * *
   4  * * *
   5  * * *
  *** Break: traceroute
```

Whenever traceroute prints an asterisk (*), this is a sign of trouble,
since no response has been received on the packets your computer sent
out.

A last example, quite easy to understand:

```
  > traceroute cray501.eifel.de
  traceroute: unknown host cray501.eifel.de
```

The reason for this error is quite obvious, as any other errors, which
are reported that way, are.

Again, I tell you, this AmigaGuide document shall just be a brief
introduction in the traceroute program and is not intended as a
full-blown documentation. Even if you want to learn the rest of
traceroute's command line option, consult src/traceroute.c, in which
you'll find every command line option, and the methods traceroute
uses, explained.

On Unix systems, a man page for this program doesn't exist, either.

1.4 traceroute.guide/TR_AMIGAPORT

```
    about the Amiga port
    ====================

 this is just a dirty, quick, and somewhat hacked port just using a
source I've got nothing to do with :-)

 Well, it's not THAT bad:

   1) the original source code remains intact; just some #ifdef
      and a $VER: tag have been added

   2) 2 functions (I still think the Amiga port doesn't really need
      them) have been written to keep the origininally distributed
      source intact.


 To keep porting as simple as possible, I just linked the net.lib
developed by the AmiTCP-Group to my objects.

 BUT: there is a real nasty bug in net.lib: The SAS/C®
autoinitialization functions included in net.lib to open the
bsdsocket.library did not, if some error on startup occured, e.g.
AmiTCP hadn't been started before trying to run a program linked with
net.lib, return an error code to let the startup code call the
apporiate autotermination functions but just called exit(), which
left all allocated system resources hanging around as corpses in your
memory. The worst thing of all, the stdios weren't closed, and so
you would never have been able to close the Shell window you were
an AmiTCP utility running from.

 I fixed this particular bug; the fixed autoinit.c of net.lib is
included as socket_autoinit.c in the src directory this distribution.



 The Amiga port was done on a single afternoon (you see, porting
networking utilities that way is quite easy) by:

    Klaus Klein
    Kampbuechel 6
    54550 Daun, FRG
    +49-6592-2206

    e-mail:   klei0001@uni-trier.de


    used compiler: SAS/C® 6.51
```

1.5 traceroute.guide/TR_COPYRIGHT

```
    Copyright
    =========
```

traceroute uses the traceroute.c source from the BSD NET/2 distribution:

The included (but somewhat fixed) source file socket_autoinit.c, which appeared originally under the name autoinit.c as part of the official AmiTCP source distribution: