

**ETHLOAD 1.02**

**USER'S GUIDE**

**A simple public domain  
Ethernet load/problems analyzer  
and events tracer**

**E. Vyncke  
vyncke@csl.sni.be**

# **1. Introduction.**

ETHLOAD is a public domain software running on any MS-DOS PC with an Ethernet controller.

Currently, ETHLOAD supports the following drivers:

- Digital Equipment Corp. DLL specification;
- Microsoft 3Com NDIS (Network Driver Interface Specification);
- packet driver as issued from PC/TCP, Clarkson University or from the Crynwr collection;
- Novell ODI (Open Datalink Interface) iff the driver supports promiscuous mode.

The purpose of ETHLOAD is threefold:

- display very simply non accurate numbers about the Ethernet load (number of frames/sec, bits/sec, ...);
- display important parameters, events and loads for the DECnet protocol;
- display important parameters, events and loads for the TCP/IP protocols suite.

ETHLOAD allows you to:

- check simply the load of your Ethernet (with error rate, interframe gap,...);
  - check which host is sending most of frames;
  - see which host is sending to wich host;
  - see what kind of protocols are in use in your Ethernet;
  - ...

In a TCP/IP network, ETHLOAD allows you to:

- see ARP table contents;
  - see which host is sending (un)resolved ARP probes;
  - see the IP host which is sending most of the IP, UDP or TCP packets;
  - see what kind of protocols are in used (either TCP or UDP);
  - see which is the mostly used telnet/rlogin server (or client);
  - see the boot sequence with important BOOTP and TFTP events;
  - see some characteristics of IP hosts (fragments size, MTU, IP retransmission,...);
  - see important TCP events: start/stop of connections,...

In a DECnet network, ETHLOAD allows you to:

- see which node are sending/receiving most of DECnet packets;
  - see all Connect Initiate packets (including object number, ...);
  - see returned packets;
  - ...

In an OSI network, ETHLOAD allows you to:

- see the top transmitter/receiver NSAP (for inactive network layer and should also work with active network layer);
- see important events for the transport layer: connection/disconnection, TSAP are displayed in hexa, ASCII and EBCDIC.

\* \* \*  
\* \*  
\*

## **2. Acknowledgments.**

### **2.1. Original copyright.**

This software is based on the very first version of ETHLOAD I have developed while I was working in a company called Network Research Belgium. This version was already in the public domain thanks to the management of this company.

Here follows the copyright included in the source files of about 10% of the current version of ETHLOAD.

/\* This software and documentation can be copied, used, modified freely as long as:

- the source contains this text

- this software, documentation is provided free of charge (but for the cost of media: paper, CD-ROM, ...).

Network Research Belgium and the individuals who have written this software DO NOT ASSUME any responsibilities in respect to the use, (un)expected side -effects of this program.

The software and documentation is provided as it is. No maintenance will be given.

Anyway, we would be pleased to hear of any use of these softwares by email, fax or phone:

bert@nrb.be

fax: +32.41.48.11.70

phone: +32.41.40.72.11 ask for a BERT member.

Suggestions, modifications are always welcome.

These softwares have been developed by a special team called BERT in a company called Network Research Belgium located in Herstal, Belgium, Europe .

This team includes:

Eric Vyncke, vyncke@nrb.be now vyncke@csl.sni.be

Frederic Blondiau, blondiau@nrb.be

Michel Ghys, now mghys@cisco.com

Marie-Christine Timmermans, timmermans@nrb.be

Jean Hotterbeex, now working in Trasys with no email

Manu Khronis, khronis@nrb.be

Vincent Keunen, keunen@nrb.be

\*/

### **2.2. Current copyright and disclaimer.**

Right now, all software developments is made home and tested after working hours in my current company: Siemens Nixdorf Informationsystems, SNI. So, here follows the usual disclaimer: Siemens Nixdorf is by no means responsible for any good or bad effects of this program. And by the way, the quality of ETHLOAD does not reflect the usual quality of NRB or SNI software.

NRB, Siemens Nixdorf and the author do not support this software.

## 2.3. Support.

Anyway, you can get some support from the author since he wants to promote this software... You can reach the author through email: [vyncke@csl.sni.be](mailto:vyncke@csl.sni.be)<sup>1</sup> or by post mail:

Eric Vyncke  
Rue Nolden, 25  
B-4432 Alleur  
Belgium.

If you are happy with ETHLOAD, my little son, Pierre, would appreciate to receive any postcard!

## 2.4. Distribution channel.

I have no access to internet, so I cannot place ETHLOAD on anonymous FTP server, if you run such a server I will appreciate that you reserved some place for ETHLOAD...

## 2.5. Thanks to testers.

I would like to thank anyone of you about his/her comments.

I thank especially my beta-testers:

Ralf Buettmeyer, [buettmeyer@hagenuk.netuse.de](mailto:buettmeyer@hagenuk.netuse.de)  
Michel Dalle, [michel@d92.cb.sni.be](mailto:michel@d92.cb.sni.be)  
Niels Kr. Jensen, [msterlje@vm.uni-c.dk](mailto:msterlje@vm.uni-c.dk)  
Hans-Joachim Koch, [koch@lifra.lif.de](mailto:koch@lifra.lif.de)  
Frank Van Uffelen, [frankvu@bix.com](mailto:frankvu@bix.com)

I thank also for comments, suggestions, ...:  
Thomas Gasser, [thomasg@staff.tc.umn.edu](mailto:thomasg@staff.tc.umn.edu)  
Derek Johnston, [ugcsjj9697@mtvms2.mtech.edu](mailto:ugcsjj9697@mtvms2.mtech.edu)  
Ross Lazarus, [rossl@westmead.health.su.oz.au](mailto:rossl@westmead.health.su.oz.au)  
Russ Nelson, [nelson@crynwr.com](mailto:nelson@crynwr.com)

And, all of you who have send a postcard :-)

## 2.6. Changes.

---

<sup>1</sup>email in Belgium is not free :-( So that's my employeer which pays any email. If any site in Belgium or BITnet is wishing to start-up a distribution list for ETHLOAD, I would really appreciate ;-)  
I should also get very soon a Fidonet address.

## 1.01:

- support for packet driver, ODI and NDIS
- support for TCP/IP
- no more load graphics
- dictionaries
- bug correction in the length display
- porting from large model in Borland C to small model in Borland C++

## 1.02:

- bug correction in DLL support
- documentation about copyright on packet drivers
- dropped packets percentage in MAC screen
- MAC flow screen
- SMTP, TFTP and BOOTP support
- Telnet/rlogin monitoring
- options in command line
- OSI support
- improved DLL, ODI, NDIS and packet driver routines

## 2.7. Trademarks.

As usual, all trademarks (Ethernet, DEC, ...) are properties of their respective owners.

## 2.8. Source code.

Source code for the version of NRB can be obtained from bert@nrp.be. I still have a copy of the source files, so if there is a 'large' demand I could put them on some anon FTP server.

\* \* \*  
\* \*  
\*

### **3. Configuration files.**

In order to run in basic mode (i.e. without translation of addresses into names,...) ETHLOAD does not require any configuration file. The configurations are required only if you want to achieve good printings: host name instead of addresses, ...

All configuration files are in the same format:

- plain ASCII files, i.e. lines ended by CR/LF;
- any line beginning with a ';' or a '#' is considered as a comment;
- empty lines are ignored;
- other lines must begin with a token generally numeric, called the key, then a serie of space or TAB characters, followed by another token, called the value. The value token is ended by the CR/LF end of line.

Most of these files are the MS-DOS image of the well known TCP/IP files for Unix: /etc/hosts, /etc/ethers, /etc/protocols, ... The simplest way to use them is to FTP them from your Unix box.

If you are using TCP/IP you should FTP /etc/hosts of a Unix host and perhaps add some MAC addresses to the ETHERS file.

If you are using DECnet, you probably don't need to modify any of these files.

If you are using another protocol, you will probably need to modify ETHERS file together with TYPES and/or SAPS.

All these optional files must be located in the current directory of the current drive or in the directory specified by the MS-DOS environment variable ETHLOAD.

### **ETHERS**

This file contains the mapping between MAC Ethernet addresses into host names.

The key token is the Ethernet MAC address in the format HH-HH-HH-HH-HH-HH where HH is a pair of hexadecimal digits.

The value token is any character string representing the name of this host.

Part of ETHERS file:

139.24.16.44	baumann
--------------	---------

Remark: ETHLOAD is smart enough to recognize a DECnet node and display the DECnet address of any MAC address. If you want to display DECnet address by node name, you may use the MKNODE.EXE program documented in annex A.3.

Remark 2: ETHLOAD is also listening for ARP requests and replies, so it can display the IP address of any MAC address.

Remark 3: ETHLOAD as it is (i.e. without ETHERS) cannot even display correctly well known address as the null address or even the broadcast address.

Remark 4: you should add your own MAC addresses only if you are not using DECnet or TCP/IP, moreover, you should add these addresses at the end of ETHERS file and keep the original contents of ETHERS.

## HOSTS

This file contains the mapping between IP address and host names.

The key token is an IP address in the format ddd.ddd.ddd.ddd where ddd is up to three decimal digits.

The value token is any character string representing the name of this host.

Part of HOSTS file:

The best way to initiate this file is to get a /etc/hosts from a Unix machine (or the stdout of the `ypcat hosts.byaddr` if you are running NIS<sup>1</sup>).

## PROTOCOL

This file contains the mapping between IP protocols and protocol names.

The key token is a decimal number up to 255.

The value token is any character string representing the name of the protocol.

---

<sup>1</sup>Also known previously by Yellow Pages.



0	80p 30ger XNS
1	81c mpPp way-LAN
3	AGgp, gGGMIB SNAP (Ethernet type in LLC)
6	81c 6 Bafy mXINES
8	80p, NongidshgkWay protocol
12	FQup IBM NetBIOS
17	udp
20	hmp, host monitoring protocol
22	xns-idp
27	rdp, reliable datagram protocol

One again, the best way to initiate this file is to get /etc/protocols from a Unix machine or using the PROTOCOL file you may have receive with ETHLOAD. The first solution is probably not useful since /etc/protocols are always nearly the same.

The shipped PROTOCOL file contains:

## SAPS

This file contains the mapping between IEEE 802.2 LLC SAP and SAP names.

The key token is two hexadecimal digits.

The value token is the name representing the Service Access Point.

Part of a sample SAPS file:

Remark: ETHLOAD has a built-in knowledge of SNAP.

## WKS.TCP (resp. WKS.UDP)

This file contains the mapping of TCP (resp. UDP) well-known services ports.

The key token is a decimal number up to 65535 which is the port number assigned to the service.

Part of a sample WKS.TCP file:

This file together with WKS.UDP contains all the information of the usual /etc/services Unix file but in a slighty different format.

Since the file /etc/services is always the same on all Unix machine, you may probably

0000-00NSCisco
0000-00NSNATress Translation
0000-100Syttek
0000-10.75Cabletron

use the files provided with ETHLOAD.

## TYPES

This file contains the mapping of the DIX Ethernet packet type into names.

The key token is 4 hexadecimal digits.

Part of a sample TYPES file:

## VENDORS

This file contains the mapping between the IEEE vendor codes and the vendor names. The IEEE vendor code is representing the most significant three bytes of the MAC address of any adapter built by this manufacturer.

The key token is 3 bytes represented each by two hexadecimal digits, each byte is separated by a dash.

Part of a sample VENDORS file:

```
* * *  
* *  
*
```

## **4. Set-up of datalink drivers.**

ETHLOAD as already said is currently running as it is on the top of four different datalink drivers. ETHLOAD automatically configures itself to use the first driver found. It tries in the following order:

- Novell ODI;
- Microsoft 3Com NDIS version 2.0.1 or higher;
- Digital Equipment DLL;
- PC/TCP packet driver.

If you use another driver and you have a specification of its API (or even some C routines in the public domain), please email me because I would like that ETHLOAD runs on nearly all datalink drivers... ;-)

If this order does not work for you, you will have to use the -d option in the command line for starting ETHLOAD (see section 5).

Some of these datalink drivers allow for simultaneous execution of ETHLOAD and of your usual protocol stack: NDIS and ODI. All other drivers prevent the execution of your usual protocol stack, it means that you will abort all current connections to any servers.

Some of these datalink drivers do not require a PC reboot after running them: DLL, NDIS version 2.0 or higher, packet driver and ODI.

Finally, only one kind of drivers namely ODI allows for the identification of faulty frame by their source or destination addresses.

In conclusion, if your Ethernet hardware has a ODI driver with promiscuous mode support, it is better to use ODI.

A final remark, packet driver does not differentiate between the various kind of errors in its statistics. So, you should use any other driver if possible.

### **4.1. Novell ODI.**

The first thing to note is that only very few ODI drivers supports the promiscuous mode which is needed for ETHLOAD. Novell has a list of those drivers since the promiscuous mode is also needed by Novell LANalyzer product.

To use ETHLOAD, you just have to load the ODI driver (preceeded as usual by LSL.COM) and having a correct NET.CFG. If you can run any other ODI application (Novell LAN Workplace for DOS, Siemens Nixdorf LAN 1, ...), you should be able to run ETHLOAD as it is.

---

<sup>1</sup>The version 1.0.1 is also supported, but with several restrictions (see further)...

```
[ETHLOAD]
```

```
  drivename = ETHLOAD$  
  bindings = MYMAC
```

The use of ETHLOAD is not disruptive to your other network application which will continue to run at very bad efficiency...

To start ETHLOAD, just issue the ETHLOAD command to the MS-DOS prompt.

## 4.2. Microsoft 3Com NDIS v 1.0.1.

Before running ETHLOAD for the first time, you must modify your PROTOCOL.INI (usually located as C:\LANMAN\PROTOCOL.INI see your C:\CONFIG.SYS file and the DEVICE=..PROTMAN... /I:<path>).

You must add the following lines in your PROTOCOL.INI (anywhere in the file but after a section):

where MYMAC is the name of the MAC module you want to use.

These modifications do not modify the usual behaviour of your PC, so you may leave these lines in your PROTOCOL.INI file even if you don't use ETHLOAD.

After you have made these changes, you must reboot your PC.

After this reboot, when you want to use ETHLOAD you must issue the ETHLOAD command to the MS-DOS prompt.

By the way, the Protocol Manager directory (containing NETBIND.EXE, ...) should be in the PATH of MS-DOS.

Remark 1: in PROTOCOL.INI the case of the left part of '=' does not matter, but uppercase characters must be used on the right part as indicated in the examples above.

Remark 2: as you are using a version of Protocol Manager older than version 2.0.1<sup>1</sup>, ETHLOAD will display some warnings and you have to pay special attention to the following points:

- don't run NETBIND.EXE before ETHLOAD (so look out in your AUTOEXEC.BAT for an automatic run of NETBIND.EXE)
- reboot your PC after running ETHLOAD since Protocol Manager cannot be reset in a correct state

---

<sup>1</sup>You can check the version by looking at the banner displayed when Protocol Manager is loaded from CONFIG.SYS. Also, if the Protocol Manager directory is missing the PROTMAN.EXE file, you can bet you have a old 1.0 version.

<pre>[<del>ETHLOAD</del>] MANAGER] deviceName = <del>ETHLOAD</del>\$ bindings = MYMAC bindstatus = YES priority = ETHLOAD</pre>
---

- some statistics are missing.

### 4.3. Microsoft 3Com NDIS v2.0.1 or higher.

Before running ETHLOAD for the first time, you must modify your PROTOCOL.INI (usually located as C:\LANMAN\PROTOCOL.INI see your C:\CONFIG.SYS file and the DEVICE=..\PROTMAN... /I:<path>).

You must add the following lines in your PROTOCOL.INI (anywhere, after a section):

where MYMAC is the name of the MAC module you want to use.

You also have to modify the [PROTOCOL MANAGER] entry to add a dynamic line. But first try without this modification before modifying further your PROTOCOL.INI file.

These modifications do not modify the usual behaviour of your PC, so you may leave these lines in your PROTOCOL.INI file even if you don't use ETHLOAD<sup>1</sup>.

After you have made these changes, you must reboot your PC.

After this reboot, when you want to use ETHLOAD you must issue the ETHLOAD command to the MS-DOS prompt.

By the way, the Protocol Manager directory (containing NETBIND, ...) should be in the PATH of MS-DOS.

Remark 1: in PROTOCOL.INI the case of the left part of '=' does not matter, but uppercase characters must be used on the right part as indicated in the examples above.

Remark 2: the use of ETHLOAD should not be disruptive for your favorite protocol stacks, so you should not have to reboot your PC.

---

<sup>1</sup>But for the bindstatus=YES, which increase the resident part of the Protocol Manager, thus, reducing the available base memory. If you are concerned with base memory, you may instead use bindstatus=NO, then ETHLOAD won't be able to display some informations about Protocol Manager but will anyway work as usual.

## 4.4. Digital Equipment DLL.

If DLL.EXE (or DLLDEPCA.EXE) is already loaded, you have nothing to do before starting ETHLOAD by the ETHLOAD command.

Note: in order to go promiscuous, DLL requires that ETHLOAD shutdown ALL connections: LAT, DECnet, ... After using ETHLOAD you probably will have to reset the whole DECnet protocol stack (so reboot your PC).

Note2: it seems that at least for version 4.1 of DLL, it is impossible to run ETHLOAD in a DOS box within MS-Windows 3.1.

## 4.5. Packet driver.

Packet drivers exist for nearly all known Ethernet adapters. There even exists 'packet driver shim' that transform some other datalink drivers into a packet driver.

You have to use a software interrupt between 0x60 and 0x7F in order to let ETHLOAD run.

ETHLOAD will use the first packet driver found while checking from interrupt 0x60 up to 0x7F.

The use of ETHLOAD is not disruptive to your other network application which will continue to run at very bad efficiency...

To start ETHLOAD, just issue the ETHLOAD command to the MS-DOS prompt.

Remark: nearly all packet drivers can be found in numerous anonymous FTP server including SIMTEL20.ARMY.MIL. For BITnet users, they can also be fetched through TRICKLE server. The Crynwr Packet Driver Collection is copyrighted using the GNU General Public License.

\* \* \*  
\* \*  
\*

## **5. Command line options.**

In nearly all configurations, ETHLOAD can be started without specifying command line options. In some case, you may need to use these command lines options: special datalink drivers configuration, few memory left, ...

Command line option can be specified in either the Unix shell format:

ETHLOAD -do1 -i65 -t

or in the MS-DOS format:

ETHLOAD /D:O1 /I:65 /T

Case does not matter.

### **5.1. Datalink driver: -d**

ETHLOAD can be forced to use a special datalink driver instead of trying to find automatically the best one.

To use Novell ODI, specify: -do or /D:O

To use Novell ODI with the MLID board 3, specify: -do3 or /D:O3

To use Microsoft/3Com NDIS, specify: -dn or /D:N

To use Digital Equipment DLL, specify: -dd or /D:D

To use Packet driver at first interrupt found between 0x60 and 0x80, specify: -dp or /D:P

To use Packet dirver at interrupt 0xHH, specify: -dphh or /D:PHH.

### **5.2. Protocols to be analyzed: -p**

ETHLOAD by default analyzes all protocols. This requires both more memory and more processing than analyzing a single protocol. By using the -p option, you can restrict the protocols to be analyzed by ETHLOAD.

To analyze DECnet, specify d after the -p.

To analyze the TCP/IP protocol suite, specify i after the -p.

To analyze the OSI protocol suite, specify o after the -p.

### **5.3. Real time frame trace: -t**

ETHLOAD can display the very first bytes of all received frames in real time on the bottom line of the display.

This behaviour is set by using the -t option on the command line.

Remark: in version 1.01, ETHLOAD always displayed the first bytes of the packet.

#### **5.4. Faster/Unsecure mode: -f**

ETHLOAD can work in fast mode with packet driver and ODI.

The fast mode is not set by default.

The secure (the default) is defined as disabled IRQ while a frame is analyzed. The advantage is that the stack of the datalink driver is not overloaded, but, the big drawback is that a lot of frames may be either dropped or even ignored.

By using this option, ETHLOAD can see much more packets but may sometimes runs into problems...

So, this option should be set ONLY if you encounter no problems with ETHLOAD (PC that hangs, inconsistent display, ...) and you have a high percentage of lost packets.

#### **5.5. Measure interval: -i**

ETHLOAD measures the load of the LAN at regular interval, the screen is also automatically refreshed at the same rate.

By default, this interval is 5 seconds. You may select another measure/screen refresh interval by using the -i option followed by the number of seconds.

```
* * *  
* *  
*
```



## 6. The different screens of ETHLOAD

### 6.1. Introduction

#### 6.1.1. Screen layout

The different screens displayed by ETHLOAD have all the same design:

- the top line is just a copyright notice + version identification + percentage of dropped frames due to internal buffer shortage (either in ETHLOAD or in data link driver or even in Ethernet controller);
- in the top right corner a character is flipping from '+' to '-' as frames are received;
- the character on the left of the '+/-' flip-flop is displayed as a 'P' when ETHLOAD is processing a frame else it is a space;
- the second line is a summary of all commands available for this screen;
- if the real time trace option was specified in the command line, the bottom line displays the first bytes of the last received frame:
  - \* six bytes of MAC destination address ;
  - \* six bytes of MAC source address ;
  - \* two byte(s) for either DIX packet type or for IEEE 802.3 frame length;
  - \* a few bytes of data.

All screens are automatically refreshed every measure interval (5 seconds by default) to reflect the current statistics or table contents. You may also press the SPACE key to refresh the screen.

#### 6.1.2. Commands.

You can enter a single character command. The case of the character is ignored.

Two commands are always recognized:

- 'Z' or '0': for resetting all statistics of ETHLOAD to zero and clearing all tables. Note that all statistics are cleared and not only the ones currently displayed;
- 'X' or <ESC>: for leaving the current screen and getting back to the previous menu.

On some screens a large table is displayed: ARP table, ... As these tables are larger than the 23 lines of display available, you have to use the **PgUp** and **PgDn** key to scroll between the different pages.

---

<sup>1</sup>This display together with the '+/-' flip-flop is only displayed by memory mapped IO on colour displays.

### 6.1.3. Data display.

Three common display are often used:

- top of sorted table display;
- raw table display;
- history of events display.

The '**top display**' consists of a title beginning with 'Top of...' and displays the contents of an internal table sorted from the highest frequency down to the lowest frequency. An example of such a display is the display of MAC Transmitter. A reference is also displayed by indicating how many frames represents 100%. Please not that %age are given with respect to the number of frames and not with respect to the number of bytes.

As all counters are 32 bits, they are limited to about 4E+9 frames. Once they reach this upper bound they are stopped and the whole table is kept unchanged. The time of this table overflow is then displayed in red.

Each line of a '**top display**' consists of:

- percentage (e.g. the percentage of Ethernet frames transmitted by the displayed Ethernet node in respect to the total number of Ethernet frames);
- display of the node (e.g. Ethernet MAC address with perhaps the corresponding host name of DECnet address);
- a bar graph for visual representation (resolution 2.5%).

The '**raw table display**' is just the display of a non sorted internal table. An example is the display of the ARP table.

Each line of a '**raw table display**' consists of two values (e.g. the Ethernet MAC address associated with an IP address).

The '**event history**' is used to display a chronological log of events (e.g. the list of ICMP requests).

Each line of an '**event history**' consists of:

- a time stamp in the form hh:mm:ss.hh;
- a description of the event.

### 6.1.4. Accuracy

A final remark must be done on the accuracy of the figures:

- some packets are lost, so the load is always higher than indicated if you are using a slow Ethernet controller or a non efficient driver;
- ETHLOAD relies on the MS-DOS timer which has a resolution of about 50 msec, moreover if the network load is high and you have a powerless CPU some timer ticks can be missed;

- if you are running with IRQ disabled (i.e. without the -f option), some datalink drivers can miss frames without further notification, so the drop percentage is always higher than the one displayed by ETHLOAD.

To summarize, ETHLOAD give reliable figure on a medium loaded Ethernet (10% ?) and on a correct CPU 80386dx 25 MHz. In all other case, ETHLOAD can only indicate that your Ethernet is probably heavily loaded and you will have to buy an expensive LAN analyzer!

## 6.2. MAC Level screen

The MAC level screen can be divided into two parts:

- three statistics summaries: last five seconds, busiest five seconds, cumulative;
- VU-meter of the peak and current load.

### 6.2.1. MAC Summary

Important figures are displayed for three important samples:

- the last five seconds;
- the busiest five seconds, i.e. the five seconds period when the Ethernet load was the highest ;
- the cumulative since the start of ETHLOAD or the last reset.

For all these samples, the following figures are displayed:

- total number of Ethernet frames: the mean interframe gap is also displayed if available;
- total number of bytes of data: i.e. MAC header + MAC data (the FCS and preamble is not taken into account) and the load of Ethernet in % of the 10 Mbps bandwidth of Ethernet;
- the number of frames containing errors + rate of error per second.

As the internal counters are 32 bits, counters are bounded to about 4E+9 frames/bytes. Once the counters reach this count; they are stopped and displayed as \*\*\*\*\*.

If the datalink driver supports error differentiation (namely all but packet driver), the kind of error is also indicated:

- CRC error (cabling problem ?);
- too long packet (babbling transceiver or controller);
- too short packet (garbage of collision).

If you are using the ODI datalink driver, by using the 'E' command you have access to the MAC source address of faulty Ethernet frames (by the way don't be amazed by unknown MAC addresses because even the source address can be faulty in faulty frames... specially for runt frames).

## 6.2.2. MAC VU-meter

The VU-meter is at the bottom of the screen and is graduated in Mbps.

The '>' is the peak marker, i.e. the highest load on five seconds since ETHLOAD has been started or reset.

The bar is the last five seconds marker.

The color of the peak marker and of the bar is changing in respect to the load:

- green under 1 Mbps;
- yellow under 5 Mbps;
- red over 5 Mbps.

## 6.2.3. MAC Commands

The MAC level screen has four main commands:

- '**Q**' to quit ETHLOAD and get back to MS-DOS (a confirmation is requested);
- '**D**' to go to the DECnet screens ;
- '**O**' to go to the OSI screens ;
- '**T**' to go to the TCP/IP screens.

## 6.3. TCP/IP screens

to be added if you ask me by email...

In very short, you can display:

- **ARP**: table of the mapping between IP addresses and MAC addresses (can be used to detect two hosts sharing the same IP address), the last ARP packet, the ARP senders, the requested IP addresses;
- the **IP fragmenters** and the size of fragments, i.e. the IP host that transmit fragmented datagram (should be empty !);
- important information about **IP hosts**: largest MTU (Maximum Transmit Unit) seen, missing IP datagrams (should be zero if host is on the same LAN and has only one interface), repeated IP datagrams (could indicate faulty transceiver or SQE test enabled were it shouldn't), minimum and maximum TTL (Time To Live) seen from this host;
- **ICMP**: the last ICMP datagrams, the senders of ICMP datagrams;
- mostly used **protocols**: UDP, TCP, ...
- **TCP**: events (connection request, end of connection), connections, most used services (ports), important events for SMTP and POP, monitoring Telnet connections, ...
- **UDP**: associations, most used services (ports), important events for BOOTP and

TFTP,...

## 6.4. DECnet screens

to be added if you ask me by email...

In very short, you can display:

- **Connect Initiate** (with nearly all fields including objects,...) history;
- **Disconnect Initiate** history;
- **Returned** frames by a router because the end-node is no more reachable;
- **Top nodes** (classified by transmitters and receivers): not to be confused with the MAC layer transmitters/receivers. On the MAC screens, DECnet routers usually represent a very high percentage but on the DECnet network layer screen, DECnet routers usually represent nothing and you can see remote DECnet address (i.e. some DECnet nodes on remote LAN).

## 6.5. OSI screens

to be added if you ask me by email...

In very short, you can display:

- the **Active** network layer hosts (not tested, if it runs please email me ;-)
- the **Inactive** network layer hosts;
- the most important **Transport** layer events: connection, disconnection, error. NSAP are displayed in hexadecimal and TSAP are displayed in hexa, ASCII and EBCDIC. Important parameters are decoded and displayed.

```
* * *  
* *  
*
```

## **A. Annexes**

### **A.1. Data Link layer references**

**Digital Equipment**, '*PCSA Data Link Programmer's Reference Manual*', April 1989, EK-PCDLL-PR-001

**FTP Software**, '*PC/TCP Packet Driver Specification*', Revision 1.09, September 1989

**3Com/Microsoft**, '*LAN Manager Network Driver Interface Specification*', Version 2.0.1, October 1990

**Novell**, '*Open Data-Link Interface - Developer's Guide for DOS Workstation Protocol Stacks*', Version 1.10, March 1992

### **A.2. Tested data links**

Here follows a very short and not restrictive list of tested datalinks:

- Protocol Manager 2.01 + Cogent LP486E NDIS driver;
- SMC 8003, packet driver 8003PKDR V2.03;
- SMC 8003, ODI promiscuous mode SMC8000 V3.03 (920925) and LSL 1.0 (900530);
- EXOS205 V 10.1.2, packet driver;
- DEPCA and DE100 with version 4.1 of DLLDEPCA;

If you can run ETHLOAD on other drivers or even on IEEE 802.5 or 802.6 LAN, please email me in order to increase the size of tested datalink drivers.

### **A.3. Adding DECnet node names to display.**

A utility program provided with ETHLOAD, MKNODE, allows to display DECnet node names after DECnet address.

MKNODE simply converts DECnet addresses in the form of area.node (e.g. 1.1) into Ethernet address in the form of AA-00-04-00-xx-yy (e.g. AA-00-04-00-01-04).

MKNODE is a MS-DOS filter program, i.e. it takes input from the stdin and its output is stdout. The usual way of using MKNODE is:

- 1) get the list of DECnet node addresses and names (e.g. by running \$ NCP SHOW KNOWN NODES TO nodes on a VAX/VMS) in a MS-DOS called NODES. The format of this file is:

```

#
# The list of DECnet addresses are built with MKNODE.EXE
#
# (c) vyncke@csl.sni.be
# Can be copied and used freely
# 1.76 MDCPC
# Input DSRV03 and consists of line in the format
# 2.4 area.node nodename

```

```

#
# Output is stdout and should be appended to ETHERS
#
# Run of Sun Jul 11 10:18:32 1993
#
#
# 1.1 RM
AA-00-04-00-01-04 RM
# 1.76 MDCPC
AA-00-04-00-4C-04 MDCPC
# 2.3 DSRV03
AA-00-04-00-03-08 DSRV03
# 2.4 DSRV04
AA-00-04-00-04-08 DSRV04

```

- area.node name
- 2) on MS-DOS, issue the command:  
MKNODE < NODES >> ETHERS
- 3) that's done !

Here is an example for the file NODES:

And here is the added lines in ETHERS:

**Remark:** I'm not really satisfied with this two-step procedure. If you have written any VMS/DCL procedure that has the same result and you wish to put this procedure into the public domain, I would be pleased to include it in the distribution kit of ETHLOAD.

```

* * *
* *
*

```