Network Working Group
Request for Comments: 1574
Obsoletes: 1139
Category: Informational

S. Hares
Merit/NSFNET
C. Wittbrodt
Stanford University/BARRNet
February 1994

Essential Tools for the OSI Internet

Status of this Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Abstract

This document specifies the following three necessary tools to debug problems in the deployment and maintenance of networks using ISO 8473 (CLNP):

- ping or OSI Echo function
- traceroute function which uses the OSI Echo function
- routing table dump function

These CLNS tools are the basics required for hosts and routers for CLNS network support. It is intended that this document specify the most basic support level required for CLNS hosts and routers.

To support some of the needed tools (ping and traceroute) this memo specifies the mechanism specified in RFC 1575 [3].

Table of Contents

1. Conventions

The following language conventions are used in the items of specification in this document:

- MUST, SHALL, or MANDATORY -- the item is an absolute requirement of the specification.
- SHOULD or RECOMMENDED -- the item should generally be followed for all but exceptional circumstances.
- MAY or OPTIONAL -- the item is truly optional and may be followed or ignored according to the needs of the implementor.

2. Introduction

Currently in the Internet, OSI protocols are being used more and more. As the network managers of an Internet once predominantly a TCP/IP network began deploying parts of the emerging OSI Internet, it became apparent that network layer OSI network debugging tools were almost nonexistent. When such tools existed, different implementations didn't work together.

As stated in RFC 1575, a simple network layer mechanism is necessary to allow systems to be probed to test network layer integrity. For the purposes of running OSI networks the authors of this document believe that other tools are necessary too. Other tools described below are an echo function, a traceroute function, and a routing table dump. What this document defines is the minimum subset of tools that are necessary to allow for the debugging of network problems.

3. Specification

This document's purpose is to specify a standard ping, traceroute, and OSI routing table dumping mechanisms for use for the ISO 8473 (CLNP) protocol in the OSI Internet. A detailed description of the specified mechanisms is below. These mechanism MUST be available on every router (inter mediate system) or host (end system) that provides OSI service for the Internet. These three functions are the basic tool set for the OSI network layer for the Internet.

3.1. Ping

3.1.1. Protocol Support

The long term echo mechanism, as described in 1575, requires the use of two new type values in the packet header of the ISO 8473 Network Protocol Data Units (NPDUs), or preferably packets. The two values are:

1E(hex) - for the echo-request Selector and,
1F(hex) - for the echo-response Selector.

Nodes which support ISO 8473 but do not support these two new values (for the type code option field in the header of an ISO 8473 packet) MUST send back an error packet if the ERROR report flag is set in the packet.

To support a ping function for ISO 8473, all end systems (hosts) and intermediate systems (routers) MUST support the "long term" echo function as defined by RFC 1575 [3] AND also set the ERROR report flag in the 8473 header.

The setting of the ERROR report flag is required because this allows a way for a compliant host or router to ping a non-compliant host or router. When a non-complaint host or router receives a "ping" packet with the new type function (Echo Request Selector), it MUST attempt to return an ISO 8473 error packet to the originating host, thus showing reachability.

3.1.2. Functions supported by the ping utility

A ping utility MUST be able to provide the Round trip time of each packet, plus the average minimum and maximum RTT over several ping packets. When an error packet is received by the node, the ping utility MUST report the error code to the user.

3.2. Traceroute

The CLNP trace is similar to the ping utility except that it utilizes the "Lifetime" field in the ISO 8473 packet. Hosts and routers that support OSI MUST also support CLNP trace. The "Lifetime" field serves the same function as the Time To Live (TTL) field does in an IP packet. A node (router or host) cannot forward ISO 8473 packet with a value for the Lifetime of zero. If the ERROR REPORT flag is set in the ISO 8473 packet, an error packet, will be returned to the originator of the packet.

3.2.1. Basic Traceroute

If a ISO 8473 echo-request packet is sent with "Lifetime" field value of 1, the first hop node (router or end system) will return an error packet to the originator the packet. If the first hop node supports the echo-request type field the error code will be either:

A0 (hex) - Lifetime Expired while Data Unit in Transit
A1 (hex) - Lifetime Expired during Re-assembly

If the first hop node does not support echo-request type field, the error code will be:

B0 (hex) - Unsupported Option not Specified.

When trying to trace a route to a remote node, the destination address in the echo-request packet sent should be this remote destination. By using increasing values in the "Lifetime" field a route can be traced through the network to the remote node. This traceroute function should be implemented on each system (host or router) to allow a user to trace a network path to a remote host or router.

The error message is used as evidence of the reachability and identity of the first hop. The originator then sends a packet with a "lifetime" field value of 2. The first hop decrements the "Lifetime" and because the "Lifetime" is still greater than 0, it forwards it on. The second hop decrements the "Lifetime" field value and sends an error packet (ER NPDU) with one of the two "Lifetime Expired" error codes listed above to the originator. This sequence is repeated until either:

- the remote host is reached an either an echo-response packet is sent back or (for nodes that do not have the required Echo support) an error packet is sent back, or

- the an error packet is received with error code (B0) indicating that a node will not pass the echo-response packet, or

- an error packet is received with one of the following errors:

80(hex) - Destination Address Unreachable
81(hex) - Destination Address Unknown.

If any of the following Error codes are received in an error packet, a second packet should be sent by the originating node:

CodeReason from 8473	
00(hex)	- Reason not specified
01(hex)	- Protocol procedure error
02(hex)	- Incorrect checksum
03(hex)	- Packet Discarded due to Congestion
04(hex)	- Header Syntax Error (cannot be parsed)
05(hex)	- Segmentation needed but not permitted
06(hex)	- Incomplete packet received
07(hex)	- Duplicate Option
B1(hex)	- Unsupported Protocol Version
B2(hex)	- Unsupported Security Option
B3(hex)	- Unsupported Source Routeing Option
B4(hex)	- Unsupported Recording of Route Option
C0(hex)	- Reassembly Interface

If one of these error is detected, an error value should be returned to the user. More than one echo packet, may be sent at a "Lifetime" value. The number of additional echo packets is left up to the implementation of this traceroute function.

If one of the following errors is received, AND "partial source route" is not specified in the echo-request packets, send a second echo-request packet to the destination at a "Lifetime" value:

Code	Reason from 8473
90(hex)	Unspecified Source Routeing Error
91(hex)	Syntax Error in Source Routeing Field
92(hex)	Unknown Address in Source Routeing Field
93(hex)	Path not Acceptable

(The echo-request packet may have been damaged while traversing through the network.)

3.2.2. Use of Partial Source route in traceroute

The current IP traceroute has a 3rd party or "loose source route" function. The ISO 8473 protocol also supports a "partial source routeing" function. However, if a node (router or host) does not support the "partial source routing" function an ISO 8473 packet gets passed along the path "exactly as though the function has not been selected. The packet shall not be discarded for this reason." [2]

In order utilize the partial source route function in the OSI traceroute, a node must set the "source routeing" option and "partial source routeing" parameter within that option. A 3rd party, or "loose source route" traceroute function requires that a node send an

echo-request packet with the "loose source routeing" field set. The functioning of the 3rd party/"loose source route" traceroute is the same except the following errors cause the traceroute to be terminated:

Code	Reason from ISO 8473
92	Unknown Address in Source Routeing Field
93	Path not Acceptable

These errors may indicate a problem with the "loose source route" listed in the echo-request packet for this destination. Additional packets with the same lifetime will only repeat this error. These errors should be reported to the user of the traceroute function.

3.2.3. Information needed from a Traceroute utility

A traceroute utility should provide the following information to the user:

- the identity of systems that comprise the path or route to the destination (the identifiers are called Network Entity Titles or NETs in OSI and ISO 8473)
- ping times (in Round trip times) for each hop in the path,
- error codes from error packet received as a response to the an echo-request packet, and
- any other error conditions encountered by traceroute.

3.3. OSI routing table dump

Each OSI host (end system) or router (intermediate system) MUST be able to dump any of its routing tables. Routing tables may come from the:

a.) the ES-IS information
b.) static
c.) IS-IS
d.) IDRP

or any other source.

Each system MUST be able to dump the routing table entries via some out of band mechanism. A method MUST exist to provide these. A show

osi routes command SHOULD be created with the following options:

- a for all routes
- esis for es-is routes
- isis for is-is routes
- idrp for idrp routes
- static for static routes
- other for routes from other sources.

In addition, routing tables SHOULD be available via either SNMP or CMIP. The specification of CMIP variables are outside the scope of this specification. Section 3.4 specifies the RFC 1238 MIB variables which MUST be available via SNMP. These two variables simply allow the user to get some basic CLNS routing information.

Please note that not all the information requested is available via the CLNS MIB. Due to this fact, it is anticipated that additional work on a CLNS MIB will be done in the future. When a new MIB is written, it is anticipated that this document will be updated to include the additional MIB variables to collect such things as the ES-IS cache.

3.4. MIB variables available via SNMP

The Simple Network Management Protocol [6] plays an important role in monitoring of multi-protocol, managed resources in the Internet. By convention, SNMP is mapped onto User Datagram Protocol (UDP), 6); however, in those situations where it is not possible to communicate with an ISO 8473 managed resource using SNMP over UDP, or where communication with an ISO 8473 managed resource using SNMP/UDP is not possible/appropriate, SNMP messages should be mapped onto an OSI transport (7) The following Managed Objects for the SNMP SHOULD be supported to facilitate remote monitoring using the SNMP:

The Simple Network Management Protocol (SNMP) plays an important role in monitoring of multi-protocol, managed resources in the Internet. By convention, SNMP is mapped onto User Datagram Protocol (UDP); however in those situations where it is not possible to communicate with an ISO 8473 managed resource using SNMP over UDP, or where communication with an ISO 8473 managed resource using SNMP/UDP is not possible/appropriate, SNMP should be mapped onto an OSI transport (8). The following Managed Objects SHOULD be supported for remoted monitoring using SNMP:

3.4.1. Summary of MIB Variables

RFC 1238 CLNS MIB [5]

1) clnpAddrTable - Addresses for Interfaces
2) clnpRoutingTable - OSI routes in system routing table.

3.4.2. ASN.1 Syntax for these MIB variables

The ASN.1 syntax for the two variables in CLNS MIB (RFC 1238) is included below for easy reference. That RFC remains the authoritative source for the MIB definitions.

1) clnpAddrTable

```
clnpAddrTable OBJECT-TYPE
object.id =  .... {clnp 21 }


clnpAddrTable = SEQUENCE OF ClnpAddrEntry
CLNPAddrEntry ::= SEQUENCE {
      clnpAdEntAddr
              CLNPAddres,
      clnpAdEntIfIndex,
              INTEGER,
      clnpAdEntReasmMaxSize
              INTEGER (0...65535);
      }

  clnpAdEntAddr = ClnpAddress
  clnpAddress = OCTET string (Size (1...20);
  clnpAdEntIfIndex = INTEGER;
  clnpAdEntReasmMaxSize = INTEGER (0...65535);   #
```

Descriptions of Table entry values:

```
clnpAdEntAddr - CLNP address for this interface value
clnpAdEntIfIndex - Interface Index value corresponding to
                   IfIndex value.
clnpAdEntReasmMaxSize = Maximum size of a pdu that can be
                        reassembled from incoming PDUs
                        received on this interface.
```

2) clnpRoutingTable

```
   object id =....{clnp 22}
   clnpRoutingTable =  SEQUENCE OF ClnpRouteEntry;
   ClnpRouteEntry = SEQUENCE OF {
                clnpRouteDest,
                clnpRouteIfIndex,
                clnpRouteMetric1,
                clnpRouteMetric2,
                clnpRouteMetric3,
                clnpRouteNextHop,
                clnpRouteType,
                clnpRouteProto,
                clnpRouteAge,
                clnpRouteInfo}

  clnpRoutDest ::= ClnpAddress;    # Address in Route table
                                   # (prefix or full address
  clnpRouteIfIndex ::= Integer;    # IfIndex value for
                                   # interface next hop can
                                   # be reached through.
  clnpRouteMetric1 ::= Integer;    # primary routing metric
                                   # for this protocol.
                                   # Specific meaning
                                   # depends on clnpRouteProto
                                   # value -1 if not used
  clnpRouteMetric2 ::= Integer;    # alternate routing metric
                                   # for this protocol.
                                   # Specific meaning
                                   # depends on clnpRouteProto
                                   # value -1 if not used
  clnpRouteMetric3 ::= Integer;    # alternate routing metric
                                   # for this protocol.
                                   # Specific meaning
                                   # depends on clnpRouteProto
                                   # value -1 if not used
  clnpRouteMetric4::= Integer;     # alternate routing metric
                                   # for this protocol.
                                   # Specific meaning
                                   # depends on clnpRouteProto
                                   # value -1 if not used
  clnpRouteNextHop::= ClnpAddress; # Address of Next Hop in
                                   # Routing
                                   # Table
  clnpRouteType::=INTEGER {
                other (1),         # none of following
                invalid (2),       # an invalid route
                direct(3),         # a direct route
```

```
                    remote(4)}          # a remote route

          clnprouteProto::= INTEGER {
                    other (1),          # none of the following
                                        # (manually configured
                                        # falls in this category)
                    local(2),           # configured entries
                    netmngt(3),         # set via Network
                                        # management
                    is-is(9),           # ISO 10589
                    ciscoIgrp(11),      # Ciscos OSI IGRP
                    ospf(13),           # OSPF set
                    bgp(14),            # BGP sets
                    idrp(15)            # addition suggested to
                                        # rfc 1238
                                        # in processing
          clnpRouteMetric5::= Integer;  # alternate routing metric
                                        # for this protocol.
                                        # Specific meaning
                                        # depends on clnpRouteProto
                                        # value -1 if not used
          clnpRouteInfo ::= OBJECT-ID;  # protocol id that
                                        # installed this route
                    }
```

4. OSI HOST.txt format

The OSI format for addresses allows addresses to be 20 bytes. In the long term, a Directory service (DNS service or OSI Directory service (X.500)), will provide a host name to address mapping. The process of getting OSI capable DNS and Directory service may require OSI pathway to already be set-up. Most host and router systems use a fixed table to provide this name to NSAP address mapping in order to get OSI working on their system. The current operational problem is each implementation has a different format. This document defines a fixed format so that these initial name to NSAP mapping files can be shared through-out the internet.

To conform to this document, a host or router supporting CLNS MUST have support a "osi host.txt" file with the format below. The "osi host.txt" file may be used for other OSI applications or TUBA applications. For these other applications, other fields may be defined but the definition of these is outside the scope of this specification.

OSI applications may use another file name for osi address information. NSAP addresses in any osi address information MUST use the format below. This host name to NSAP mapping MUST be available

for use by the following utilities on CLNS hosts and routers:

- OSI Echo (Ping) function,
- OSI traceroute function, and
- router table look-up for CLNS routing information

Host and router systems MUST also support a NSAP to name mapping by the Domain Name Service Directory or or the OSI Directory service (X.500).

Format of osi hosts file:

<NSAP Address> <name1> <name2> ...<name>

The NSAP Address should be in the following format:

<first octet>.<2nd octet 3rd octet>.<4th octet 5 octet>.

comments on the above format:

The NSAP octets should be expressed in hexidecimal. The dots are aids to help read the NSAP address, and MUST NOT be required for an NSAP address parsing. However, each NSAP address file MUST be able to have the ability to handle the insertion of dots. The location of the inserted dots within an NSAP address MUST NOT have any significance other than to make the address easier to read.

An example of this use in the GOSIP format is:

47.0005.80ff.ff00.0000.0001.0001.0a0b.0c0d.0204.00

An example of this format in ANSI format is:

39.480f.8000.0500.0000.0001.0001.0a0b0c0d.0204.00

This value quickly shows the AFI and the NSEL octets on either end.

<name1> <name2> <name> - Indicates a sequence of name associated with this nsap address.

5. Acknowledgements

The authors would like to acknowledge the contributions made by Dave Piscitello. He not only kept the document accurate, but also helped us to get rid of the ISO jargon and make the document more readable. Thanks to Paulina Knibbe for her work with the host.txt format. We would also like to thank members of the Network OSI Operations

Working Group of the IETF for their comments.

6. References

[1] ISO/IEC 8473, Information Processing Systems, "Protocol for Providing the Connectionless-mode Network Service and Provision of Underlying Service", May 1987.

[2] Hagens, R., "An Echo Function for ISO 8473", RFC 1139, IETF-OSI Working Group, January 1990.

[3] Hares, S., and C. Wittbrodt, "CLNP echo (ISO 8473)", RFC 1575, Merit/NSFNET, Stanford University/BARRNet, February 1994.

[4] ISO/IEC DIS 10747 Information Processing Systems - Telecommunications and Information Exchange between Systems - Protocol for Exchange of Inter-domain Routeing Information among Intermediate Systems to Support Forwarding of ISO 8473 packets.

[5] Satz, G., "Connectionless-mode Network Service Management Information Base - for use with Connectionless Network Protocol (ISO 8473) and End system to Intermediate System Protocol (ISO 9452)", RFC 1238, cisco Systems, Inc., June 1991.

[6] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, SNMP Research, Performance Systems International, Performance Systems International, MIT Laboratory for Computer Science, May 1990.

[7] Rose, M., "SNMP over OSI", RFC 1418, Dover Beach Consulting, Inc., March 1993.

[8] Information processing systems - Open Systems Interconnection - Protocol for Providing the Connectionless-mode Transport Service, International Organization for Standardization. International Standard 8602, December 1987.

7. Security Considerations

Security issues are not discussed in this memo.

8. Authors' Addresses

Susan K. Hares
MERIT/NSFNET
Internet Engineering
1075 Beal Avenue
Ann Arbor, MI 48109-2112

Phone: (313) 936-3000
EMail: skh@merit.edu

Cathy J. Wittbrodt
Stanford University/BARRNet
Networking Systems
Pine Hall 115
Stanford, CA 94305

Phone: (415) 725-5481
EMail: cjw@magnolia.Stanford.EDU