Network Working Group
Request for Comments: 2603
Category: Standards Track

M. Davison
Cisco Systems

June 1999

ILMI-Based Server Discovery for NHRP

Status of this Memo

Copyright Notice

Abstract

This memo defines how ILMI-based Server Discovery, which provides a method for ATM-attached hosts and routers to dynamically determine the ATM addresses of servers, shall be used to locate NHRP servers.

1. Introduction

Presently, configuring a host or router to use NHRP [1] is cumbersome and error-prone since it requires at least one ATM address to be statically configured on each host or router in the network. Further, it is impossible to implement a diskless host to use NHRP since local configuration is required. ILMI-based Server Discovery, hereafter referred to as "server discovery," provides a solution to these problems.

A brief overview of the Integrated Local Management Interface (ILMI) and the Service Registry MIB, as defined by the ATM Forum, are provided in this memo. The reader should consult [2] for a complete description of ILMI and this MIB, but the information contained here is sufficient for an understanding of its use to support NHRP server discovery.

2. Integrated Local Management Interface

The Integrated Local Management Interface (ILMI) [2] provides a mechanism for ATM-attached devices, such as hosts, routers, and ATM switches, to transfer management information. It is based on the Simple Network Management Protocol (SNMP), Version 1, and supports

get, get-next, set and trap operations.

The ILMI specification designates the switch side of the ATM link as the 'network side' and the host/router side of the ATM link as the ' user side.' The Service Registry MIB, which is outlined in Section 3, is implmented on the network side and is queried from the user side.

3. ILMI 4.0 Service Registry MIB

Server discovery utilizes the Service Registry MIB defined by the ATM Forum in ILMI Specification Version 4.0 [2]. To support the existing framework for IP over ATM, ATM switches must support the Service Registry MIB.

A row in the service registry table [2] is defined as:

```
AtmfSrvcRegEntry ::= SEQUENCE {
    atmfSrvcRegPort          INTEGER,
    atmfSrvcRegServiceID     OBJECT IDENTIFIER,
    atmfSrvcRegATMAddress    AtmAddress,
    atmfSrvcRegAddressIndex  INTEGER,
    atmfSrvcRegParm1         OCTET STRING
}
```

The definition of each field in this structure is:

atmfSrvcRegPort - The ATM port number for which this entry contains management information. The value of zero may be used to indicate the ATM interface over which a management request was received.

atmfSrvcRegServiceID - This is the service identifier that uniquely identifies the type of service at the address provided in the table. (See Section 3.2 for NHRP OID.)

atmfSrvcRegATMAddress - This is the full address of the service. The ATM client will use this address to establish a connection with the service.

atmfSrvcRegAddressIndex - An arbitrary integer to differentiate multiple rows containing different ATM addresses for the same service on the same port.

atmfSrvcRegParm1 - An octet string whose size and meaning is determined by the value of atmfSrvcRegServiceID.

The service registry table is indexed by atmfSrvcRegPort, atmfSrvcRegServiceID and atmfSrvcRegAddressIndex.

3.1 Service Parameter String

A generic parameter string is defined in the service registry table, thus allowing protocol-specific parameters to be specified. To be consistent with [1], the parameter string for NHRP shall be:

```
    ar$pro.type   16   bits     Protocol type
    ar$pro.snap   40   bits     Optional extension to protocol type
    ar$plen        8   bits     Length of protocol address
    ar$addr       plen octets   Network address
    ar$mask       plen octets   Network mask
```

Where

ar$pro.type - See [1]. (IPv4 is 0x0800, IPv6 is 0x86DD)

ar$pro.snap - See [1]. (IPv4 and IPv6 are 0)

ar$plen - Length of the protocol address. (IPv4 is 4, IPv6 is 16)

ar$addr - Network address represented in network byte order

ar$mask - Network mask represented in network byte order

3.2 Service Object Identifier

This OID, assigned in the ATM Forum Service Registry MIB, names ATMARP within the context of server discovery.

```
  atmfSrvcRegNHRP OBJECT IDENTIFIER ::= { 1.3.6.1.4.1.353.1.5.5 }
```

It does not name any managed objects, rather is used to locate appropriate rows in the service registery table.

4. Next Hop Client Behavior

An Next Hop Client NHC) will access the service registry table via ILMI using the SNMP GetNext operator to "sweep" (SNMP parlance for a linear search) beginning with {Port = 0, ServiceID = <see Section 3.2>, Index = 0} while holding the port number and the serviceID constant. (Port number 0 is used within ILMI to indicate "this port.")

An NHC with no local configuration, such as a diskless workstation,

must use the row with the lowest index value if multiple Next Hop Server (NHS), possibly for multiple networks, are listed.

NHC that have local IP configuration must use a row that has the appropriate IP address. For example, consider the case where an IP router has 3 logical interfaces defined on a single physical interface with IP addresses 1.0.0.1/8, 128.10.0.1/16 and 171.69.150.226/24. The router will sweep the service registry table looking for rows that have atmfSrvcRegParm1 values as shown below:

Net number/mask	atmfSrvcRegParm1
1.0.0.0/8	08 00 00 00 00 00 00 04 01 00 00 00 ff 00 00 00
128.10.0.0/16	08 00 00 00 00 00 00 04 80 0a 00 00 ff ff 00 00
171.69.150.0/24	08 00 00 00 00 00 00 04 ab 45 96 00 ff ff ff 00

When the correct atmfSrvcRegParm1 values are located, the router may then establish an SVC to the selected NHS and perform the appropriate protocol operations.

Redundant NHS are supported with multiple rows in the service registry table. This list of NHS is ordered with the primary NHS having the lowest index value. The NHC must attempt to utilize the primary NHS before utilizing a secondary NHS. Administrators must ensure that the listed NHS are synchronized.

5. NHRP Server (NHS) Behavior

A Next Hop Server (NHS) shall be locally configured. The NHS may retrieve the NHRP service registry data to validate the results. If an incorrect row is retrieved the error may be flagged in a locally significant way.

6. Relationship with PNNI Augmented Routing

An augmented version PNNI ("PNNI Augmented Routing," or PAR) [3] has been developed by the ATM Forum. PAR can distribute data such as NHS addresses. Further, the ATM Forum is developing a proxy mechanism for PAR (Proxy PAR) that would allow a UNI-attached host or router to access PAR data without a full PAR implementation.

These mechanisms offer a promising way to manage the service registry tables maintained on each switch in an ATM network, yet would not require changes to the mechanism defined in this memo. Hosts and routers can continue to utilize ILMI-based or Proxy PAR-based server discovery and network administrators could manage the service registry data with local configuration or via PAR and Proxy PAR.

7. Security Considerations

The server discovery mechanism is built on the ILMI managment framework and the security embodied in that framework. Access, to user- or network-side information is controlled by MIB design rather than protocol security mechanisms.

The service registery MIB, the table containing information for server discovery, is defined in [2] with read-only access. This means that any user-side device may query the service registry, but may not modify the service registry via ILMI. Instead, the sevice registry table must be modified via local configuration on the ATM switch.

References

[1] Luciani, J., et al., "NBMA Next Hop Resolution Protocol", RFC 2332, April 1998.

[2] ATM Forum, "Integrated Local Management Interface (ILMI) Specification Version 4.0," af-ilmi-0065.000, September 1996.

[3] ATM Forum, "PNNI Augmented Routing (PAR) Version 1.0," af-ra-0104, January 1999.

Author's Address

Mike Davison
Cisco Systems
170 West Tasman Drive
San Jose, California 95134

Phone: (408) 526-4000
EMail: mike.davison@cisco.com

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.