# The Beholder

The SNMP-able Ethernet Monitor

By the DNPAP development group
JPMvOorschot@et.tudelft.nl
5/7/91

# 1. The Beholder

The Beholder is a software-only product that implements a ethernet network monitor on standard PC hardware. The data collected can be obtained in three different ways, by looking at the PC screen, by requesting the data as SNMP variables, and by using TFTP to collect files with data. A standard ethernet Local Area Network (LAN) can contain several Beholder monitoring stations, each containing several network interfaces. Normal use will be to collect the data of all present Beholder stations to the network management node via SNMP. This data can then be processed to obtain problem reports, growth figures and performance measurements.

The Beholder was developed by the Data Network Performance Analysis Group (DNPAP) of the Delft University of Technology. It is used as an important data collector in the "Intelligent Network Management (INEMA)" project. This project seeks to apply automated reasoning techniques to network managements.

Main design goals of The Beholder were:

- Minimum loss of packets
- Continues operation
- Appliance to Standards wherever possible.
- Ease of Use

The result of the developments is a PC based software package, capable of monitoring all traffic on one or more ethernet segments. The Beholder can be easily integrated in a SNMP based network management environment like Sun Net Manager.

## 1.1. Hardware and Software

The Beholder software runs on a standard 80<x>86 based Personal Computer, containing a network interface for which a "packet driver" network device driver is available. The Beholder was developed in ANSI C using the Microsoft C 6.0 compiler and two very small assembler files. For busy ethernet network, the PC should be at least a 80286 at 10 Mhz, but a 80386 at 20 Mhz to be save.

The WD 8003 family of ethernet cards is the preferred choice for ethernet network interface, but the 3COM line and the Novell NI1000/2000 will work fine.
If the Beholder is used as monitoring station in combination with a network management station that collects the measurement data, no keyboard, mouse or display are needed as far as The Beholder is concerned. If The Beholder is used stand-alone, a standard PC display (colour is nice), keyboard and mouse can be used to view the results of the monitoring PC. The Beholder uses no graphics.

## 1.2. Installation

### 1.2.1. Hardware

For installation of the hardware needed, the PC and the ethernet board, see the documentation that came with those products. The most common parameters that have to be set are:

- I/O address: (like: 0x280)
- RAM address: (like: 0xd000)
- IRQ: (like: 0x03)

These parameters have to be chosen for the ethernet board so that there is no conflict with other hardware in the PC like the disk controller, the VGA video adapter or other build-in hardware. The best way to check the validity of a set of values is to start a well known network product, to use the test software that comes with the "packet-driver" set, to use the diagnostics tools that comes with the ethernet card, or to just start The Beholder and see what happens. Remember, an ethernet is never silent for more then a few seconds in other then test environments.

### 1.2.2. Software

The installation of the software consists of several steps:

- copy beholder software to the desired directory on hard-disk or floppy-disk.
- configure the beholder by editing BEHOLDER.INI
- edit BEHOLDER.BAT to start the correct packet driver
- start The Beholder

Note that no other network software must use the same ethernet card as The Beholder is using.

### 1.2.3. Beholder.ini

Beholder.ini is the configuration file of The Beholder. It is somewhat like the Microsoft .INI files found in Windows, OS/2 and other software packages.

The file is split in several sections, each section headed by a line:

[*section-name*]

Each section contains text lines with configuration information. The format depends on the section in which the line is in. Comments can be inserted by preceding the comment by the '#' or the ';' character.

See appendix A for a detailed description of the parameters in Beholder.ini. For a quick start, follow the following guidelines.

You should edit Beholder.ini using a standard asci editor. Change only the following parameters:

Section [BUFFER]
numberbuffers = 5 # lower this to 4 if your PC doesn't have enough
# memory
buffersize = 65500

Section [IPDOS]

nd0address = <*your-ip-address*>

Section [ROUTES]
hoststat default <*your-ip-gateway*>
hoststat 127.0.0.1 127.0.0.1
hoststat <*your-ip-address*> 127.0.0.1
netstat <*your-ip-net*> <*your-ip-address*>

Section [SYSTEM]
Description = <*Description-of-your-monitoring-pc*>
Contact = < *Name-and-telephone-of-contact-person* >
Name = < *Name-of-monitoring-pc* >
Location = < *description-of-location-of-monitoring-pc* >

Section [AUTHENTICATION]

Community public
AddAddress <*mgmnt-station-ip-address*> <*and-mask*>
{ AddAddress < *your-ip-address* > <*next-and-mask*>}

Community trap
AddAddress <*mgmnt-station-ip-address*> <*and-mask*>
{ AddAddress < *your-ip-address* > <*next-and-mask*>}

Section [AGENT]

TrapAddress = <*mgmnt-station-ip-address*>

You can look at the file Beholder.xmp for the values the DNPAP group uses.

Notice that The Beholder needs its own IP-address. The Beholder will use the first ethernet-card it finds as its output port for UDP/IP traffic.You will also have to determine which community you want to use, and which IP hosts are allowed to collect the measurement data.

### 1.2.4. Beholder.bat

The Beholder.bat file loads the packet driver(s) and starts The Beholder. After The Beholder is finished, the packet drivers(s) is removed.

You should adjust the Beholder.bat file to fit your ethernet card and packet driver. The packet driver should be from the 8.x distribution. Older version will probably work, but the 8.x version and later are the only ones tested.

Look at the Beholder.bat in the distribution for an example of a Western Digital WD8003 driver loaded on IRQ 0x3, IO/address 0x280 and RAM address 0xd000.

The beholder will find all the packet drivers that are activated in the system and use them for monitoring purposes.

# 2. The Beholder Output

The Beholder has three main method of presenting the measurement results, the screen, SNMP variables and TFTP-able files. The last two methods require you to have a network management workstation with a TCP/IP stack and SNMP capabilities. By use of the screen, the monitoring PC can be used as a stand-alone tool. This is not the standard mode of operation, but if it is all you have, use it.

## 2.1. Screen Output

When you start The Beholder, the screen is filled with a window-based representation of the activity on your network. There are four major windows, the network load, the ethernet-type distribution, the packet-length distribution and a status window. By using the <SHIFT-F4> key, you can position and arrange the windows. By using the <ESC> key, you can start, stop and reset applications. During normal operation, you won't need this keys. If you have more then one ethernet card in you monitoring station, you can switch between these cards pressing <0>, <1> ....

## 2.2. SNMP Output

The Beholder has a full featured SNMP interface. It can report all its findings using an extension to the standard MIB2 database. The variables are defined in the file "Beholder.snm". The SNMP interface present the data as variables named by a ASN.1 number. These variables can be requested through the UDP/IP network protocol. These requests are normally generated by a network management station.

## 2.3. TFTP-able File Output

The current version of The Beholder has a limited capability of dumping data to files. The only files that can be retrieved in this version are the debug- and the error files. In future versions of The Beholder, a source/destination matrix and packet-trace files can be generated and collected. TFTP is a file transfer protocol of the TCP/IP suite and is implemented by every TCP/IP implementation known to us.

The Beholder has one extension to the standard TFTP file system. A normal TFTP file request has the following layout:

get */directory1/directory2/filename*

With The Beholder, it is possible to refer to a disk by using the following filename:

get *//disk/directory1/directory2/filename*

for example:

get //c/beholder/error.out

# 3. Network Management

When using The Beholder to really manage your ethernet network, you should have a beholder tentacle in each of the segments that make up your ethernet. The data should be collected in a central network management station on a regular bases. Reports can then be generated of the load and traffic characteristics of any period of time.

# 4. Developments

There are a number of development under way which concern the Beholder.

The first is a developers toolkit. The structure of the Beholder is such that independent applications can be linked to the kernel of The Beholder. Each application present in the runtime version of The Beholder can be activated at any time. An application gets a message in cases such as the arrival of a packet, the elapse of a second, or if there is freetime to be burned.

The seconf development is of a packet-tracing application that can be activated through SNMP variables. The resulting trace-file can then be collected using the TFTP file transfer protocol.

The third development is on the network management side of the medal. A SAS database is being build and SAS procedures to interpret the result and generate report. The data is currently being collected by the SUN NetManagement software. We are also looking in to the possibility to connect the G2 real-time expert system environment to the Sun software to make a real-time analysis of traps and other network events.

# 5. Credit and Disclaimer

The Beholder is the result of a lot of work by a number of people:

| | |
|---|---|
| Jan van Oorschot | Project "leader" and Sage |
| Ling Thio | First version and user interface |
| Wim van Campen | UDP/IP and applications |
| Dirk Wisse | SNMP ,kernel adjustments and SD matrix |
| Alfred Kayser | Debugging ,assembler and DSCHEME |
| Kees and Wil | First version of the Source/Destination matrix |
| Bert Meijs | Technical support |

We don't want money for our work. As we work on a University, we would like invitations to publish and present our papers concerning The Beholder and the INEMA project. If you really have use of our products, you could even pay for the trip! (Hotelroom with shower would be nice). We have papers on the kernel of The Beholder, the UDP/IP stack, ethernet performance measurements, bridge positioning and a lot more. A few of them have been published, but repetition is the essence of learning !

The very least one could is to send us a note with bugs, comments, compliments and The Answer To The Final Question.

Greetings and be careful out there

Jan

JPMvOorschot@et.tudelft.nl

# Appendix A: Beholder.ini parameters

The file Beholder.ini contains all configuration parameters for The Beholder system. The file is partitioned in serveral *sections*. Each section contains configuration onformation about a part of The Beholder. This appendix describes the configuration *parameters* according to the section in which they appear.

section in Beholder.ini is identified by a line with the following contents

[*section-name*]

## 5.1. Buffer

This section configures the memory allocation of The Beholder. This section only contains parameters.

numbuffers

| | |
|---|---|
| **Name** | numbuffers |
| **Description** | number of buffers to be allocated |
| **Values** | 3 4 5 .... |
| **Example** | 5 |

buffersize

| | |
|---|---|
| **Name** | buffersize |
| **Description** | size of one buffer |
| **Values** | 65500 |
| **Example** | 65500 |
| **Note** | no other value then 65500 is accepted |

## 5.2. DISPATCHER

This section configures the ring buffers of The Beholder. These buffers are used to store network packets that can't be handled imidiatedly by The Beholder.

SizeSmall

| | |
|---|---|
| **Name** | SizeSmall |
| **Description** | Maximum size of a small packet |
| **Values** | 64...1514 |
| **Example** | 192 |
| **Note** | 192 is probably the best value |

CountSmall

| | |
|---|---|
| **Name** | CountSmall |
| **Description** | number of buffers for small packets |
| **Values** | 1... |
| **Example** | 75 |
| **Note** | increasing this value will let The Beholder lose less packets, but eats memory. |

SizeLarge

| | |
|---|---|
| **Name** | SizeLarge |
| **Description** | Maximum size of a large packet |
| **Values** | 64...1514 |
| **Example** | 1514 |
| **Note** | 1514 is probably the best value |

CountLarge

| | |
|---|---|
| **Name** | CountLarge |
| **Description** | Number of Buffers for large packets |
| **Values** | 1 ... |
| **Example** | 100 |
| **Note** | increasing this one will cost you ... |

## 5.3. IPDOS

This section configures the IP stack in The Beholder. Each Beholder is a full functional IP node, and should have all information needed by an IP node. The routing information is stored in the section [ROUTES].

Forwarding

| | |
|---|---|
| **Name** | Forwarding |
| **Description** | Indicates if The Beholder should forward IP packets not mend for the IP address of the Beholder. Setting this parameter to 'no' will disable The Beholder to function as IP router. |
| **Values** | yes/no |
| **Example** | yes |
| **Note** | |

## 5.4. ROUTES

This section is not formatted like the other sections. Each line contains information for the IP routing done by The Beholder. Each line has the following format:

<operation> <IP-address> <and-mask>

The following operations are possible:

hostmodify: add a dynamic route to a host , this can be changed by redirect messages.
hoststat: add a static route to a host .
netmodify: add a dynamic route to a net.
netstat: add a static route to a net.

If The Beholder receives an IP message, and tries to find the correct routing entry, it takes the destination IP address, "AND"'s it with the <and-mark>, and does a byte-compare with the <IP-address> of each entry in the routing table.
If <IP-address> is "default", it will be used for all messages that don't match an other entry in the routing table.

The routing Section should always contain:

- definition of the default IP gateway on your own net
- definition of the loopback interface 127.0.0.1
- route to your own IP address through 127.0.0.1
- route to your own network

## 5.5. SYSTEM

This section contains system information that is replied when SNMP requests are send to this Beholder.

Description

| **Name** | Description |
|---|---|
| **Description** | String describing this beholder |
| **Values** | any-string |
| **Example** | "The Beholder, version 1bA" |
| **Note** | |

Contact

| **Name** | Contact |
|---|---|
| **Description** | Name of contact person for this Beholder |
| **Values** | any-string |
| **Example** | jan van Oorschot (6179) |
| | |

Name

| **Name** | Name |
|---|---|
| **Description** | Name of for this Beholder |
| **Values** | any-string |
| **Example** | Beholder1 |

Location

| **Name** | Location |
|---|---|
| **Description** | Location of the Beholder |
| **Values** | any-string |
| **Example** | Room 9.03 |

## 5.6. AUTHENTICATION

The Authentication section configures the communities for the SNMP variables. It determines which users get access to which SNMP variables. The layout of this section is again not conform the normal variable/value standard.
This section contains subsection, each subsection of the form:

Community <Community-name>
AddAddress <hostnumber1> <and-mask1>
AddAddress <hostnumber2> <and-mask2>

There is a subsection for each community you there is in The Beholder. At the moment, all variables are in the "public" community.

## 5.7. AGENT

The AGENT section describes the SNMP agent as it is implemented in The Beholder. The section section used to configure the SNMP agent is the AUTHENTICATION section.

ObjectID

| **Name** | ObjectID |
|---|---|
| **Description** | ASN1 object ID of Beholder variable-tree |
| **Values** | ASN1-variable |
| **Example** | 1.3.6.1.4.1.99 |

Port

| **Name** | Port |
|---|---|
| **Description** | UDP port used by SNMP agent |
| **Values** | 161 |
| **Example** | 161 |

Trap

| | |
|---|---|
| **Name** | Trap |
| **Description** | Enable/disable SNMP trap generation |
| **Values** | enable/disable |
| **Example** | enable |

TrapPort

| | |
|---|---|
| **Name** | TrapPort |
| **Description** | UDP port used to send traps to |
| **Values** | 162 |
| **Example** | 162 |

Trapaddress

| | |
|---|---|
| **Name** | TrapAddress |
| **Description** | IP address of network managent station handling SNMP traps |
| **Values** | IP address |
| **Example** | 130.161.144.171 |

TrapCommunity

| | |
|---|---|
| **Name** | TrapCommunity |
| **Description** | Community used when sending traps |
| **Values** | any-string |
| **Example** | trap |

## 5.8. MATRIX0

This section configures the source destination matrix for interface 0.

HostTableLength

| | |
|---|---|
| **Name** | HostTableLength |
| **Description** | Maximum number of hosts that can be kept by the SD matrix |
| **Values** | integer |
| **Example** | 1500 |

ConnectioTableLength

| | |
|---|---|
| **Name** | ConnectionTableLength |
| **Description** | Maximum number of connections that can be kept by the SD matrix. |
| **Values** | integer |
| **Example** | 3000 |

HashTableLength

| | |
|---|---|
| **Name** | HashTableLength |
| **Description** | Number of entries in the hosts hash table. Should be bigger then HostTableLength |
| **Values** | integer |
| **Example** | 2000 |

## 5.9. ERRORS

ErrorFile

| | |
|---|---|
| **Name** | ErrorFile |
| **Description** | name of file to which error messages will be send |
| **Values** | file-name |
| **Example** | error.out |

DebugFile

| | |
|---|---|
| **Name** | DebugFile |
| **Description** | name of file to which debug messages will be send. |
| **Values** | file-name |
| **Example** | debug.out |

DebugLevel

| | |
|---|---|
| **Name** | DebugLevel |
| **Description** | Level of debugging. 0 is no debugging, 6 is highest level of debugging |
| **Values** | integer 0<= int <= 6 |
| **Example** | 0 |

## 5.10. GENERIC

## 5.11. MATRIX

## 5.12. DISPLAY

etc

Every application in The Beholder has its own section. The name of the section is the name of the application. Type ESC in a running Beholder to see the applications. If you are not sure how to set these variables, leave them out, the defaults are OK.
In each application section the following variables can be defined:

EventMask

| | |
|---|---|
| **Name** | EventMask |
| **Description** | bitmask describing which events to send to the application during Beholder run-time |
| **Values** | #define DPE_SHOW 0x0001 /* Dispatcher Events */<br>#define DPE_START 0x0002<br>#define DPE_STOP 0x0004<br>#define DPE_HIDE 0x0008<br>#define DPE_RESET 0x0010<br>#define DPE_KEYPRESSED 0x0020<br>#define DPE_INIT 0x0040<br>#define DPE_END 0x0080<br>#define DPE_RECEIVEPKT 0x0100<br>#define DPE_FREETIME 0x0200<br>#define DPE_EVERYSECOND 0x0400<br>#define DPE_TIMER 0x0800 |
| **Example** | 0xffff |

StartMask

| **Name** | EventMask |
|---|---|
| **Description** | bitmask describing which events should be generated during startup of the application. These can be used to initialise the application. |
| **Values** | #define DPE_SHOW 0x0001 /* Dispatcher Events */<br>#define DPE_START 0x0002<br>#define DPE_STOP 0x0004<br>#define DPE_HIDE 0x0008<br>#define DPE_RESET 0x0010<br>#define DPE_KEYPRESSED 0x0020<br>#define DPE_INIT 0x0040<br>#define DPE_END 0x0080<br>#define DPE_RECEIVEPKT 0x0100<br>#define DPE_FREETIME 0x0200<br>#define DPE_EVERYSECOND 0x0400<br>#define DPE_TIMER 0x0800 |
| **Example** | 0x0003 |

TimerValue

| **Name** | TimerValue |
|---|---|
| **Description** | time-interval in seconds in which the application runs. After an interval, the application is reset, and starts again. |
| **Values** | integer |
| **Example** | 500 |