PREFACE

This users guide and its associated disc are a set of unsupported tools for use with the HP 4971S LAN Protocol Analyzer. These tools were developed to help the HP 4971S be more useful in analyzing Local Area Networks. The utilities can be easily modified and customized to your particular needs.

This users guide is not intended to document protocol functions or features; it is meant to document the utilities that are contained on the associated disc.

While most of these utilities have been tested they have not gone through a rigorous testing/Q.A. program. If bugs are found please let us know by using the bug/enhancement reporting form at the back of this guide.

If you develop utilities that you feel would be useful to other users and would like to share them, please send a printed copy of your utility with a completed bug/enhancement form to the address shown on the form.

Note

These tools are UNSUPPORTED and are provided on an as is basis.
# TABLE OF CONTENTS

## CHAPTER 1: NETWORK PERFORMANCE MEASUREMENTS

- Address Analysis - Adr_Anl  
- Collision Counting - CollCnt  
- Network Loading Bad data - LoadNtB  
- Network Loading Good data - LoadNtG  
- Frame Size Analysis - SizeAnl  
- Traffic analysis over a day - TimDay  
- Traffic analysis over an hour - TimHour  
- Traffic analysis over a shift - TimShft

## CHAPTER 2: PROTOCOL ANALYSIS

- Digital Equipment Maintenance Operation Protocol - DECNMOP  
- Digital Equipment Network Service Protocol - DECNNSP  
- Digital Equipment Routing Layer Protocol - DECNROT  
- Hewlett-Packard Probe Protocol - HPProbe  
- ISO Transport Protocol - ISOtrns  
- Transmission Control Protocol and Internet Protocol - TCPIP  
- Ethernet Type Protocol Analysis - Typ_Anl  
- Xerox Network System Protocol - XNS

## CHAPTER 3: TESTING

- Digital Equipment System ID request - DECNSID  
- Ethernet Configuration Testing Protocol - EthrCTP  
- Xerox Network System Echo Protocol - XNSLoop  
- IEEE 802.2 XID and Test - XID_Tst

## CHAPTER 4: MISCELLANEOUS

- Manufacture Address Analysis - Mfg_Anl
# CONTENTS BY FILE NAME

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adr Anl</td>
<td>Address Analysis</td>
<td>1-2</td>
</tr>
<tr>
<td>CollCnt</td>
<td>Collision Counting</td>
<td>1-3</td>
</tr>
<tr>
<td>DECNMOP</td>
<td>Digital Equipment Maintenance Operation Protocol</td>
<td>2-2</td>
</tr>
<tr>
<td>DECNNSP</td>
<td>Digital Equipment Network Service Protocol</td>
<td>2-4</td>
</tr>
<tr>
<td>DECNROT</td>
<td>Digital Equipment Routing Layer Protocol</td>
<td>2-5</td>
</tr>
<tr>
<td>DECNSSID</td>
<td>Digital Equipment System ID request</td>
<td>3-2</td>
</tr>
<tr>
<td>EthrCTP</td>
<td>Ethernet Configuration Testing Protocol</td>
<td>3-3</td>
</tr>
<tr>
<td>HPProbe</td>
<td>Hewlett-Packard Probe Protocol</td>
<td>2-6</td>
</tr>
<tr>
<td>ISOtrns</td>
<td>ISO Transport Protocol</td>
<td>2-7</td>
</tr>
<tr>
<td>LoadNtB</td>
<td>Network Loading Bad Data</td>
<td>1-4</td>
</tr>
<tr>
<td>LoadNtG</td>
<td>Network Loading Good Data</td>
<td>1-6</td>
</tr>
<tr>
<td>Mfg Anl</td>
<td>Manufacture address analysis</td>
<td>4-2</td>
</tr>
<tr>
<td>SizeAnl</td>
<td>Frame Size Analysis</td>
<td>1-8</td>
</tr>
<tr>
<td>TCPIIP</td>
<td>Transport Communication and Internet Protocol</td>
<td>2-8</td>
</tr>
<tr>
<td>TimDay</td>
<td>Traffic analysis over a day</td>
<td>1-9</td>
</tr>
<tr>
<td>TimHour</td>
<td>Traffic analysis over an hour</td>
<td>1-10</td>
</tr>
<tr>
<td>TimShft</td>
<td>Traffic analysis over a shift</td>
<td>1-11</td>
</tr>
<tr>
<td>Typ Anl</td>
<td>Ethernet Type Protocol Analysis</td>
<td>2-10</td>
</tr>
<tr>
<td>XID Tst</td>
<td>IEEE 802.2 XID and Test</td>
<td>3-6</td>
</tr>
<tr>
<td>XNS</td>
<td>Xerox Network System Protocol</td>
<td>2-11</td>
</tr>
<tr>
<td>XNSLoop</td>
<td>Xerox Network System Echo Protocol</td>
<td>3-5</td>
</tr>
</tbody>
</table>
MAKING A BACKUP DISC

We recommend you keep the supplied utility disc as a master copy and make backup copies for your daily use. Chapter 18, "<Disc Functions>" in the HP 4971S Users Manual describes how to make copies of a disc.

Keep the master copy of the utility disc in a protected environment.
CHAPTER 1

NETWORK PERFORMANCE MEASUREMENTS

This chapter discusses various utilities for looking at network performance characteristics.

Utilities are included which count frames and collisions in single samples or in intervals over an hour, shift, or day. There are utilities to analyze traffic by frame size or address type.

Additionally, there are traffic generator programs to generate traffic at various levels.

These programs can be easily modified to monitor different time intervals, etc.
Network Performance Measurements

FILE NAME: **Adr_Anl**

PURPOSE: Categorize traffic by destination address type over 60 seconds.

FILTERS:
- **Broadcast_Address**: Matches broadcast destination address
- **Individual_Addr**: Matches individual destination address
- **Group_Address**: Matches group destination address
- **Global_Address**: Matches global destination address
- **Local_Address**: Matches local destination address

MESSAGES: None

PROGRAMS:
- **AddressAn**: Count frames by address type

DESCRIPTION: This program counts the number of frames by destination address type over a 60 second period.
FILE NAME: **CollCnt**

PURPOSE: Count the number of frames and collisions over a 60 second period.

FILTERS: None

MESSAGES: None

PROGRAMS: CountColl  Count frames and collisions over 60 seconds

DESCRIPTION: This simple program counts the number of frames and collisions over a 60 second period.
**FILE NAME:** LoadNtB  

**PURPOSE:** To generate background network traffic load at various levels with error frames.

**FILTERS:** None

**MESSAGES:**
- 62 _Byte_Runt_ 62 byte data frame
- 64 _Byte_Message_ 64 byte data frame
- 115 _Byte_Message_ 115 byte data frame
- 226 _Byte_Message_ 226 byte data frame
- 337 _Byte_Message_ 337 byte data frame
- 448 _Byte_Message_ 448 byte data frame
- 559 _Byte_Message_ 559 byte data frame
- 670 _Byte_Message_ 670 byte data frame
- 781 _Byte_Message_ 781 byte data frame
- 892 _Byte_Message_ 892 byte data frame
- 1003 _Byte_Message_ 1003 byte data frame
- 1115 _Byte_Bad_FCS_ 1115 byte data frame
- 1226 _Byte_Message_ 1226 byte data frame
- 1337 _Byte_Message_ 1337 byte data frame
- 1448 _Byte_Message_ 1448 byte data frame
- 2026 _Byte_Jabber_ 2026 byte data frame

**PROGRAMS:**
- BadTraf10  Generate 10% traffic load
- BadTraf20  Generate 20% traffic load
- BadTraf30  Generate 30% traffic load
- BadTraf40  Generate 40% traffic load
- BadTrafMX  Generate 94% Traffic load
DESCRIPTION: This set of programs generates a network traffic load, including frames in error, by transmitting frames of various sizes at various intervals over time. Because the HP 4971 follows the CSMA/CD algorithm the percentages shown for each program are maximums. If a collision occurs while the HP 4971 is transmitting, it follows the standard back off algorithm and therefore, less traffic is generated.

The following table gives an estimate of the percentage distribution of frames sizes that are generated for each network loading program:

<table>
<thead>
<tr>
<th>Frame Size</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>94%</th>
</tr>
</thead>
<tbody>
<tr>
<td>62</td>
<td>1.0%</td>
<td>.9%</td>
<td>.8%</td>
<td>2.2%</td>
<td>0.0%</td>
</tr>
<tr>
<td>64</td>
<td>78.5%</td>
<td>74.1%</td>
<td>74.1%</td>
<td>47.4%</td>
<td>0.0%</td>
</tr>
<tr>
<td>115</td>
<td>3.6%</td>
<td>4.8%</td>
<td>4.8%</td>
<td>8.7%</td>
<td>0.0%</td>
</tr>
<tr>
<td>226</td>
<td>2.2%</td>
<td>2.8%</td>
<td>3.1%</td>
<td>5.7%</td>
<td>0.0%</td>
</tr>
<tr>
<td>337</td>
<td>1.5%</td>
<td>2.0%</td>
<td>1.9%</td>
<td>4.7%</td>
<td>0.0%</td>
</tr>
<tr>
<td>448</td>
<td>1.2%</td>
<td>1.2%</td>
<td>1.3%</td>
<td>3.2%</td>
<td>0.0%</td>
</tr>
<tr>
<td>559</td>
<td>1.2%</td>
<td>1.2%</td>
<td>1.3%</td>
<td>3.2%</td>
<td>0.0%</td>
</tr>
<tr>
<td>670</td>
<td>1.5%</td>
<td>2.0%</td>
<td>1.9%</td>
<td>4.7%</td>
<td>0.0%</td>
</tr>
<tr>
<td>781</td>
<td>2.2%</td>
<td>2.8%</td>
<td>3.1%</td>
<td>5.7%</td>
<td>0.0%</td>
</tr>
<tr>
<td>892</td>
<td>3.6%</td>
<td>4.8%</td>
<td>4.8%</td>
<td>8.7%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1003</td>
<td>1.0%</td>
<td>.9%</td>
<td>.8%</td>
<td>2.2%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1115</td>
<td>.8%</td>
<td>.7%</td>
<td>.7%</td>
<td>1.6%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1226</td>
<td>.6%</td>
<td>.6%</td>
<td>.5%</td>
<td>1.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1337</td>
<td>.4%</td>
<td>.6%</td>
<td>.5%</td>
<td>.7%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1448</td>
<td>.4%</td>
<td>.4%</td>
<td>.4%</td>
<td>.4%</td>
<td>0.0%</td>
</tr>
<tr>
<td>2024</td>
<td>.0%</td>
<td>.0%</td>
<td>.0%</td>
<td>.0%</td>
<td>100.0%</td>
</tr>
</tbody>
</table>
Network Performance Measurements

FILE NAME: LoadNtG

PURPOSE: To generate background network traffic load at various levels.

FILTERS: None

MESSAGES:
- 64_Byte_Message1 64 byte data frame
- 64_Byte_Message2 64 byte data frame
- 115_Byte_Message 115 byte data frame
- 226_Byte_Message 226 byte data frame
- 337_Byte_Message 337 byte data frame
- 448_Byte_Message 448 byte data frame
- 559_Byte_Message 559 byte data frame
- 670_Byte_Message 670 byte data frame
- 781_Byte_Message 781 byte data frame
- 892_Byte_Message 892 byte data frame
- 1003_Byte_Message 1003 byte data frame
- 1115_Byte_Message 1115 byte data frame
- 1226_Byte_Message 1226 byte data frame
- 1337_Byte_Message 1337 byte data frame
- 1448_Byte_Message 1448 byte data frame
- 1518_Byte_Message 1518 byte data frame

PROGRAMS:
- Traffic10 Generate 10% traffic load
- Traffic20 Generate 20% traffic load
- Traffic30 Generate 30% traffic load
- Traffic40 Generate 40% traffic load
- TrafficMX Generate 94% Traffic load

DESCRIPTION: This set of programs generates a network traffic load by transmitting frames of various sizes at various intervals over time. Because the HP 4971 follows the CSMA/CD algorithm, the percentages shown for each program are maximums. If a collision occurs while the HP 4971 is transmitting, it follows the standard back off algorithm and therefore, less traffic is generated.
The following table gives an estimate of the percentage distribution of frames sizes that are generated for each network loading program:

<table>
<thead>
<tr>
<th>Frame Size</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>94%</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>79.3%</td>
<td>74.8%</td>
<td>74.6%</td>
<td>49.5%</td>
<td>0.0%</td>
</tr>
<tr>
<td>115</td>
<td>3.6%</td>
<td>4.8%</td>
<td>4.8%</td>
<td>8.7%</td>
<td>0.0%</td>
</tr>
<tr>
<td>226</td>
<td>2.2%</td>
<td>2.8%</td>
<td>3.1%</td>
<td>5.7%</td>
<td>0.0%</td>
</tr>
<tr>
<td>337</td>
<td>1.5%</td>
<td>2.0%</td>
<td>1.8%</td>
<td>4.7%</td>
<td>0.0%</td>
</tr>
<tr>
<td>448</td>
<td>1.2%</td>
<td>1.2%</td>
<td>1.3%</td>
<td>3.2%</td>
<td>0.0%</td>
</tr>
<tr>
<td>559</td>
<td>1.2%</td>
<td>1.2%</td>
<td>1.3%</td>
<td>3.2%</td>
<td>0.0%</td>
</tr>
<tr>
<td>670</td>
<td>1.5%</td>
<td>2.0%</td>
<td>1.9%</td>
<td>4.6%</td>
<td>0.0%</td>
</tr>
<tr>
<td>781</td>
<td>2.2%</td>
<td>2.8%</td>
<td>3.1%</td>
<td>5.7%</td>
<td>0.0%</td>
</tr>
<tr>
<td>892</td>
<td>3.6%</td>
<td>4.8%</td>
<td>4.8%</td>
<td>8.7%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1003</td>
<td>1.0%</td>
<td>.9%</td>
<td>0.8%</td>
<td>2.2%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1115</td>
<td>.8%</td>
<td>.7%</td>
<td>.7%</td>
<td>1.6%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1226</td>
<td>.6%</td>
<td>.6%</td>
<td>.5%</td>
<td>1.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1337</td>
<td>.4%</td>
<td>.6%</td>
<td>.5%</td>
<td>.7%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1448</td>
<td>.3%</td>
<td>.4%</td>
<td>.4%</td>
<td>.4%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1518</td>
<td>.3%</td>
<td>.4%</td>
<td>.4%</td>
<td>.3%</td>
<td>100.0%</td>
</tr>
</tbody>
</table>
Network Performance Measurements

FILE NAME: SizeAnl

PURPOSE: Categorize traffic by frame size over 60 seconds.

FILTERS:
- Frame_64_100Byte Matches frames of 64-100 bytes
- Frame_101_200Byte Matches frames of 101-200 bytes
- Frame_201_300Byte Matches frames of 201-300 bytes
- Frame_301_400Byte Matches frames of 301-400 bytes
- Frame_401_500Byte Matches frames of 401-500 bytes
- Frame_501_600Byte Matches frames of 501-600 bytes
- Frame_601_700Byte Matches frames of 601-700 bytes
- Frame_701_800Byte Matches frames of 701-800 bytes
- Frame_801_900Byte Matches frames of 801-900 bytes
- Frame_901_1000B Matches frames of 901-1000 bytes
- Frame_1001_1100B Matches frames of 1001-1100 bytes
- Frame_1101_1200B Matches frames of 1101-1200 bytes
- Frame_1201_1300B Matches frames of 1201-1300 bytes
- Frame_1301_1400B Matches frames of 1301-1400 bytes
- Frame_1401_1518B Matches frames of 1401-1518 bytes

MESSAGES: None

PROGRAMS: Length

DESCRIPTION: This program uses filters that match frames by size. When complete, the counters display the number of frames in each size category over the last 60 seconds.
FILE NAME: TimDay

PURPOSE: Count the number of frames during a 24 hour period.

FILTERS: Any_Frame Match any frame

MESSAGES: None

PROGRAMS: DayWatch Analyze traffic over a 24 hour period

DESCRIPTION: This program counts the number of frames and collisions over the period of 24 hours. Traffic is broken down in 2 hour intervals.

The program is designed to update the timer/counters display every two hours. Display updating is minimized so the program can keep up with high network loads. If the load is light, the display can be set to be continuously updated.

Note

If traffic is too heavy for the program to keep up, try removing all start and stop display statements except the last.
FILE NAME:  \textbf{TimHour}

PURPOSE: Count the number of frames in each 5 minute period during an hour.

FILTERS: Any\_Frame \quad Match any frame

MESSAGES: None

PROGRAMS: HourWatch \quad Analyze traffic over an hour

DESCRIPTION: This program counts the number of frames and collisions over the period of an hour. Traffic is broken down in five minute intervals.

The program is designed to update the timer/counters display every five minutes. Display updating is minimized so the program can keep up with high network loads. If the load is light, the display can be set to be continuously updated.

\textbf{Note}

If traffic is too heavy for the program to keep up, try removing all start and stop display statements except the last.
FILE NAME: TimShft

PURPOSE: Count the number of frames during an 8 hour period.

FILTERS: Any_Frame Match any frame

MESSAGES: None

PROGRAMS: ShftWatch Analyze traffic over an 8 hour period

DESCRIPTION: This program counts the number of frames and collisions over the period of 8 hours. Traffic is broken down in 1 hour intervals.

The program is designed to update the timer/counters display every hour. Display updating is minimized so the program can keep up with high network loads. If the load is light, the display can be set to be continuously updated.

Note

If traffic is too heavy for the program to keep up, try removing all start and stop display statements except the last.
This chapter covers the utilities designed to format and analyze various common level 3 and 4 protocols.

Utilities are provided to format DECNET, TCP/IP, XNS, and others.
Protocol Analysis

FILE NAME: DECNMOP

PURPOSE: Protocol format for Digital Equipment’s Maintenance Operation Protocol

FILTERS:
- Dump_Load
- Memory_Load
- Req_Memory_Dump
- Request_Program
- Req_Memory_Load
- Req_Dump_Service
- Memory_Dump_Data
- Parm_Load_with_TA
- Dump_Complete
- Boot
- Req_ID
- System_ID
- Request_Counters
- Counters
- Reserve_Console
- Release_Console

Matches dump load messages
Matches memory load messages
Matches request memory dump messages
Matches request program messages
Matches request memory load messages
Matches request dump service messages
Matches memory dump data messages
Matches program load with transfer address
Matches dump complete messages
Matches boot messages
Matches request identification messages
Matches system identification messages
Matches request counters message
Matches a counter response message
Matches a reserve console message
Matches a release console message

MESSAGES: None

PROGRAMS:
- MOP_Count

Program to count the number of frames of each MOP message type
**File** **DECNMOP** (cont.)

**DESCRIPTION:** This set of filters can be used to format Digital Equipment Maintenance Operation Protocol (MOP) messages. This is accessed through the filter format function in examine data.

The MOP protocol provides support for the control and management of nodes on a DECNet network.

MOP messages are in two categories:
- One group has an Ethernet type of 6001<sub>h</sub>.
- The other group has an Ethernet type of 6002<sub>h</sub>.

Because of the limit of 16 filters, the following two messages are not formatted:

**Console Command and Poll**

<table>
<thead>
<tr>
<th>Destination</th>
<th>Source</th>
<th>6002&lt;sub&gt;h&lt;/sub&gt; Byte Count</th>
<th>11&lt;sub&gt;h&lt;/sub&gt; Control Flag</th>
<th>Command Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte</td>
<td>6</td>
<td>6</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

**Console Response and Acknowledge**

<table>
<thead>
<tr>
<th>Destination</th>
<th>Source</th>
<th>6002&lt;sub&gt;h&lt;/sub&gt; Byte Count</th>
<th>13&lt;sub&gt;h&lt;/sub&gt; Control Flag</th>
<th>Command Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte</td>
<td>6</td>
<td>6</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
FILE NAME:  DECNNSP  

PURPOSE:  Protocol format for the DECnet layer 4 NSP protocol

FILTERS:  
- Data_Segment  Matches data from level 5
- Interrupt  Matches interrupt messages
- Link_Service  Matches link management messages
- Data_Acknowledge  Matches acknowledge data messages
- Other_Data_Ack  Matches acknowledge interrupt messages
- Connect_Ack  Matches connection acknowledge messages
- No_Operation  Matches No Op messages
- Connect_Initiate  Matches initiate a connection messages
- Retran_Conn_Init  Matches initiation retransmission messages
- Connect_Confirm  Matches confirm a connection messages
- Disconnect_Init  Matches initiate a disconnect messages
- Disconnect_Confirm  Matches confirm a disconnection messages

MESSAGES:  None

PROGRAMS:  NSP_Count  Program to count the number of each NSP message type.

DESCRIPTION:  This set of filters can be used to format Digital Equipment's DECnet NSP layer messages. DECnet messages have an Ethernet Type value of 6003h.

NSP message filters have optional padding at the routing layer. The padding can be from 1 to 8 bytes in size. Most DECnet traffic we have seen that has padding bytes, only has a single pad byte. For this reason, you may have to modify the filters to match the packet with or without padding.
FILE NAME: DECNROT

PURPOSE: Protocol format for the DECnet layer 3 protocol known as the Routing layer

FILTERS:
- Long_Data_Packet: Matches higher level protocol messages
- Routing_Hello: Matches Level 3 node notification messages
- Endnode_Hello: Matches level 3 node notification messages
- Routing_Level1: Matches level 3 routing information messages
- Routing_Level2: Matches level 3 routing information messages
- Long_Data_PacketP: Matches opt padding for higher level protocol messages
- Routing_HelloP: Matches opt padding for level 3 node notification messages
- Endnode_HelloP: Matches opt padding for level 3 node notification messages
- Routing_Level1P: Matches opt padding for level 3 routing info messages
- Routing_Level2P: Matches opt padding for level 3 routing info messages

MESSAGES: None

PROGRAMS: ROUT_Count
Program to count the number of routing layer messages.

DESCRIPTION: This set of filters can be used to format Digital Equipment's DECnet Routing layer messages. DECnet messages have an Ethernet type value of 6003_{16}. Routing messages have an optional padding field that can be from 1 to 8 bytes in size. Most DECnet traffic we have seen that has padding bytes only has a single pad byte. For this reason, there are two copies of each filter, one with and one without the pad byte.
FILE NAME: **HPProbe**

**PURPOSE:** Protocol format for the Hewlett-Packard Probe protocol.

**FILTERS:**
- 802PR_Name_Req: Matches name request message in a 802.3 frame
- 802PR_Name_Reply: Matches name reply message in a 802.3 frame
- 802PR_VirtAdr_Req: Matches virtual address request message in a 802.3 frame
- 802PR_VirtAdr_Rep: Matches virtual address reply message in a 802.3 frame
- 802PR_Usolic_Req: Matches unsolicited reply message in a 802.3 frame
- 802PR_Gateway_Req: Matches where is gateway request message in a 802.3 frame
- 802PR_Gateway_Rep: Matches where is gateway reply message in a 802.3 frame
- EthPR_Name_Req: Matches name request message in an Ethernet frame
- EthPR_Name_Reply: Matches name reply message in an Ethernet frame
- EthPR_VirtAdr_Req: Matches virtual address request message in an Ethernet frame
- EthPR_VirtAdr_Rep: Matches virtual address reply message in an Ethernet frame
- EthPR_Usolic_Rep: Matches unsolicited reply message in an Ethernet frame
- EthPR_Gateway_Req: Matches where is gateway request message in an Ethernet frame
- EthPR_Gateway_Rep: Matches where is gateway reply message in an Ethernet frame

**MESSAGES:** None

**PROGRAMS:** None

**DESCRIPTION:** This set of filters can be used to format the Hewlett-Packard Probe Protocol.
FILE NAME: ISOtrs


FILTERS:
- Connection_Req: Matches a connection request
- Connection_Confrm: Matches a connection confirmation
- Disconnect_Req: Matches a disconnection request
- Disconnect_Confrm: Matches a disconnection confirmation
- Data: Matches data from the upper layers
- Expedited_Data: Matches expedited data packets
- Acknowledgment: Matches acknowledgment
- Expedited_Ack: Matches expedited acknowledgment
- Reject: Matches rejection sequence
- Error: Matches error frames

MESSAGES: None

PROGRAMS: ISO_Count Program to count the number of ISO Transport messages by type.

DESCRIPTION: This set of filters can be used to format the ISO Transport Protocol packets. Additionally, there is a program to count the number of packets by type.
FILE NAME: **TCPIP**

PURPOSE: Protocol format for the Transmission Control Protocol (TCP) and the Internet Protocol (IP) from DOD.

FILTERS:

- **802_TCPIP_SYNACK** Matches synchronize & acknowledge message in a 802.3 frame
- **802_TCPIP_FIN** Matches finish message in a 802.3 frame
- **802_TCPIP_RST** Matches a reset connection message in a 802.3 frame
- **802_TCPIP_EOL** Matches an end of letter message in a 802.3 frame
- **802_TCPIP_URG** Matches an urgent message in a 802.3 frame
- **802_TCPIP_ACK** Matches an acknowledge message in a 802.3 frame
- **802_TCPIP_SYN** Matches a synchronize message in a 802.3 frame
- **Ethr_TCPIP_SYNACK** Matches synchronize & acknowledge message in a 802.3 frame
- **Ethr_TCPIP_FIN** Matches finish message in a 802.3 frame
- **Ethr_TCPIP_RST** Matches a reset connection message in a 802.3 frame
- **Ethr_TCPIP_EOL** Matches an end of letter message in a 802.3 frame
- **Ethr_TCPIP_URG** Matches an urgent message in a 802.3 frame
- **Ethr_TCPIP_ACK** Matches an acknowledge message in a 802.3 frame
- **Ethr_TCPIP_SYN** Matches a synchronize message in a 802.3 frame
- **802_IP_Detail** Matches an IP header in a 802.3 frame
- **Ethr_IP_Detail** Matches an IP header in an Ethernet frame

(continued)
File **TCPIP** (cont.)

**MESSAGES:** None

**PROGRAMS:** None

**DESCRIPTION:** This set of filters can be used to format Transmission Control Protocol and the Internet Protocol from the Department of Defense. This set of filters matches TCP/IP in either IEEE 802 or Ethernet format.

Because some of the different message types can be combined, multiple filters may match a given message.

The format can be accessed through the filter format in the <Examine Data> Menu.
Protocol Analysis

FILE NAME: Typ_Anl

PURPOSE: Categorize network traffic by Ethernet type field.

FILTERS:

- XNS Matches XNS frames
- DOD_IP Matches DOD IP (TCP/IP) frames
- X75_Internet Matches X75 Internet frames
- NBS_Internet Matches NBS Internet frames
- ECMA_Internet Matches ECMA Internet frames
- DEC_MOP_Dump_Load Matches DEC MOP frames
- DEC_MOP_Console Matches DEC MOP frames
- DEC_DECNet Matches DECNet frames
- DEC_LAT Matches DEC LAT (terminal server) frames
- DEC_User Matches DEC User defined frames
- Ethernet_Loopback Matches Ethernet Loopback frames

MESSAGES: None

PROGRAMS: Type_Anal Program to count the number of frames of each message type.

DESCRIPTION: This utility can be used to analyze traffic on a network operating in Ethernet mode. Traffic is broken down by the Ethernet Type field according to higher level protocols.
FILE NAME: XNS

PURPOSE: Protocol format for the Xerox Network System protocols at levels 3 and 4.

FILTERS:
- Sequenced_Packet: Matches a sequenced packet message
- Packet_Exchange: Matches a packet exchange message
- Routing_Info_Req: Matches a routing information request message
- Routing_Info_Resp: Matches a routing information response message
- Echo_Req: Matches an echo request message
- Echo_Reply: Matches an echo reply message
- Error: Matches an error message
- Internet_Datagram: Matches any level 3 datagrams

MESSAGES: None

PROGRAMS:
- XNS_Count: Program to count the number of XNS messages by type.

DESCRIPTION: This set of filters can be used to format Xerox Network Systems Internet Datagram protocol and the XNS transport protocols. Additionally, there is a program to count the number of packets by type. The format function can be accessed through the filter format function of the <Examine Data> Menu.
Protocol Analysis
CHAPTER 3
TESTING

This chapter documents utilities that can be used to test the proper connection and operation of nodes on the network.

Utilities are provided to test a node using IEEE 802 XID test and Ethernet configuration testing protocol as well as others.
FILE NAME: DECNSID

PURPOSE: Send a Request ID message to a DEC node and trap response

FILTERS:
- Any,System_ID: Matches any System ID responses
- Request_Sys_ID: Matches specific system ID request
- Response_Sys_ID: Matches specific system ID response

MESSAGES:
- Req_Sys_ID_Mesg: Request System ID message

PROGRAMS:
- Req_Sys_ID: Program to send Request Sys ID message and time the response
- Any_SysID: Program to collect and count Sys ID messages

DESCRIPTION: This utility may be used to send a Digital Equipment Maintenance Operation Protocol (MOP) Request ID message and trap the response. This is a simple test to see if a DEC node is alive and is responding.

To execute, the following parameters must filled in:

Messages: Source Address: HP 4971S address
Destination Address: DEC node (Use the DECnet address ie. AA-00-04...)

Filters: Req_Sys_ID: Source Address-HP 4971S address
Destination Address-DEC node address
Sys_ID_Resp: Source Address-DEC node address
Destination Address-HP 4971S address
FILE NAME: EthrCTP

PURPOSE: Test nodes with the Ethernet Loop Back Configuration Testing Protocol

FILTERS:  
1_Hop_Send Matches a CTP 1 Hop request  
1_Hop_Reply Matches a CTP 1 Hop reply  
1_Hop_Generic Matches a CTP 1 Hop message  
2_Hop_Send Matches a CTP 2 Hop request  
2_Hop_Reply Matches a CTP 2 Hop reply  
2_Hop_Generic Matches a CTP 2 Hop message  
3_Hop_Send Matches a CTP 3 Hop request  
3_Hop_Reply Matches a CTP 3 Hop reply  
3_Hop_Generic Matches a CTP 3 Hop message  
Multicast_Send Matches a CTP Multicast request  
Multicast_Reply Matches a CTP Multicast reply  
Generic_CTP Matches a CTP message

MESSAGES:  
1_Hop_Send_Msg A single hop CTP request  
2_Hop_Send_Msg A two step hop CTP request  
3_Hop_Send_Msg A three step hop CTP request  
Multicast_Assist A Broadcast CTP assist message

PROGRAMS:  
1_Hop_Tst Send a 1 hop request and wait for response  
2_Hop_Tst Send a 2 hop request and wait for responses  
3_Hop_Tst Send a 3 hop request and wait for responses  
Multicast Send a multicast request and display responses

DESCRIPTION: This utility may be used to send various types of Ethernet loopback messages. This loopback protocol is known as the Ethernet Configuration Testing Protocol (CTP). The messages defined executes a 1, 2 or 3 hop loopback test. There is also a message defined using the loopback assist multicast address as the destination address. This causes all nodes that can act as loopback assistants to reply to the HP 4971.
DESCRIPTION: (cont.)

To execute any of the programs the following parameters for the appropriate messages and filters must be filled in (format the message as a filter for ease of entry).

**Messages:**
- Source Address: The HP4971.
- Destination Address: The 1st node you wish to loop to.
- Forward Address: The next node you want the message forwarded to.
- Final Destination Address: The HP 4971 address.

**Filters:**
- **Send filters:**
  - Destination Address: X’s (don’t cares).
  - Source Address: The HP 4971.
  - Final Destination Address: The HP 4971 (or all don’t cares).

- **Reply filters:**
  - Destination Address: The HP 4971.
  - Source Address: X’s (don’t cares)
  - Final Destination Address: The HP 4971 (or all don’t cares)

The only difference between send and reply filters, other than Source and Destination Addresses, is the Skip_count field:

For the Send filters, the Skip_count field is 00-00.

For the Reply filters, it is XX-XX.

The Generic_CTP filter matches all messages that have a Type field of 90-00.
FILE NAME: XNSLoop

PURPOSE: Send a XNS echo request and trap the reply message.

FILTERS: XNS_Echo_Request Matches an XNS Echo request
XNS_Echo_Reply Matches an XNS Echo response

MESSAGES: XNS_Echo_Request XNS Echo request message

PROGRAMS: XNSEchoRq Program to send XNS echo request and trap the response

DESCRIPTION: This utility may be used to send a Xerox Network System echo request. This is a simple test to see if a XNS node is alive and is responding.

To execute, the following parameters must filled in:

Messages: Source Address The HP4971 Destination Address - the node you wish to loop to.
IDP_Dest_Host The Ethernet address of the node you wish to loop to.
IDP_Source_Host The Ethernet address of the HP 4971.

Optionally, you may want to change the IDP socket or network numbers depending on your local environment

Filters: Send Filters: Source Address The HP4971.
Destination Address The node you wish to loop to.

Reply Filters: Source Address The HP4971.
Destination Address The node you wish to loop to.

Optionally, you may want to choose to use the IDP host, socket or network numbers depending on your needs.
FILE NAME: XID_Tst

PURPOSE: Send a 802.2 XID or Test message and trap response

FILTERS:
- XID_Command: Matches a XID command
- XID_Response: Matches a XID response
- TEST_Command: Matches a Test command
- TEST_Response: Matches a Test response

MESSAGES:
- XID_Command_Brdcs: Broadcast XID command
- TEST_Command_Brdcs: Broadcast Test command
- XID_Command_Indiv: XID command to an individual address
- TEST_Command_Indiv: TEST command to an individual address

PROGRAMS:
- XID_Req: Send a XID request to an individual address
- TEST_Req: Send a TEST request to an individual address
- XID_Broad: Send a XID broadcast message
- TEST_Broad: Send a TEST broadcast message

DESCRIPTION: This utility may be used to send either 802.2 Exchange ID commands or 802.2 Test loopback commands. These can be used to test the attachment and functioning of nodes to the network. For each message there are two programs: one to transmit to an individual address, and another, that sends a message with the broadcast address.

To execute, the following parameters must filled in:

Messages: Source Address
           Destination Address
           HP 4971.
           For non broadcast, set to appropriate node.

Filter: Source Address
        Destination Address
        For request, use HP 4971.
        For reply, use X’s (don’t cares).
        For non broadcast request, use object node.
        For reply, use HP 4971.
The following utilities do not fit in any of the prior categories.
FILE NAME: **Mfg_Anl**

PURPOSE: Categorize traffic by manufacture according to the first three bytes of the source address.

FILTERS:
- ATT Matches ATT’s address prefix
- Bridge Matches Bridge’s address prefix
- Digital_Equipment Matches DEC’s address prefix
- Hewlett_Packard Matches HP’s address prefix
- SUN Matches SUN’s address prefix
- Ungermann-Bass Matches UB’s address prefix
- Symbolics Matches Symbolic’s address prefix
- DECNet Matches DECNet Addresses
- DECNet_Multicast Matches DECNet Multicast Addresses
- None_of_the_Above Matches when the above don’t

MESSAGES: None

PROGRAMS: Mfg_Count Count frames by manufacture address

DESCRIPTION: This program uses filters that match the first three bytes of the source address based on the manufactures registered address prefix. The program is currently set to run for 60 seconds.