

The ATM Forum
Technical Committee

Multi-Protocol Over ATM
Version 1.1

AF-MPOA-0114.000

May, 1999

© 1999 by The ATM Forum. This specification/document may be reproduced and distributed in whole, but (except as provided in the next sentence) not in part, for internal and informational use only and not for commercial distribution. Notwithstanding the foregoing sentence, any protocol implementation conformance statements (PICS) or implementation conformance statements (ICS) contained in this specification/document may be separately reproduced and distributed provided that it is reproduced and distributed in whole, but not in part, for uses other than commercial distribution. All other rights reserved. Except as expressly stated in this notice, no part of this specification/document may be reproduced or transmitted in any form or by any means, or stored in any information storage and retrieval system, without the prior written permission of The ATM Forum.

The information in this publication is believed to be accurate as of its publication date. Such information is subject to change without notice and The ATM Forum is not responsible for any errors. The ATM Forum does not assume any responsibility to update or correct any information in this publication. Notwithstanding anything to the contrary, neither The ATM Forum nor the publisher make any representation or warranty, expressed or implied, concerning the completeness, accuracy, or applicability of any information contained in this publication. No liability of any kind shall be assumed by The ATM Forum or the publisher as a result of reliance upon any information contained in this publication.

The receipt or any use of this document or its contents does not in any way create by implication or otherwise:

- Any express or implied license or right to or under any ATM Forum member company's patent, copyright, trademark or trade secret rights which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor
- Any warranty or representation that any ATM Forum member companies will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor
- Any form of relationship between any ATM Forum member companies and the recipient or user of this document.

Implementation or use of specific ATM standards or recommendations and ATM Forum specifications will be voluntary, and no company shall agree or be obliged to implement them by virtue of participation in The ATM Forum.

The ATM Forum is a non-profit international organization accelerating industry cooperation on ATM technology. The ATM Forum does not, expressly or otherwise, endorse or promote any specific products or services.

NOTE: The user's attention is called to the possibility that implementation of the ATM interoperability specification contained herein may require use of an invention covered by patent rights held by ATM Forum Member companies or others. By publication of this ATM interoperability specification, no position is taken by The ATM Forum with respect to validity of any patent claims or of any patent rights related thereto or the ability to obtain the license to use such rights. ATM Forum Member companies agree to grant licenses under the relevant patents they own on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. For additional information contact:

The ATM Forum
Worldwide Headquarters
2570 West El Camino Real, Suite 304
Mountain View, CA 94040-1313
Tel: +1-650-949-6700
Fax: +1-650-949-6705

Acknowledgments

Much work went into the development of the specification of MPOA versions 1.0 and 1.1. They could not have been completed without the ATM Forum Contributions and participation in the working group by many people. In particular, the Editor would like to recognize the following members who made significant contributions to this effort:

Cedell Alexander
Loa Andersson
Grenville Armitage
Scott Brim
Caralyn Brown
Geetha M. Brown
Ross Callon
Andrew Carter
Michael Craren
Joan Cucchiara (MPOA MIB Editor)
John Drake
Rob Enns
Norm Finn
Barbara Fox
Andre N. Fredette (MPOA 1.0 Editor)
Martha Gailus
Eric Gray
Bryan Gleeson
Joel M. Halpern
Dave Husak
John Keene
Ali Kujoory
James Luciani
Keith McCloghrie
Drew Perkins
Andrew Smith
Matt Squire (LANE/MPOA Chairman)
Vijay Srinivasan
Hiroshi Suzuki
George Swallow
James Watt

The assistance by these members and all who participated in the MPOA Sub-working Group is greatly appreciated.

Bernhard Petri, Editor

Contents

1. INTRODUCTION	11
1.1 What is MPOA?	11
1.2 Services Required by MPOA	12
1.3 Relation between MPOA 1.1 and MPOA 1.0	12
1.3.1 Errata and PICS Proforma	12
1.3.2 Functional Enhancements to MPOA 1.0 by MPOA 1.1	12
1.3.3 Integration of the MPOA 1.1 Management Information Base (MPOA 1.1 MIB)	12
2. TERMS AND DEFINITIONS	13
2.1 Definitions	13
2.2 Acronyms and Abbreviations	15
2.3 Normative Statements	17
2.4 List of Symbols used in Annex D (PICS Proforma)	17
3. MPOA DESCRIPTION	18
3.1 MPOA Components	18
3.1.1 MPOA Client (MPC)	18
3.1.2 MPOA Server (MPS)	18
3.1.3 Examples of MPOA Enabled Devices	18
3.1.4 Relationship Between LECs, MPOA Components, and MPOA Devices	19
3.2 Control and Data Flows	19
3.2.1 MPOA Control Flows	20
3.2.1.1 Configuration Flows	20
3.2.1.2 MPC-MPS Control Flows	21
3.2.1.3 MPS-MPS Control Flows	21
3.2.1.4 MPC-MPC Control Flows	21
3.2.2 MPOA Data Flows	21
3.2.2.1 MPC-MPC Data Flow	21
3.2.2.2 MPC-NHC Data Flows	21
3.3 MPOA Operations	21
3.3.1 Configuration	21
3.3.2 Discovery	21
3.3.3 Target Resolution	22
3.3.3.1 Ingress MPC Perspective	22
3.3.3.2 Ingress MPS Perspective	22
3.3.3.3 Egress MPS Perspective	22
3.3.3.4 Egress MPC Perspective	23
3.3.4 Connection Management	23
3.3.5 Data Transfer	23
3.4 Routing Protocol Interaction	23

3.5	NHRP/ION Interaction	24
3.6	A Day in the Life of a Data Packet	24
4.	MPOA SPECIFICATION	26
4.1	Configuration Parameters	26
4.1.1	MPS Configuration	26
4.1.1.1	MPS Parameters	26
4.1.1.2	MPS Constants	26
4.1.2	MPC Configuration	27
4.1.2.1	MPC Parameters	27
4.1.2.2	MPC Constants	27
4.2	Device Discovery	27
4.2.1	Register Protocol	28
4.2.2	Address Resolution Protocol	28
4.2.3	Implications for Co-Located MPS, MPC and Non-MPOA Devices	28
4.2.4	Change of Device Status	29
4.3	MPOA Retry Mechanism	29
4.4	Detailed MPC Behavior	30
4.4.1	MPC Configuration	32
4.4.2	Inbound Data Flow	32
4.4.3	Outbound Data Flow	33
4.4.4	Cache Management	34
4.4.4.1	Ingress Cache Entry Creation and Management	34
4.4.4.2	Egress Cache Entry Creation and Management	35
4.4.5	LAN-to-LAN Flows Within the Same MPOA Device	35
4.4.6	Control Information in MPC Caches	36
4.4.6.1	Ingress Cache	36
4.4.6.2	Egress Cache	37
4.5	Detailed MPS Behavior	38
4.5.1	MPS Configuration	38
4.5.2	MPOA Resolution and NHRP Resolution	39
4.5.2.1	Translating MPOA Resolution Requests to NHRP Resolution Requests	40
4.5.2.2	Translating NHRP Resolution Requests to MPOA Cache Imposition Requests	40
4.5.2.3	Translating MPOA Cache Imposition Replies to NHRP Resolution Replies	40
4.5.2.4	Translating NHRP Resolution Replies to MPOA Resolution Replies	41
4.5.2.5	MPS to MPS NHRP	41
4.6	Keep-Alive Protocol	41
4.7	Cache Maintenance	42
4.7.1	Egress Cache Maintenance	42
4.7.1.1	Egress MPS Purges and Cache Updates	42
4.7.1.2	Egress MPC Invalidation of Imposed Cache Entries	43
4.7.1.3	Invalidation of State Information Relative to Imposed Cache	43
4.7.1.4	Recovery From Receipt of Invalid Data Packets	44
4.7.1.5	Egress Encapsulation	44
4.7.1.6	MPC-Initiated Egress Cache Purge	44
4.7.2	Ingress Cache Maintenance	45
4.7.2.1	MPOA Trigger	45

4.7.2.2	Ingress MPSs and NHRP Purges	45
4.7.2.3	Data Plane Purge Protocol	45
4.8	Connection Management	46
4.8.1	Generic VCC Management Procedures	46
4.8.2	Scope of MPOA VCCs	47
4.8.3	Initiating VCCs	47
4.8.4	Receiving Incoming VCCs	47
4.8.5	Support for Multiple VCCs	48
4.8.6	Internetwork Layer-to-ATM Address Mapping	48
4.8.7	Establishment of Bi-directional Data Flow	48
4.8.8	VCC Termination	48
4.8.9	Use of UNI Signaling Information Elements	49
4.8.9.1	Traffic Descriptor	49
4.8.9.2	QoS Parameter	51
4.8.9.3	AAL5 Parameters	51
4.8.9.4	B-LLI	53
4.8.9.5	Broadband Bearer Capability	53
4.8.9.6	ATM Addressing information	54
5.	MPOA FRAME FORMATS	56
5.1	Encapsulation	56
5.1.1	Data Frame Encapsulation	56
5.1.2	Control Frame Encapsulation	56
5.2	LANE TLVs	57
5.2.1	MPS Configuration TLVs	57
5.2.2	MPC Configuration TLVs	57
5.2.3	Device Type TLV	58
5.3	Frame Formats	59
5.3.1	MPOA CIE Codes	60
5.3.2	Control Message Format	60
5.3.2.1	Fixed Header	60
5.3.2.2	Common Header	61
5.3.2.3	Client Information Element	61
5.3.2.4	Extensions	62
5.3.3	MPOA Resolution Request Format	65
5.3.4	MPOA Resolution Reply Format	66
5.3.5	MPOA Cache Imposition Request Format	67
5.3.6	MPOA Cache Imposition Reply Format	69
5.3.7	MPOA Egress Cache Purge Request Format	70
5.3.8	MPOA Egress Cache Purge Reply Format	71
5.3.9	MPOA Keep-Alive Format	72
5.3.10	MPOA Trigger Format	72
5.3.11	NHRP Purge When Used on the Data Plane	73
5.3.12	NHRP Ingress Cache Purge Request Format	74
5.3.13	NHRP Ingress Cache Purge Reply Format	75
5.3.14	MPOA Error Indication Format	75
6.	REFERENCES	77
Annex A.	Protocol-Specific Considerations	78

A.1	IP Packet Handling in MPOA	78
A.1.1	Requirements	78
A.1.2	Encapsulation	78
A.1.3	MPS Role	80
A.1.4	Ingress MPC Role	80
A.1.5	Egress MPC Role	81
A.2	IPX Packet Handling in MPOA	81
A.2.1	Requirements	81
A.2.2	Encapsulation	81
A.2.3	MPS Role	82
A.2.4	Ingress MPC Role	82
A.2.5	Egress MPC Role	83
Annex B.	MPOA Request/Reply Packet Contents	84
B.1	Ingress MPC-Initiated MPOA Resolution	84
B.2	Egress MPC-Initiated Egress Cache Purge	85
B.3	Egress MPS-Initiated Egress Cache Purge	86
B.4	Data-Plane Purge	87
B.5	MPOA Trigger	87
B.6	MPOA Keep-Alive	88
Annex C.	NBMA Next Hop Resolution Protocol (NHRP)	89
Annex D: MPOA 1.1 PICS Proforma		90
D.1.	Introduction	90
D.1.1.	Scope	90
D.1.2.	Normative references	90
D.1.3.	Definitions	90
D.1.4.	Acronyms	90
D.1.5.	Conformance	90
D.2.	Instructions	91
D.2.1.	Instructions for completing the PICS proforma	91
D.2.2.	Additional Information	91
D.2.3.	Exception Information	91
D.2.4.	Legend for the columns of the PICS proforma tables	91
D.2.5.	Legend for further indications of the PICS proforma tables	92
D.3.	Identification of the implementation	93
D.3.1.	Date of statement	93
D.3.2.	Identification of the implementation	93
D.3.3.	Identification of the system in which it resides	93
D.3.4.	Product supplier	93
D.3.5.	Customer	94
D.3.6.	PICS contact person	94
D.3.7.	Protocol for which this PICS applies	94
D.4.	Global Statement of Conformance	95
D.5.	Roles	96
D.6.	Additional Specifications implemented to support ATM Forum MPOA Vs. 1.1	96
D.7.	MPC Generic Protocol Handling	97
D.8.	MPC Configuration	98
D.9.	MPC Device Discovery	99
D.10.	MPC Target Resolution	100
D.10.1.	MPC Data Path Processing	100
D.10.2.	MPC Cache Management	101
D.10.3.	MPC Cache Maintenance	102
D.11.	MPC Connection Management	103
D.12.	MPC Data Transfer	104

D.13.	MPS Configuration	105
D.14.	MPS Device Discovery	106
D.15.	MPS Connection Management	107
D.16.	MPS Generic Protocol Handling	108
D.16.1.	Keep-Alive Protocol	108
D.16.2.	Cache Maintenance	109
D.16.3.	Trigger	110
D.17.	MPS and NHRP Relationship	111
D.18.	MPS Data Transfer	111
D.19.	Co-Located MPS and MPC	111
D.20.	Frame Formats	112
D.21.	IP Packet Handling	113
D.22.	IPX Packet Handling	114
D.23.	MPOA Authentication	114
D.24.	MPOA MIB	115
 Annex E. MPOA 1.0 Errata List		 117
E.1 Introduction to MPOA Errata		117
E.2 References		117
E.3 Clarifications		118
E.3.1.	Section 2.1 (add an additional clarifying definition of the term "NHRP Purge")	118
E.3.2.	Sections 4.2 through 4.2.2 (various editorial clarifications, also on "every time they are sent")	118
E.3.3.	Paragraph 5 of Section 4.2.3	119
E.3.4.	7th Line of 8th (=second to last) Paragraph of Section 4.4.2	119
E.3.5.	Section 4.7.1.4, 2nd line (clarification on Data Plane Purge)	119
E.3.6.	Section 4.7.1.6, Paragraph 1, 2nd line (clarification that action is mandatory)	119
E.3.7.	Section 4.7.1.6, Paragraph 3, after 1 st sentence (addition on no-reply flag)	119
E.3.8.	Section 4.7.2.2, Ingress MPSs and NHRP Purges	119
E.3.9.	MPOA Keep-Alive Lifetime Extension Format of Section 5.3.2.4.4	119
E.3.9.1.	Proposed Diagram	120
E.3.10.	Section 5.3.2.4.6 MPOA Original Error Code Extension	120
E.3.11.	Appendix IV.2, last paragraph (Clarification related to Tag Extension)	120
 E.4 Corrections		 121
E.4.1.	Last Paragraph in Section 3.6	121
E.4.2.	Figure 7 in Section 4.4	121
E.4.3.	Last Line of Paragraph 1 of Section 4.4.4.1	121
E.4.4.	Last Line of Last Paragraph of Section 4.4.4.2	121
E.4.5.	1st Paragraph of Section 4.4.6.1.4	122
E.4.6.	1st Bullet Item in Paragraph 2 of Section 4.4.6.1.4 (replace "should" by "must")	122
E.4.7.	Paragraph 3 of Section 4.4.6.1.4 (Typo)	122
E.4.8.	Paragraphs 3, 4 and 5 of Section 4.5 - 5 Typos	122
E.4.9.	Section 4.5.2.2	122
E.4.10.	Paragraph 2 of Section 4.5.2.4 - incomplete specification	122
E.4.11.	Bulleted List in Section 4.7.1.6	123
E.4.12.	MPOA Trigger Inconsistency	123
E.4.12.1.	Last paragraph of Section 4.7.2.1	123
E.4.12.2.	Section 5.3.10	123
E.4.12.3.	Section B.5 MPOA Trigger	124
E.4.13.	Paragraph 3 of Section 4.7.2.1 (correct reference)	124
E.4.14.	Paragraph 2 of Section 4.7.2.3 - E-MPC not forced to send purges when an MPS fails	124
E.4.15.	Paragraph 2 of Section 4.8.9	124
E.4.16.	Paragraph 1 of Section 5.2.3	125

E.4.17.	Paragraph 4 of Section 5.3.3 - "Largest" should be "Widest"	125
E.4.18.	Section 5.3.2.4.7 (MPOA Authentication Extension)	125
E.4.19	Section 5.3.7 (MPOA Egress Cache Purge Request Format)	125
E.4.20	Section 5.3.11, Fixed Header of NHRP Purge (Data Plane)	125
E.4.21	Paragraph 4 of Section 5.3.11	126
E.4.22	Sections 5.3.12 and 5.3.13	126
E.4.23	Section 5.3.14	126
E.4.24	Section 6 (References) (apply appropriate RFC References for IPOA and NHRP)	126
E.4.25	Note (2) Section B.1	126
E.4.26	Request phase of Egress MPC-Initiated Cache Purge	127
E.4.26.1	Section B.2	127
E.4.26.2	Section 4.7.1.6	127
E.4.27	Reply phase of Egress MPC-Initiated Cache Purge	128
E.4.28	Section B.3 - Egress MPS-Initiated Cache Purge	128
E.4.29	Annex C (replace cover sheet and copy of version 11 of the draft NHRP by the RFC reference)	129
E.4.30	Annex D (PICS Proforma))	129
E.4.31	Figure 23 - MPC-c4 through MPC-c7 changed to MPC-c3 through MPC-c6 and add New_MPS event generation	130
E.4.32	Figure 24 - MPS-p4 replaced by MPS-p6 c6 and add New_MPC event generation	131
E.4.33	Figure 27 - MPC-c4 through MPC-c7 changed to MPC-c3 through MPC-c6	132
E.4.34	Apx I, Title I.7 - Applies to both Egress and Ingress ("Egress" removed from beginning)	132
E.4.35	Appendix V - add two more entries	132
E.4.36	Appendix VII - Tip Sheet, also illustrating corrections	132
E.5.	Corrections to MPOA 1.0 MIB	134
E.5.1	Summary of Changes related to MPS Multinetting	134
E.5.2	Correction of a Syntax Error for Index Item "mpsEgressCacheId"	137
Annex F:	MPOA 1.1 MIB	139
Appendix I.	State Machine View of MPOA Component Behavior	208
I.1	Conventions	208
I.2	Ingress MPC Control State Machine	208
I.3	Ingress MPS Control State Machine	209
I.4	Egress MPS Control State Machine	210
I.5	Egress MPC Control State Machine	211
I.6	Reliable Delivery State Machines	212
I.7	MPC and MPS Keep-Alive State Machines	212
Appendix II.	Examples of MPOA Control and Data Flows	214
II.1	Scenarios	214
II.1.1	Intra-ELAN Scenarios	214
II.1.2	Inter-ELAN Scenarios	215
II.2	Flows	215
II.2.1	Intra-ELAN	215
II.2.2	Inter-ELAN	218
Appendix III.	Related Work	223
III.1	LANE	223
III.2	Classical IP	223
III.3	MARS	223
III.4	RFC 1483	224
Appendix IV.	Ambiguity at the Edge	225
IV.1	Ambiguous Encapsulation Information At The Egress MPC	225
IV.2	Resolving Egress Ambiguity	225

IV.3	Ambiguity At The Ingress	225
Appendix V MPOA-friendly NHRP implementations		226
Appendix VI. MPOA Requirements for Co-Located LEC		227
VI.1	Support MPOA Device Type TLV Association	227
VI.2	Support for LECs that do Source Learning	227
VI.3	Support for LLC Multiplexing	227
Appendix VII MPOA 1.1 Tip Sheet		228
VII.1	MPS Multinetting	228
VII.2	Address Indications in NHRP Ingress Cache Purge	229

1. Introduction

[Informative]

Internetwork layer protocols such as IP, IPX and AppleTalk use routers to allow communication across subnet boundaries. Subnets are often built using LAN technologies, Ethernet and Token Ring being the most popular.

The ATM Forum's LAN Emulation LANE provides Emulated LANs (ELANs) that emulate the services of Ethernet and Token Ring LANs across an ATM network. LANE provides many benefits including interoperation with Ethernet and Token Ring hardware and software, allowing a subnet to be bridged across an ATM/LAN boundary. LANE allows a single ATM network to support multiple ELANs. By using ELANs, internetwork layer protocols may operate over an ATM network in essentially the same way that they operate over Ethernet and Token Ring LANs. While LANE provides an effective means for bridging intra-subnet data across an ATM network, inter-subnet traffic still needs to be forwarded through routers.

The IETF Internetworking Over NBMA Networks (ION) Working Group's Next Hop Resolution Protocol (NHRP) [NHRP] and Multicast Address Resolution Server (MARS) [MARS] protocols also allow internetwork layer protocols to operate over an ATM network. These protocols allow the ATM network to be divided into *ION Subnets*, also known as Logical IP Subnets (LISs) or Local Address Groups (LAGs). Routers are required to interconnect these subnets, but NHRP allows intermediate routers to be bypassed on the data path. NHRP provides an extended address resolution protocol that permits Next Hop Clients (NHCs) to send queries between different subnets. Queries are propagated by Next Hop Servers (NHSs) along the routed path as determined by standard routing protocols. This enables the establishment of ATM VCCs across subnet boundaries, allowing inter-subnet communication without requiring routers in the data path.

Even with both LANE and NHRP, a common situation exists where communicating LAN devices are behind LANE edge devices. MPOA allows these edge devices to perform internetwork layer forwarding and establish direct communications without requiring that the LANE edge devices be full function routers.

1.1 What is MPOA?

The goal of MPOA is the efficient transfer of inter-subnet unicast data in a LANE environment. MPOA integrates LANE and NHRP to preserve the benefits of LAN Emulation, while allowing inter-subnet, internetwork layer protocol communication over ATM VCCs without requiring routers in the data path. MPOA provides a framework for effectively synthesizing bridging and routing with ATM in an environment of diverse protocols, network technologies, and IEEE 802.1 Virtual LANs. This framework is intended to provide a unified paradigm for overlaying internetwork layer protocols on ATM. MPOA is capable of using both routing and bridging information to locate the optimal exit from the ATM cloud.

MPOA allows the physical separation of internetwork layer route calculation and forwarding, a technique known as virtual routing. This separation provides a number of key benefits:

1. It allows efficient inter-subnet communication;
2. It increases manageability by decreasing the number of devices that must be configured to perform internetwork layer route calculation;
3. It increases scalability by reducing the number of devices participating in internetwork layer route calculation;
4. It reduces the complexity of edge devices by eliminating the need to perform internetwork layer route calculation.

MPOA provides MPOA Clients (MPCs) and MPOA Servers (MPSs) and defines the protocols that are required for MPCs and MPSs to communicate. MPCs issue queries for shortcut ATM addresses and receive replies from the MPS using these protocols.

MPOA also ensures interoperability with the existing infrastructure of routers. MPOA Servers make use of routers that run standard internetwork layer routing protocols, such as OSPF, providing a smooth integration with existing networks.

1.2 Services Required by MPOA

1. ATM Signaling [UNI 3.0, UNI 3.1, or UNI 4.0].
2. LANE 2.0 [LANE] (as defined in Section 3, Appendix D [LANE]).
3. Next Hop Resolution Protocol [NHRP].

1.3 Relation between MPOA 1.1 and MPOA 1.0

1.3.1 Errata and PICS Proforma

The MPOA 1.0 specification of the ATM Forum was published in July 1997 as AF-MPOA-0087.000. Since that time a number of Errata in MPOA 1.0 were identified (see overview of all identified Errata in Annex E of this specification), and a PICS Proforma was developed (see Annex D below).

Rather than publishing these as one or more separate MPOA-related specifications, it was decided to publish this version 1.1 of the full MPOA specification, in order to avoid any inconveniences with an Errata document separate from the main specification it refers to, and also to facilitate the provision of PICS for implementations claiming conformance to MPOA.

In addition, since the official RFC version of the IETF NBMA Next Hop Resolution Protocol (NHRP) has meanwhile been made available by IETF as RFC 2332, it has now become possible to replace the previous Annex C of MPOA 1.0 by just a reference to RFC 2332. The previous Annex C of MPOA 1.0 had reproduced the close-to-final draft version 11 of NHRP.

1.3.2 Functional Enhancements to MPOA 1.0 by MPOA 1.1

Additional Functions of MPOA 1.1 compared to MPOA 1.0:

- NHRP Authentication Extension

Since the RFC version of NHRP (RFC 2332) - unlike the draft version 11, reproduced as Annex C of MPOA 1.0 - provides for an optional authentication extension (see section 5.3.4 of [NHRP]) to convey authentication information between NHRP speakers, MPOA 1.1 inherits this optional extension for its MPSs by referring to the RFC version rather than draft version 11 of NHRP.

- MPOA Authentication Extension

In order to allow for an extension of the NHRP authentication environment to the MPCs, MPOA 1.1 also specifies a similar optional MPOA-specific authentication mechanism for the communication between MPCs and MPSs. Also, an MPOA Error Indication is added to allow for the transfer of an authentication failure indication, if an authentication test fails.

1.3.3 Integration of the MPOA 1.1 Management Information Base (MPOA 1.1 MIB)

The MPOA 1.0 MIB had been approved as af-mpoa-0092.000 [MPOA 1.0 MIB] in July 1998. In principle, this MIB specification is valid for both MPOA 1.0 and MPOA 1.1. However, after the approval of af-mpoa-0092.000, it was noted that the MPOA 1.0 MIB does not allow a particular configuration for "MPS multinetting" (see detailed description in Appendix VII) which is not precluded by MPOA 1.0, and that it also contained a minor syntax error.

It was therefore agreed to produce a new version of this MIB with these errors corrected, and to integrate it into this MPOA 1.1 specification as Annex F. Section E.5 provides an overview of the changes / corrections to the MPOA 1.0 MIB.

2. Terms and Definitions

[Informative]

2.1 Definitions

Control ATM Address:	The address used to set up an SVC to send control packets to an MPOA component. Each MPOA Component has a single Control ATM Address. The Control ATM address may be different from the Data ATM Address.
Control Flow:	A bi-directional flow of Control Messages between two MPOA Components.
Control Messages:	NHRP and MPOA messages, and any other non-data message used by an MPOA Component.
Data ATM Address:	An ATM address used to set up a shortcut. This address may be different from the Control ATM Address.
Data Flow:	A uni-directional flow of data packets to a single destination Internetwork Layer Address.
Data Plane Purge	An NHRP Purge Message sent on the data plane (i.e. a shortcut) by an MPC.
Default Path:	The hop-by-hop path between Routers that a packet would take in the absence of shortcuts, as determined by routing protocols.
DLL Header:	All headers before the Internetwork Layer packet. For example, an Ethernet frame DLL Header includes the Destination MAC Address, the Source MAC Address, and the Ethertype or length and 802.2 LLC/SNAP information.
Edge Device:	A physical device capable of bridging packets between one or more LAN interfaces and one or more LAN Emulation Clients. An Edge Device also contains one or more MPOA Clients allowing it to forward packets across subnet boundaries using an Internetwork Layer protocol.
Egress:	The point where an Outbound Data Flow exits the MPOA System.
Egress Cache:	The collection of Egress Cache Entries in an MPC.
Egress Cache Entry:	Information describing how Internetwork Layer packets from a particular Shortcut are to be encapsulated and forwarded.
Egress MPC:	An MPC in its role at an Egress.
Egress MPS:	The MPS serving an Egress MPC for a particular Outbound Data Flow.
Emulated LAN:	See [LANE].
Flow:	A stream of packets between two entities. Multiple flows may be multiplexed over a single VCC.
Higher Layers:	The software stack above MPOA and LANE, e.g. LLC, bridging, etc.
Inbound Data Flow:	A Data Flow entering the MPOA System.
Inbound Flow:	Data entering the MPOA System.
Ingress:	The point where an Inbound Data Flow enters the MPOA System.
Ingress Cache:	The collection of Ingress Cache Entries in an MPC.
Ingress Cache Entry:	The collection of information dealing with inbound data flows. This information is used to detect flows that may benefit from a shortcut, and, once detected, indicates the shortcut VCC to be used and encapsulation information to be used on the frame.
Ingress MPC:	An MPC in its role at an Ingress.

Ingress MPS:	The MPS serving an Ingress MPC for a particular Inbound Data Flow.
Internetwork Layer:	The protocols and mechanisms used to communicate across subnet boundaries. E.g., IP, IPv6, IPX, DECnet routing, CLNP, AppleTalk DDP, Vines, SNA, etc.
LANE Service Interface:	The interface over which a LEC communicates with an MPC.
MPC Service Interface:	The interface over which an MPC communicates with the Higher Layers.
MPOA Client (MPC):	A protocol entity that implements the client side of the MPOA protocol.
MPOA Component:	An MPC or MPS.
MPOA Device:	A physical device (e.g., router, bridge, or host) that contains one or more MPOA components.
MPOA Host	A host containing one or more LAN Emulation Clients allowing it to communicate using LAN Emulation. An MPOA Host also contains one or more MPOA Clients allowing it to transmit packets across subnet boundaries using an Internetwork Layer protocol.
MPOA Server (MPS):	A protocol entity that implements the server side of the MPOA protocol. An MPOA Server is co-located with a Router.
MPOA System	The set of inter-communicating MPOA Clients and MPOA Servers.
NHRP Purge	Within the context of MPOA, this term refers to an MPOA purge mechanism where the packet type values of NHRP purge are used (see sections 5.3.11, 5.3.12 and 5.3.13 below).
Outbound Data Flow:	A Data Flow exiting the MPOA System.
Outbound Flow:	Data exiting the MPOA System from a Shortcut.
Protocol Address	An internetwork layer address.
Protocol Data Unit (PDU)	A message sent between peer protocol entities.
Router	A device allowing communication across subnet boundaries using an Internetwork Layer protocol. A Router maintains tables for Internetwork Layer packet forwarding and may participate in one or more Internetwork Layer routing protocols for this purpose. A Router forwards packets between subnets in accordance with these tables. When referred to in this specification, a Router contains , one or more LAN Emulation Clients, one or more MPOA Servers, one or more Next Hop Servers, zero or more Next Hop Clients, and zero or more MPOA Clients.
Routing Protocol:	A protocol run between Routers to exchange information used to allow computation of routes. The result of the routing computation will be one or more next hops.
Service Data Unit (SDU)	A message sent between an entity and its service user or service provider.
Shortcut	An ATM VCC used to forward data packets in lieu of the default routed path.
Target:	An Internetwork Layer Address to which a Shortcut is desired.
Tag:	A 32 bit opaque pattern that an Egress MPC may provide to an Ingress MPC. If a Tag is provided to an Ingress MPC by an Egress MPC, the Ingress MPC must include the tag in the MPOA packet header for packets sent to the given MPC for the given internetwork destination.

2.2 Acronyms and Abbreviations

af	ATM Forum
ARP	Address Resolution Protocol
B-LLI	Broadband Low Layer Information
BCOB-C	Broadband Bearer Connection Oriented Service Type C
BCOB-X	Broadband Bearer Connection Oriented Service Type X
BUS	Broadcast and Unknown Server
CIE	NHRP Client Information Element
CPCS-PDU	Common Part Convergence Sub-layer Protocol Data Unit
CPCS-SDU	Common Part Convergence Sub-layer - Service Data Unit
DLL	Data Link Layer
ELAN	Emulated LAN
E-MPC	Egress MPC
E-MPS	Egress MPS
IE	Information Element
IETF	Internet Engineering Task Force
I-MPC	Ingress MPC
I-MPS	Ingress MPS
InATMARP	Inverse ATM Address Resolution Protocol
ION	Internetworking Over NBMA (Non-Broadcast Multi-Access)
IP	Internet Protocol
IPX	Internetwork Packet Exchange
L3	Internetwork Layer
LANE	LAN Emulation
LE __	LAN Emulation __
LEC	LAN Emulation Client
LECS	LAN Emulation Configuration Server
LES	LAN Emulation Server
LIS	Logical IP Subnet
LLC	Logical Link Control
LUNI	LAN Emulation User-Network Interface
MAC	Media Access Control (see e.g. IEEE 802.3 or IEEE 802.5)
MARS	Multicast Address Resolution Server
MPC	MPOA Client
MPOA	Multiprotocol Over ATM
MPS	MPOA Server
MTU	Maximum Transmission Unit

N/A	Not applicable
NAK	Negative NHRP Resolution Reply
NBMA	Non-Broadcast Multi-Access (e.g. ATM, Frame Relay)
NHC	Next Hop Client
NHRP	Next Hop Resolution Protocol
NHS	Next Hop Server
NLSP	NetWare Link State Protocol
OSPF	Open Shortest Path First
PCR	Peak Cell Rate
PICS (proforma)	Protocol Implementation Conformance Statement (proforma)
PICS (proforma) item	- A row in a PICS (proforma) table
PDU	Protocol Data Unit
QoS	Quality of Service
RIP	Routing Information Protocol
RSVP	Resource ReSerVation Protocol
SCSP	Server Cache Synchronization Protocol
SDU	Service Data Unit
SNAP	SubNetwork Attachment Point
Status (Value)	In a PICS proforma, an allowed entry in the status column for an item in a table
Support (Answer)	In a PICS proforma, an allowed entry in the support or supported values columns for an item in a PICS (proforma) question
SVC	Switched Virtual Channel Connection
TLV	Type-Length-Value Encoding
TTL	Time To Live
UBR	Unspecified Bit Rate
v1.0	Version 1.0
VCC	Virtual Channel Connection

2.3 Normative Statements

The normative sections of this specification are Section 4, Section 5, and all Annexes. Throughout these normative sections, normative statements are used as follows:

Table 1. Normative Statements

Statement	Verbal Form
Requirement	must/must not
Recommendation	should/should not
Permission	may

The term “may” is used to indicate that a particular procedure is allowed but not required. It is an implementation choice. “may” is also used to indicate allowed behaviors that must be accommodated.

Annex D (PICS Proforma) provides further information on the status of normative statements.

2.4 List of Symbols used in Annex D (PICS Proforma)

]	logical symbol "NOT" (see section 2.5 of PICS Proforma)
&	logical symbol "AND" (see section 2.5 of PICS Proforma)
v	logical symbol "OR" (see section 2.5 of PICS Proforma)
a.<i>	indication for additional information (see section 2.2 of PICS Proforma)
I	notation for "irrelevant" (see section 2.5 of PICS Proforma)
M	notation for "mandatory" (see section 2.5 of PICS Proforma)
N/A	notation for "not applicable" (see section 2.5 of PICS Proforma)
O	notation for "optional" (see section 2.5 of PICS Proforma)
O.<n>	notation for "qualified optional" (see section 2.5 of PICS Proforma)
X	notation for "eXcluded" (see section 2.5 of PICS Proforma)
x.<i>	indication for exceptional information (see section 2.2 of PICS Proforma)

3. MPOA Description

[Informative]

MPOA is designed with a client/server architecture. MPOA Clients (MPC) and their MPOA Server(s) (MPS) are connected via LANE.

3.1 MPOA Components

There are two types of MPOA logical components: MPC and MPS.

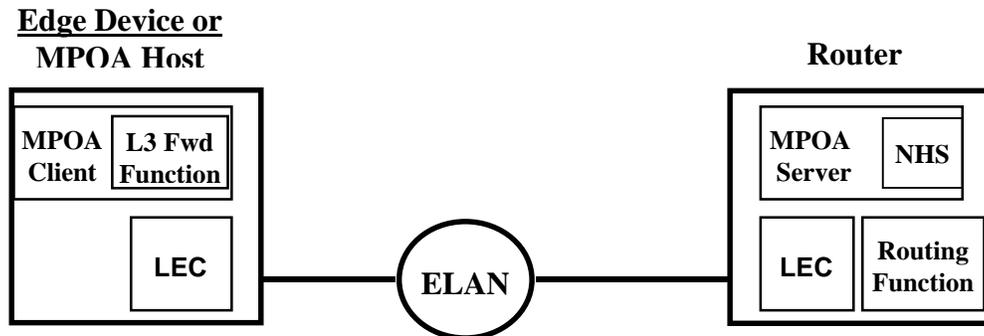


Figure 1 The Components in an MPOA System.

3.1.1 MPOA Client (MPC)

The primary function of the MPC is to source and sink internetwork shortcuts. To provide this function, the MPC performs internetwork layer forwarding, but does not run internetwork layer routing protocols.

In its ingress role, an MPC detects flows of packets that are being forwarded over an ELAN to a router that contains an MPS. When it recognizes a flow that could benefit from a shortcut that bypasses the routed path, it uses an NHRP-based query-response protocol to request the information required to establish a shortcut to the destination. If a shortcut is available, the MPC caches the information in its ingress cache, sets up a shortcut VCC, and forwards frames for the destination over the shortcut.

In its egress role the MPC receives internetwork data frames from other MPCs to be forwarded to its local interfaces/users. For frames received over a shortcut, the MPC adds the appropriate DLL encapsulation and forwards them to the higher layers (e.g., a bridge port or an internal host stack). The DLL encapsulation information is provided to the MPC by an egress MPS and stored in the MPC's egress cache.

An MPC can service one or more LECs and communicates with one or more MPSs.

3.1.2 MPOA Server (MPS)

An MPS is the logical component of a router that provides internetwork layer forwarding information to MPCs. It includes a full NHS as defined in [NHRP] with extensions as defined in this document. The MPS interacts with its local NHS and routing functions to answer MPOA queries from ingress MPCs and provide DLL encapsulation information to egress MPCs.

An MPS converts between MPOA requests and replies, and NHRP requests and replies on behalf of MPCs.

3.1.3 Examples of MPOA Enabled Devices

- MPOA Edge Device (including the MPC, the LEC and a bridge port.)
- MPOA Host (including the MPC, the LEC and an internal host stack.)
- Router (including the MPS, which in turn includes an NHS; the LEC and the routing function)

There are other possibilities to create MPOA enabled devices, e.g. co-locating the MPS and MPC in the router and thereby creating a device that is capable of internetwork routing/forwarding and detecting flows and creating ATM shortcuts for these flows.

3.1.4 Relationship Between LECs, MPOA Components, and MPOA Devices

As shown in Figure 2, there may be one or more MPCs in an edge device and one or more LECs associated with an MPC; however, a given LEC may be associated with one and only one MPC.

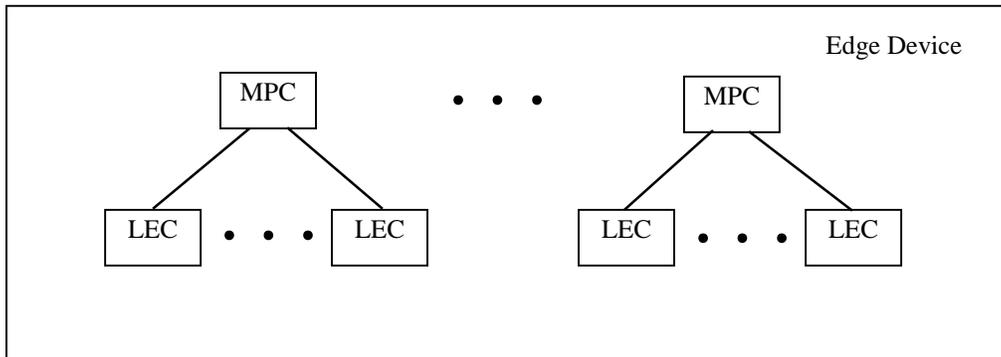


Figure 2 Relationship Between LECs, MPCs, and Edge Devices

Similarly, as shown in Figure 3, there may be one or more MPSs in a router and one or more LECs associated with an MPS; however, a given LEC may be associated with one and only one MPS.

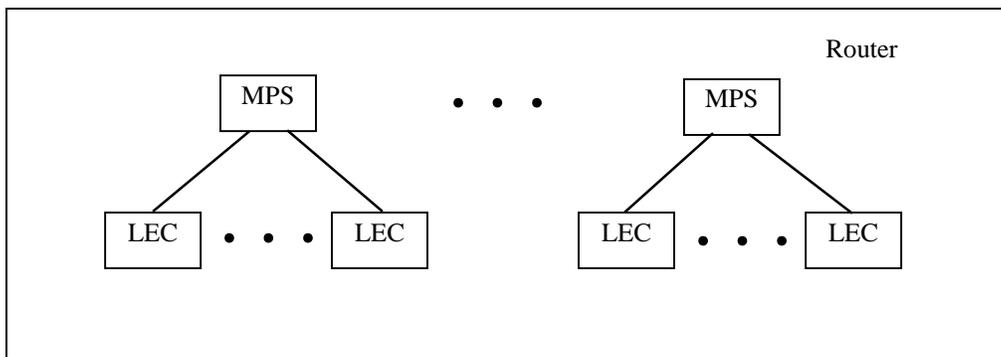


Figure 3 Relationship Between LECs, MPSs, and Routers

3.2 Control and Data Flows

The MPOA solution involves a number of information flows, shown in Figure 4, that can be categorized as MPOA control flows and MPOA data flows.

By default, all control and data flows are carried over ATM VCCs using LLC/SNAP [RFC 1483] encapsulation. Configuration flows use the formats described in [LANE]. More detailed discussion of the information flows is contained in the area descriptions in Section 3.3.

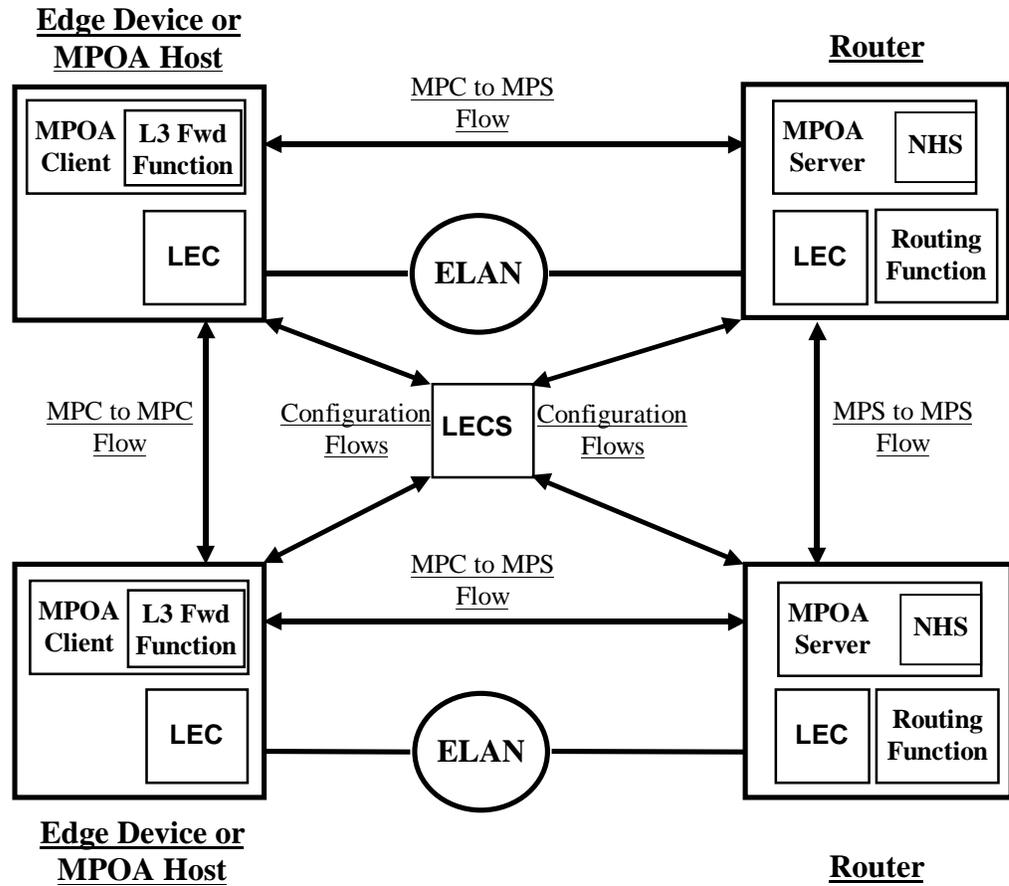


Figure 4 Information Flows in an MPOA System

3.2.1 MPOA Control Flows

3.2.1.1 Configuration Flows

By default, MPSs and MPCs communicate with the LAN Emulation Configuration Server (LECS) to retrieve configuration information.

3.2.1.2 MPC-MPS Control Flows

MPC-MPS control flows are used for MPC cache management. The MPOA Resolution Request/Reply allows the ingress MPC to obtain shortcut information. The ingress MPS may trigger the ingress MPC to make a request by sending the MPOA Trigger Message. The MPOA Cache Imposition Request/Reply allows the egress MPS to give the egress MPC egress cache information. Finally, either the egress MPC or an MPS may send a Purge message if it discovers that cached information has become invalid.

3.2.1.3 MPS-MPS Control Flows

MPS-MPS control flows are handled by standard internetwork layer routing protocols and NHRP. MPOA does not define any new MPS-MPS protocols. MPOA requires no new replication techniques and relies upon the standard replication techniques provided by LANE and internetwork layer routing protocols.

3.2.1.4 MPC-MPC Control Flows

An egress MPC may send a data plane purge to an ingress MPC if it receives misdirected packets from that MPC. This message causes the ingress MPC to invalidate its erroneous cache information.

3.2.2 MPOA Data Flows

3.2.2.1 MPC-MPC Data Flow

MPC-MPC flows are used primarily for the transfer of data between MPCs over MPOA shortcut VCCs.

3.2.2.2 MPC-NHC Data Flows

An MPC may send unicast data to an NHC and an NHC may send unicast data to an MPC.

3.3 MPOA Operations

MPOA performs the following operations:

Configuration	Obtaining the appropriate configuration information.
Discovery	MPCs and MPSs learning of each others' existence.
Target Resolution	Determining the mapping of a Target to an egress ATM address, an optional Tag, and a set of parameters used to set up a Shortcut VCC to forward packets across subnet boundaries.
Connection Management	Creating, maintaining, and terminating VCCs for the purpose of transferring control information and data.
Data Transfer	Forwarding internetwork layer data across a Shortcut.

3.3.1 Configuration

MPCs and MPSs each require configuration. By default, MPOA components retrieve their configuration parameters from the LECS. MPOA components must be capable of configuration via the LECS, although they may be administered to obtain their configuration by some other means. Other methods for obtaining configuration may include manipulation of the MPOA MIB, or through unspecified mechanisms.

3.3.2 Discovery

To reduce operational complexity, MPOA components automatically discover each other using extensions to the LANE LE_ARP protocol that carry the MPOA device type (MPC or MPS) and ATM address. This information is discovered dynamically and used as needed. This information may change and must be periodically verified.

MPCs are not NHCs and do not register host internetwork layer addresses with NHSs using NHRP Registration.

3.3.3 Target Resolution

MPOA target resolution uses an extended NHRP Resolution Request protocol to allow MPCs to determine the ATM address for the end points of a shortcut. In the following subsections, the protocol is described from the perspectives of the ingress MPC, the ingress MPS, the egress MPS, and the egress MPC.

3.3.3.1 Ingress MPC Perspective

An ingress MPC learns the MAC addresses of MPSs attached to its ELANs from the device type TLVs in LE_ARP responses. The MPC is required to perform flow detection, based on internetwork layer destination address, on packets destined for these learned MAC addresses. Additionally, an MPC is permitted to perform other types of flow detection. An example of this is if the MPC is co-located with an MPOA host, it may "detect flows" based on higher-layer information readily available from the host. In addition, the MPC should issue a request to an MPS from which it has received an MPOA Trigger (described in Section 4.7.2).

Default forwarding for the MPOA System is via routers. When an MPC becomes aware of a particular traffic flow that might benefit from a shortcut, the ingress MPC needs to determine the ATM address associated with the egress device. Note that the terms ingress and egress apply even if both MPCs are part of MPOA hosts. To obtain the ATM address for a shortcut, the ingress MPC sends an MPOA Resolution Request to the appropriate ingress MPS. When this MPS is able to resolve the MPOA Resolution Request, a reply is returned to the ingress MPC that contains an ATM address of the egress device.

The reply may contain information in addition to the requested ATM address. An example of information that may be included is encapsulation/tagging to be used for data sent on this shortcut. Note that NHRP is specified in such a way that only that information requested by the Resolution Request initiator may be included in the reply.

3.3.3.2 Ingress MPS Perspective

The ingress MPS processes MPOA Resolution Requests sent by local MPCs. The ingress MPS can answer the request if the destination is local, otherwise it re-originates the request along the routed path through its local NHS. The ingress MPS uses its internetwork layer address as the source protocol address in the re-originated request. This ensure that the reply is returned to the originating MPS. The MPS copies all other fields from the MPOA Resolution request. In particular, the MPC's data ATM address is used as the source NBMA address and all TLVs are copied. The MPS generates a new Request ID for the re-originated request. The MPS must set the S bit in the re-originated request to zero so that downstream NHSs do not cache the association of the resulting internetwork layer and ATM addresses.

On receiving a reply to this re-originated request, the ingress MPS restores the Request ID field and source protocol address to the original values and returns an MPOA Resolution Reply to the ingress MPC.

3.3.3.3 Egress MPS Perspective

When an NHRP Resolution Request targeted for a local MPC arrives at the egress MPS serving that MPC (the MPS, in this case, is the NHRP "authoritative responder"), the egress MPS sources an MPOA Cache Imposition Request.

The MPOA Cache Imposition Request is generated by the egress MPS and sent to the egress MPC. It is part of a cache management protocol that serves multiple purposes; the MPOA Cache Imposition Request provides encapsulation and state maintenance information needed by the egress MPC, while the MPOA Cache Imposition Reply provides status, address and ingress tagging information needed by the egress MPS to formulate the NHRP Resolution Reply.

After receiving the MPOA Cache Imposition Reply from the egress MPC, the egress MPS sends an NHRP Resolution Reply toward the request originator. Additional information requested by the ingress MPC (and included in the MPOA Cache Imposition Request and MPOA Cache Imposition Reply messages) must be included in the NHRP Resolution Reply as well.

3.3.3.4 Egress MPC Perspective

The egress MPC must send an MPOA Cache Imposition Reply for every MPOA Cache Imposition Request. To formulate its reply, the MPC must determine if it has the resources necessary to maintain the cache entry and potentially receive a new VCC. If the MPOA Cache Imposition Request is an update of an existing egress cache entry, the resources are likely available. If the MPC cannot accept either the cache entry or the VCC that will likely result from a positive reply, it sets the appropriate error status and returns the MPOA Cache Imposition Reply to the MPS. If it can accept this cache entry, the MPC inserts an ATM address and, if present, may modify the MPOA Egress Cache Tag Extension to be used by the ingress MPC in connection with this shortcut, sets a success status, and sends the MPOA Cache Imposition Reply to the egress MPS.

In some configurations, it is possible for an egress MPC to receive conflicting next hop forwarding instructions for the same source ATM address and destination internetwork layer address pair (as described in Appendix IV). If this conflict occurs, the MPC must take one of the following actions so that packets are forwarded properly:

1. If there is an MPOA Egress Cache Tag Extension present, the egress MPC may include an appropriate tag (unique to the source-destination ATM address pair and internetwork layer destination address) in the MPOA Cache Imposition Reply.
2. Return a distinct destination ATM address in the MPOA Cache Imposition Reply (thus forcing the requesting MPC to open a new VCC).
3. Refuse the cache imposition - indicating in an MPOA Cache Imposition Reply either that additional shortcuts are not possible or that a shortcut for this particular flow is refused.

While the primary technical reason for including a tag is to solve the egress cache conflict referenced above, it is important to note that tags can also be used to optimize the egress cache lookup. This optimization can be achieved by providing an index into the egress cache as the tag. When the tag is an index into the cache, the cache search is reduced to a direct cache lookup.

3.3.4 Connection Management

MPOA components establish VCCs between each other as necessary to transfer data and control messages over an ATM network. For the purpose of establishing control VCCs, MPOA components learn of each others existence by the discovery process described in Section 3.3.2. For the purpose of establishing data VCCs, MPOA components learn of each others existence by the resolution process described in Section 3.3.3.

3.3.5 Data Transfer

The primary goal of MPOA is the efficient transfer of unicast data. Unicast data flow through the MPOA System has two primary modes of operation: the default flow and the shortcut flow. The default flow follows the routed path over the ATM network. In the default case, the MPOA edge device acts as a layer 2 bridge. Shortcuts are established by using the MPOA target resolution and cache management mechanisms.

When an MPC has an Internetwork protocol packet to send for which it has a shortcut, the MPOA edge device acts as an internetwork level forwarder and sends the packet over the shortcut.

3.4 Routing Protocol Interaction

Routing information is supplied to MPOA via the NHS and its associated routing function. MPSs interact with NHSs to initiate and answer resolution requests. Ingress and egress NHSs (associated with MPSs) must maintain state on NHRP Resolution Requests that they have initiated or answered so that they can update forwarding appropriately if routing information changes. MPSs receive these updates from their co-located router/NHS and update or purge relevant MPC caches as appropriate. Interactions between an NHS and internetwork layer routing protocols are beyond the scope of this document.

3.5 NHRP/ION Interaction

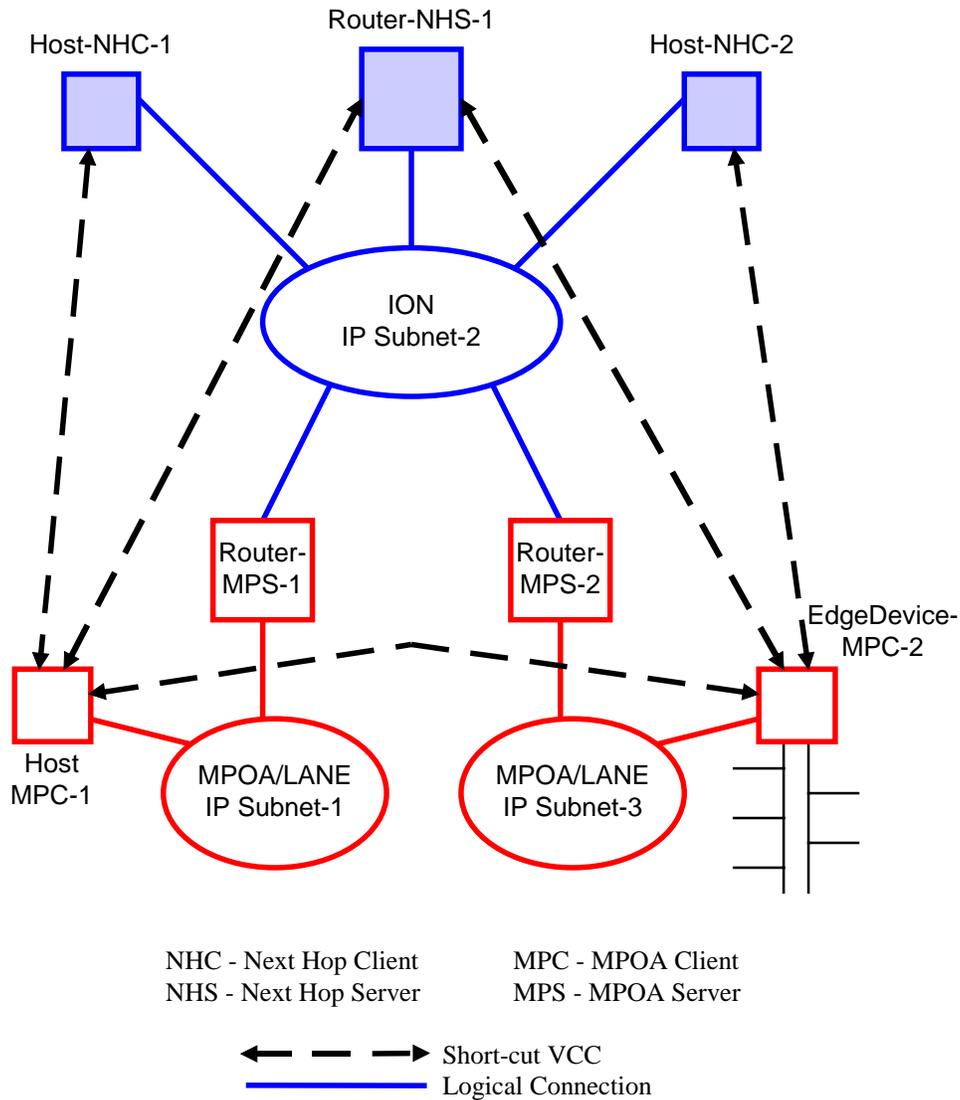


Figure 5 MPOA / ION Interaction

MPOA supports interoperation between MPOA devices and NHRP-only devices. To the NHRP devices in the ION domain, MPCs appear to be standard NHCs. As shown in Figure 5, unicast shortcuts can be established between MPOA devices and NHRP-only devices in different IP subnets. In particular, unicast shortcuts can be established between MPOA hosts and NHRP-capable hosts and routers, and unicast shortcuts can be established between MPOA edge devices and NHRP-capable hosts and routers. The one restriction related to MPOA/NHRP interoperability is that MPOA devices and NHRP (and Classical IP [IPOA]) devices must be on different subnets because intra-subnet unicast and multicast between MPOA and ION devices are not specified in this document.

3.6 A Day in the Life of a Data Packet

A packet enters the MPOA System at the ingress MPC (MPC 1). The decision process that takes place relative to each inbound packet at an MPC is outlined in Figure 8. By default, the packet is bridged via LANE to a router. If the packet follows the default path, it leaves the MPOA System via the ingress MPC's internal LEC Service Interface. However, if this packet is part of a flow for which a shortcut has been established, the ingress MPC strips the DLL encapsulation from the packet and sends it via the shortcut. The MPC may be required to prefix the packet

with tagging information (provided to the MPC via target resolution process - Section 3.3.3) prior to sending it via the shortcut.

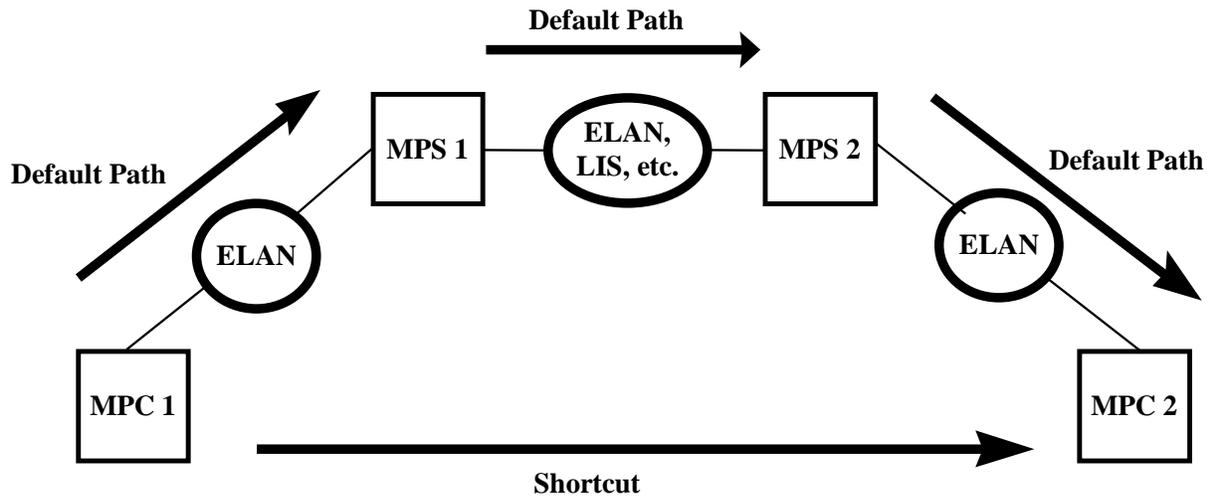


Figure 6 Example of a Day in the Life of a Packet

If no flow has been detected previously, each packet being sent to an MPS is tallied by internetwork layer destination address as it is being sent via LANE. When a threshold (given as a number of packets for a single internetwork layer address in a fixed period of time) is exceeded, the MPC is required to send an MPOA resolution request to obtain the ATM address to be used for establishing a shortcut to a specific downstream element - most likely an egress MPC (e.g. MPC 2).

On arriving via shortcut at the egress MPC, a packet is examined and either a matching egress cache entry is found or the packet is dropped. If a match is found, the packet is encapsulated using the information in the egress cache, and it is forwarded to the higher layer.

Appendix II provides example scenarios for the data and control flows.

4. MPOA Specification

[Normative]

MPOA uses a protocol based on the Next Hop Resolution Protocol [NHRP] to manage caches and establish shortcuts. This section describes the MPOA protocol including all configuration parameters, initial and operating states, and packet processing procedures. Section 4.1 describes all MPS and MPC configuration parameters and procedures. Section 4.2 describes the procedure by which MPOA components automatically discover each other. Section 4.3 describes the generic retry mechanism that MPOA components must use when retrying requests. Section 4.4 describes detailed MPC Behavior, and section 4.5 describes detailed MPS Behavior. Section 4.6 describes a Keep-Alive protocol by which MPCs detect the death of MPSs to ensure cache consistency. Section 4.7 describes cache maintenance. Finally, Section 4.8 describes connection management.

4.1 Configuration Parameters

Sections 4.1.1 - 4.1.2.2 describe the MPOA configuration parameters and constants. The granularity for the parameters and constants described is the same as is given in the tables.

4.1.1 MPS Configuration

Most MPS configuration information (such as what ELANs to operate over) can be derived from the underlying router configuration. Some additional configuration information is specific to the MPS.

4.1.1.1 MPS Parameters

The following parameters apply to each MPS:

Variable	Name	Description and Values
MPS-p1	Keep-Alive Time	The MPS must transmit MPOA Keep-Alives every MPS-p1 seconds. Minimum=1 second, Default=10 seconds, Maximum=300 seconds.
MPS-p2	Keep-Alive Lifetime	The length of time an MPC may consider a Keep-Alive valid in seconds. Minimum=3 seconds, Default=35 seconds , Maximum=1000 seconds (MPS-p2 must be at least three times MPS-p1)
MPS-p3	Internetnetwork-layer Protocols	The set of protocols for which MPOA resolution is supported. Default ={ }.
MPS-p4	MPS Initial Retry Time	Initial retry time used by the MPOA retry mechanism. Minimum=1 second, Default=5 seconds, Maximum=300 seconds
MPS-p5	MPS Retry Time Maximum	Maximum retry time used by the MPOA retry mechanism. Minimum=10 seconds, Default=40 seconds, Maximum=300 seconds
MPS-p6	MPS Give Up Time	Minimum time to wait before giving up on a pending resolution request. Minimum = 5 seconds, Default = 40 seconds, Maximum = 300 seconds
MPS-p7	Default Holding Time	The default Holding Time used in NHRP Resolution Replies. An egress MPS may use local information to determine a more appropriate Holding Time. Minimum=1 Minute, Default=20 Minutes, Maximum=120 Minutes

4.1.1.2 MPS Constants

The following constants are used by MPSs.

Constant	Name	Description and Values
MPS-c1	Retry Time Multiplier	Value: 2

4.1.2 MPC Configuration

4.1.2.1 MPC Parameters

The following parameters apply to each MPC:

Variable	Name	Description and Values
MPC-p1	Shortcut-Setup Frame Count	See parameter MPC-p2. Minimum=1, Default=10, Maximum=65535.
MPC-p2	Shortcut-Setup Frame Time	If an MPC forwards at least MPC-p1 frames to the same target within any period MPC-p2 via the default forwarding path, it should initiate the procedure to establish a shortcut.. The MPC-p1 and MPC-p2 parameters specify a default mechanism for automatically detecting flows in the absence of other information. Other mechanisms (e.g. RSVP) may be used in specific cases to override this default. Minimum=1 second, Default=1 second, Maximum=60 seconds.
MPC-p3	Flow-detection Protocols	A set of protocols on which to perform flow detection. Default ={}.
MPC-p4	MPC Initial Retry Time	Initial retry time used by the MPOA retry mechanism. Minimum=1 second, Default=5 seconds, Maximum=300 seconds
MPC-p5	MPC Retry Time Maximum	Maximum retry time used by the MPOA retry mechanism. Minimum=10 seconds, Default=40 seconds, Maximum=300 seconds
MPC-p6	Hold Down Time	Minimum time to wait before reinitiating a failed resolution attempt. This is usually set to a value greater than MPC-p5. Minimum=30 seconds, Default=MPC-p5*4, Maximum=1200 seconds

4.1.2.2 MPC Constants

The following constants are used by MPCs.

Constant	Name	Description and Values
MPC-c1	Retry Time Multiplier	Value: 2
MPC-c2	Initial Keep-Alive Lifetime	Keep-Alive Lifetime to use before the first Keep-Alive message is received. Value: 60 seconds.

4.2 Device Discovery

Discovery of control addresses of MPOA components is an essential part of the MPOA system. It is necessary for the MPOA Client to know the MAC and ATM addresses of local MPOA Servers so that MPOA Requests may be sent. The MPOA Server must know if an NHRP Request resolves to the ATM address of an MPOA Client so that a

cache imposition may be sent. Finally, MPSs sharing an ELAN must be able to discover each other to facilitate the forwarding of NHRP messages. To this end, an MPOA Device Type TLV, defined in Section 5.2.3, is included in the following LANE messages every time they are sent:

- LE_REGISTER_REQUEST
- LE_REGISTER_RESPONSE
- LE_ARP_REQUEST
- LE_ARP_RESPONSE
- Targetless LE_ARP_REQUEST (see 7.1.30 [LANE])

The information carried in the MPOA Device Type TLV includes the type of device (MPS, MPC, MPS/MPC, or non-MPOA), MPS MAC addresses (if the type is MPS), and the respective control addresses. Each MPOA component must register the MPOA Device Type TLV with each LEC on which it is configured.

4.2.1 Register Protocol

If a LEC is being served by an MPOA Server or Client, it must include the MPOA Device Type TLV in the LE REGISTER REQUEST for the relevant MAC addresses. The LEC indicates the type of device in the TLV.

If the LES has no existing entry for the MAC-ATM binding, it must register the new MAC-ATM binding with MPOA Identification information in its address table. However, if the LES has an existing entry for the specified MAC-ATM binding, it must overwrite any existing MPOA Identification with the new MPOA Identification.

If the status of an MPOA device changes, it should request each of its served LECs to send an LE UNREGISTER REQUEST to the LES for each registered MAC address. After unregistering, the LEC should send an LE REGISTER REQUEST with the new set of TLVs to the LES for each MAC address.

4.2.2 Address Resolution Protocol

Address resolution requests and replies sent by a LEC with an ATM address that is associated with an MPOA device must include the MPOA Device Type TLV (as described in Section 5.2.3). The TLV will indicate whether the sending MPOA device is an MPOA Server, an MPOA client, or both. A LEC receiving an address resolution request or response must update its MAC-ATM binding entry to reflect the MPOA Identification TLV.

If the status of an MPOA device changes, it should request each of its served LECs to send a targetless LE_ARP_REQUEST to the LES for MAC addresses previously included in an address resolution response with the new set of TLVs.

4.2.3 Implications for Co-Located MPS, MPC and Non-MPOA Devices

A device may have one or more MAC addresses on a LANE LEC which are associated with an MPS or MPC in the device, and one or more MAC addresses which are not. For example, a device may be both a router and a bridge. It might have a router MAC address associated with an MPS, but still respond to other MAC addresses which it is bridging. Those bridged MAC addresses would not be associated with the MPS. As another example, a bridge with an MPC might or might not want to associate the MPC with the MAC address it uses for SNMP traffic to the bridge, itself.

Any MPOA device which has a LEC that has a set of MAC addresses associated with an MPC, and a set of MAC addresses not associated with any MPOA capability must have separate LEC ATM addresses (or sets of ATM addresses) associated with those two sets of MAC addresses. No LEC ATM address given out with a non-MPOA MAC address can also be given out for an MPC MAC address. This means that traffic for MPC MAC addresses and non-MPOA MAC addresses cannot share the same VCC, but must be carried on separate VCCs. This is to allow other MPOA devices which are performing learning of MAC to VCC/ATM address bindings from LANE data frames, to determine the correct MPOA capabilities of the MAC addresses learned in this manner.

A router with a LEC that has a set of MAC addresses associated with an MPS, is not required to use a separate LEC ATM address for the MPS MAC addresses. Instead it may share the LEC ATM address with MPC or non-MPOA MAC addresses. It may also choose to use a separate LEC ATM address and not share.

The ability to share is achieved by including the list of MPS MAC addresses in a Device Type TLV in LANE control messages sent pertaining to the LEC ATM address serving the set of MPS MAC addresses. For Requests this means that the SOURCE-ATM -ADDRESS field contains the LEC ATM address, for Responses this means that the TARGET-ATM-ADDRESS field contains the LEC ATM address.

Note that if sharing is used, the list of MPS MAC addresses on the given ELAN will be included in LANE control messages (as defined in Section 4.2) issued for or on behalf of an MPC or non-MPOA MAC address served by that shared LEC ATM address.

By including an explicit list of MPS MAC addresses in the Device Type TLV, other MPOA devices which are performing learning of MAC to VCC/ATM address bindings from LANE data frames can determine the correct MPOA capabilities of the MAC addresses learned in this manner, even if a separate LEC ATM address is not used for MPS MAC addresses. All MAC addresses learned that are not in the MPS list, are associated with either an MPC or as not being MPOA capable, as determined by the capability associated with the VCC.

A device that is performing learning of MAC to VCC/ATM address bindings from LANE data frames is required to perform at least one LE-ARP on at least one MAC address for every ATM address to which it has a Data Direct VCC to discover the MPOA capabilities of the device at the other end of the VCC. It does this to associate the correct MPOA capabilities with the MAC addresses it learns in this way. This rule means that, if a LEC accepts a Data Direct VCC from another device, and it has no LE_ARP cache entry for that ATM address, and it then receives data frames from that device, it must not simply learn the source MAC addresses from the frames and populate its LE_ARP cache with that learned information. It may populate its LE_ARP cache in this manner, but it must, in addition, issue an LE_ARP for one of those learned MAC addresses, and receive the reply, to find out whether the associated ATM address is or is not associated with an MPC or MPS.

Note that these arguments do not prevent an MPS and MPC in the same device from sharing a common ATM address for their non-LANE control connections. The MPOA packet formats are chosen so that messages for those two entities cannot be confused. Furthermore, an MPS or MPC may share an ATM address with one used by one or more of its LECs for MPS/MPC associated MAC addresses.

4.2.4 Change of Device Status

Note that, in order for the MPOA-aware devices to be able to trust just one LE_ARP request or response, a device must be careful when changing its configuration to start or stop an MPS or MPC, start or stop an associated LEC, or associate a LEC with, or disassociate a LEC from, an MPS or MPC. There are a number of ways to notify other MPOA-aware devices of a change in the configuration of an MPS, MPC, or associated LEC:

1. A LEC or MPC or MPS may terminate ELAN membership and/or cease operations for a significantly longer period than the maximum LE_ARP time-out period (5 minutes in [LANE]). This ensures that information about the device will age out.
2. A LEC may send one or more Targetless LE_ARP_REQUESTs to update other LECs' MPS/MPC associations. Such a Targetless LE_ARP_REQUEST may be sent more than once for reliability, and may be sent each time an indication (such as an MPOA request sent to a device in which the MPC or MPS is no longer active) is received that some other device is confused.
3. An MPC (MPS) may respond to an MPOA request with error code 0x86 (0x87), meaning, "this device is no longer an MPC (MPS)."

Of all these methods, only the first is a reliable means for a device to notify all other devices of a change in its configuration.

4.3 MPOA Retry Mechanism

MPOA Requests may be retried if a response is not received within a reasonable amount of time. The following retry mechanism is defined for MPOA components when retrying request messages. While retrying a failed request, the MPOA component must use the same Request Id that was used in the initial request (if appropriate). Retries for failed requests must use the retry procedure as follows: The retry procedure includes an initial time-out of MPS-p4/MPC-p4 seconds, a retry multiplier of MPS-c1/MPC-c1, and a cumulative maximum time-out of MPS-p5/MPC-p5. When an MPOA component sends a request, it sets a RetryTimer to the value of MPS-p4/MPC-p4 seconds. If a

corresponding reply is not received before RetryTimer seconds have elapsed, the MPOA component may send a new request (retry). Each time a retry is sent, the RetryTimer is set to $\text{RetryTimer} * \text{MPS-c1} / \text{MPC-c1}$. If the value of RetryTimer exceeds the Retry Time Maximum ($\text{MPS-p5} / \text{MPC-p5}$), the request is considered to have failed.

4.4 Detailed MPC Behavior

The MPC lies between a LANE LEC and its higher layers. Each LEC for which MPOA is enabled is associated with exactly one MPC. Each MPC serves a set of one or more LECs, and has a single MPC control ATM address. This address may be the same as an ATM address used by one or more of its LECs. If there are multiple MPCs within a device, each must serve a disjoint set of LECs, and must use different MPC control ATM addresses. For example, Figure 7 shows an MPOA edge device with a single MPC and two LECs.

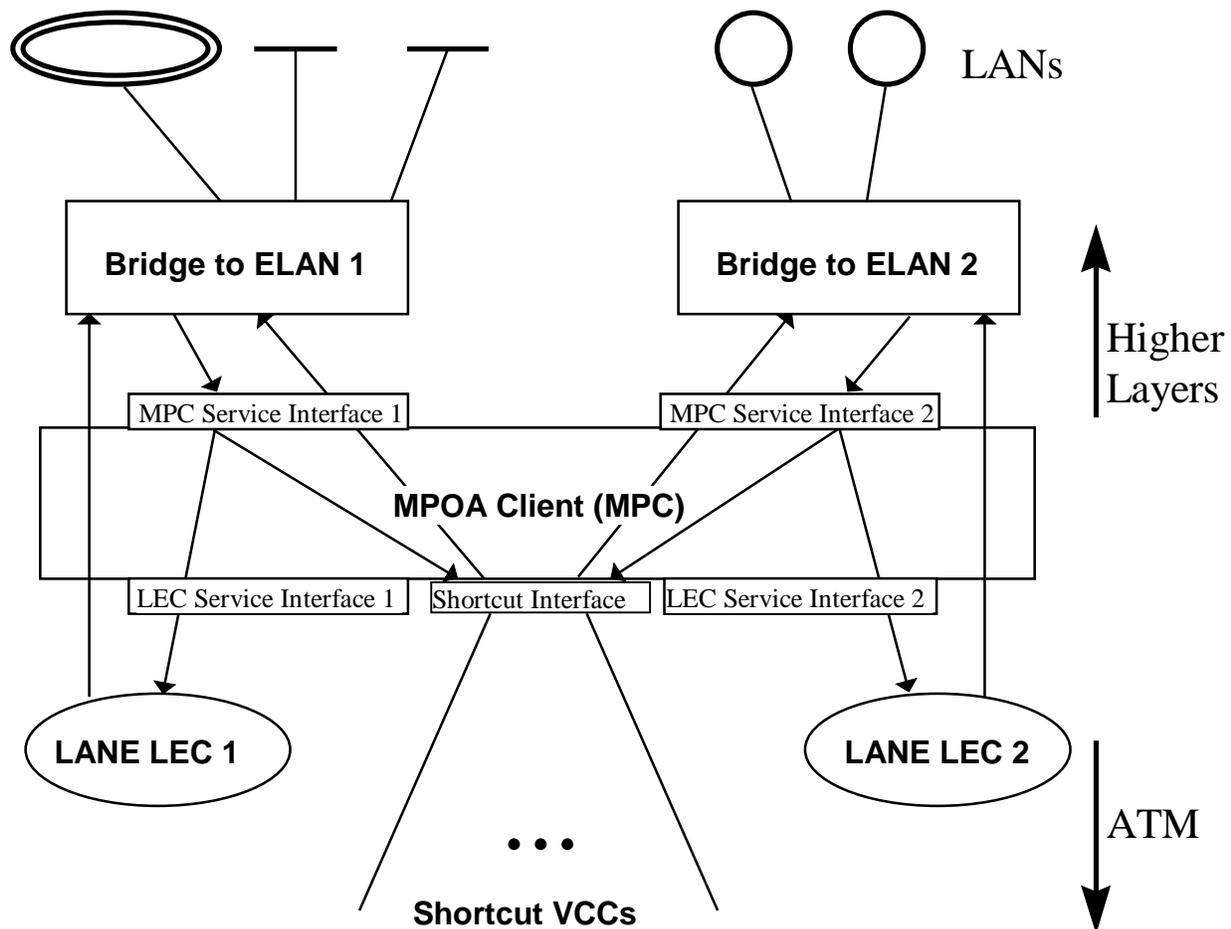


Figure 7 MPOA Edge Device MPC Example

For each LEC on which an MPC is to be enabled, the MPC supplies the LEC with the MPC's control ATM address via the MPOA Device Type TLV. Each LEC so notified includes the MPC device type TLV in every LE_ARP response which indicates to the recipient that an MPC is serving to the responding LEC, and indicates the control ATM address of the MPC.

The MPC presents the same Service Interfaces to its higher layers as its associated LECs present to it, except that the MPOA Device Type TLV may be stripped from the information provided to higher layers. An MPC analyzes packets from the MPC Service Interfaces for flow classification, collects statistics, and redirects appropriate packets to shortcuts. Non-redirected packets are passed on to the LEC Service Interface corresponding to the MPC Service Interface over which the packet was received. Packets received from a LEC Service Interface are passed transparently up to the corresponding MPC Service Interface. The difference between a LANE-capable bridge and

an MPOA edge device lies in the MPC. Both LANE and bridging are outside the scope of MPOA. The detailed behavioral diagram of the MPC is shown in Figure 8. Note that the MPC sees only:

1. Packets sent by the higher layers of the MPC Service Interface destined for a LEC (i.e. the inbound data flow);
2. Packets received on a Shortcut Service Interface and relayed to the higher layers as if they came from a LEC (i.e. the Outbound data flow);

Data packets received on a LEC Service Interface destined for the higher layers are passed without examination by the MPC.

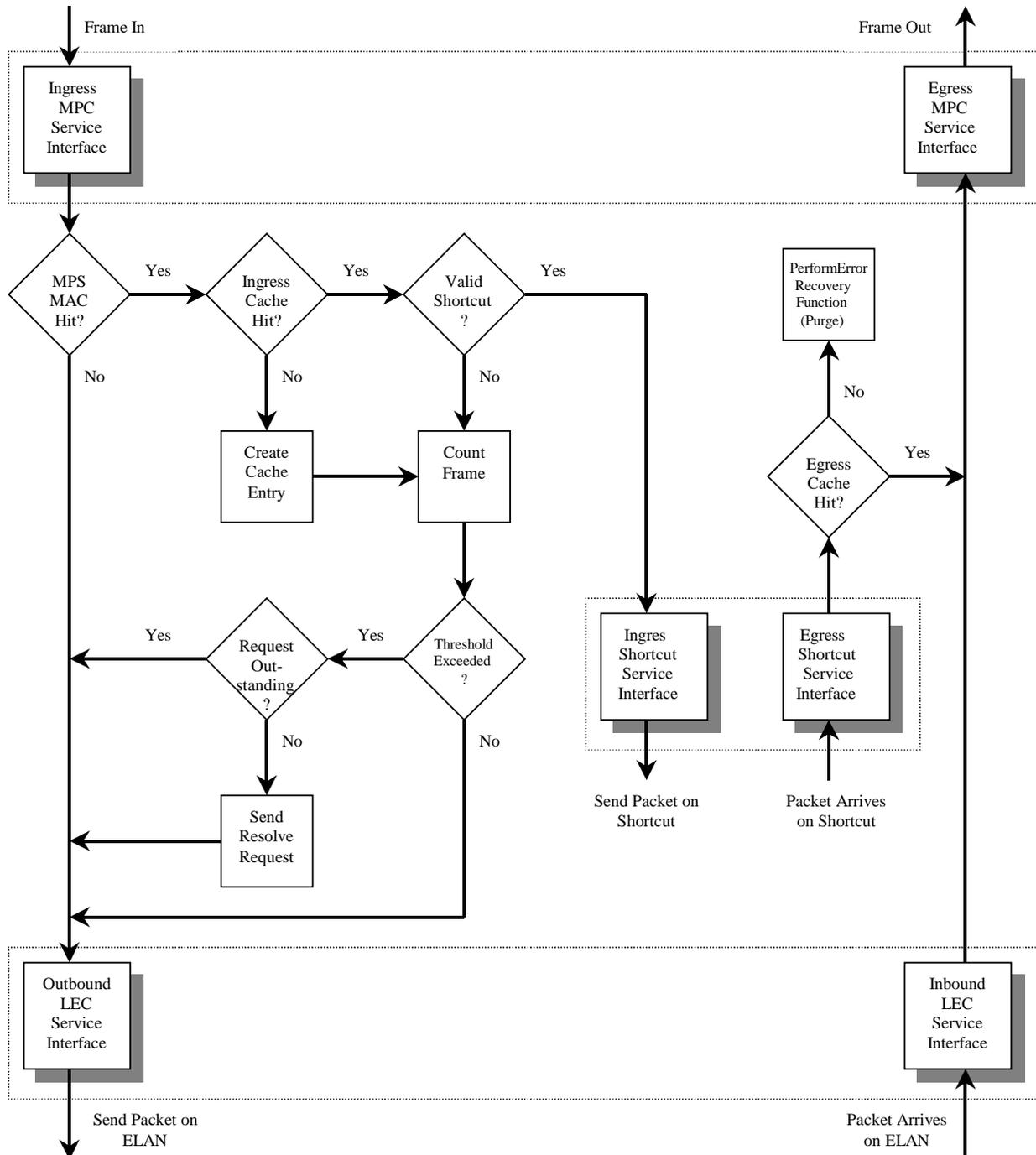


Figure 8 MPC Data Path Processing Logical Block Diagram

4.4.1 MPC Configuration

MPCs must be capable of obtaining configuration information from an LECS. To obtain this configuration information, the MPC sends a Configuration Request with a MPOA Device Type TLV identifying the device as an MPC to the LECS. The LECS uses the same mechanism when mapping Configuration Requests with MPOA Device Type TLVs to configuration information as it does for standard LEC requests. The LECS may use the MPOA Device TLV to return only MPC specific TLVs. MPC TLVs may be returned by the LECS in any Configuration Response. MPC TLVs sent by the LECS override the default values of the corresponding MPC parameters. If no MPC TLVs are returned, the MPC must use default or locally configured values.

The MPC may be configured to bypass the configuration phase and use default or locally configured parameters. An MPC should retrieve configuration over only one of its LECs. MPC TLVs may be returned to a MPC's LEC during the LEC's initialization phase. The MPC may use this information without issuing another Configuration Request.

The TLV encodings of MPC configuration parameters are specified in Section 5.2.2.

4.4.2 Inbound Data Flow

All inbound packets (packets sent by the higher layers toward a LEC) are examined to see whether they have the destination MAC address of an MPS. The MAC addresses of MPSs on ELANs served by an MPC, and their associated control ATM addresses, are known to the MPC through the discovery mechanism described in Section 4.2.

If detection is enabled for the protocol in the packet, the MPC examines the internetwork layer destination address of the packet, and looks up that <MPS Control ATM Address, Internetwork Layer Address> tuple in its ingress cache.

It is important to note that, in a given MPC, more than one <LEC, MPS MAC address> tuple may map to the same MPS Control ATM address because both the MPS and MPC may be on more than one ELAN, and the MPS may have a different MAC address on each ELAN. In this instance, LEC refers to a particular LEC co-located with the MPC. Flow detection and requests, therefore, are on a per <MPS Control ATM Address, Destination Internetwork Layer Address> pair basis. An implication of this design is that when an MPS receives an MPOA resolution request, it does not know which of the LEC interfaces the packets related to the request would have arrived on; therefore, different inbound filters per interface cannot be supported with a single MPS Control ATM Address. To support per interface inbound filters, a router must use a different MPS Control ATM Address (and therefore a different MPS) per unique set of inbound filters.

The lookup process described above may be modeled as a two stage process:

1. <LEC, MAC> \Rightarrow MPS Control ATM Address (from LE_ARP cache).
2. <MPS Control ATM Address, Destination Internetwork Layer Address> \Rightarrow Cache Entry.

The contents of the ingress cache are shown in Table 2.

Table 2 Ingress Cache

Keys		Contents		
MPS Control ATM Address	Internetwork Layer Destination Address	Dest. ATM Address or VCC	Encapsulation Information	Other information needed for control (e.g. Flow Count and Holding Time)

If the <MPS Control ATM Address, Internetwork Layer Address> tuple is not found in the ingress cache, a new ingress cache entry is created. The ATM address/VCC field is invalidated, and the "count" field is set to 1 to count the frame. The frame is then sent on to the LEC for output to the ELAN.

If the <MPS Control ATM Address, Internetwork Layer Address> tuple is found, but the ATM address/VCC field does not specify an operational VCC, then the packet is counted in the count field. The frame is then sent on to the LEC for output to the ELAN.

By the time the count for a given <MPS Control ATM Address, Internetwork Layer Address> tuple exceeds the configured threshold for number of packets (MPC-p1) sent within a configured time period (MPC-p2), then the MPC is responsible for sending an MPOA Resolution Request to the MPS to which the packet's MAC destination address is associated, requesting a shortcut. The MPOA Resolution Request, specified in Section 5.3.2.4.6, is very similar to the NHRP Resolution Request. The main difference is that the source protocol address in the MPOA Resolution Request may be NULL, since the MPC may not have an internetwork-layer address. The MPOA Resolution Request also must include an MPOA Egress Cache Tag extension, should include an MPOA ATM Service Category extension, and may include additional extensions as specified in [NHRP]. Note that during the period of time between when the threshold has been exceeded and a shortcut has been established, the MPC must limit the number of MPOA Resolution Requests it sends for the destination as described in Section 4.4.6.1.4.

If the <MPS Control ATM Address, Internetwork Layer Address> tuple is found in the ingress cache, and the ATM address/VCC field specifies an operational shortcut, then the packet's DLL encapsulation is stripped, the packet is encapsulated in the appropriate internetwork layer encapsulation (as defined in Section 5.1), and the packet is sent over the specified shortcut. The ingress MPC may receive padded frames. The ingress MPC may strip padding, or forward the frame with padding included.

4.4.3 Outbound Data Flow

Before an egress MPC can forward any packets, it must have received an egress cache entry from an egress MPS in an MPOA Cache Imposition Request. When an MPS determines that it must impose an egress cache entry on an MPC, as described in Section 4.5.2.2, the MPS sends an MPOA Cache Imposition Request to the MPC. The egress MPC must send an MPOA Cache Imposition Reply for every MPOA Cache Imposition request. To formulate its reply, the MPC must determine if it has the resources necessary to maintain the cache entry and potentially receive a new VCC. If the MPOA Cache Imposition is an update of an existing egress cache entry, the resources are likely available. If the MPC cannot accept either the cache entry or the likely resulting VCC, it sets the appropriate error status and returns the MPOA Cache Imposition Reply to the MPS. If it can accept this cache entry, it constructs an MPOA Cache Imposition Reply message. The MPOA Cache Imposition message includes an MPC data ATM address and a success status. The MPOA Egress Cache Tag Extension may also be included in the reply if it was included in the Cache Imposition Request. If an MPOA Service Category extension is present in a Cache Imposition Request, the egress MPC must set the Service Category extension in its Cache Imposition Reply to indicate the Service Categories supported by the egress MPC. If the MPOA Original Error Code TLV was in the request, the egress MPC must set it to the value as is set in the Cache Imposition Reply.

For all packets received on a shortcut, the egress MPC searches its egress cache for a matching entry. A cache hit is defined as a match on two main keys (source/destination ATM address pair, internetwork layer destination address) and/or one optional key (tag). In the non-tagged case, the source/destination ATM address pair is used as a key because it is possible for packets for a given internetwork layer destination to be forwarded to different next hops based on where they came from (see Appendix V).

The specific cache structure and lookup order is implementation dependent. The contents of the egress cache are shown in Table 3.

Table 3 Egress Cache Without Tags

Keys		Contents		
Internetwork Layer Destination Address	Source/Dest. ATM Addresses	LEC	DLL header	Other information needed for control (e.g. Holding Time)

An egress MPC may optionally use a tag for an egress cache entry. The tag may be used as the entire key, or as part of the key, to locate the relevant egress cache entry for a packet received on a shortcut VCC. The ingress MPC treats the tag as an opaque data string to be prepended to outgoing data frames.

If there are multiple egress cache entries that use the same source ATM address and internetwork layer destination address, but different DLL headers, then by using a different tag value for each one a unique key can be determined.

An alternative method is to use a different destination ATM address, which also results in a unique key. An egress MPC may choose to associate a distinct tag value with every egress cache entry. In this case the egress cache entry can be determined by using the tag field alone.

Because an explicit indication of the internetworking protocol is not carried in packets using the tagged encapsulation on shortcut VCCs, an egress MPC must be able to determine the protocol from the egress cache entry retrieved using the tag. The set of tag values used for different protocol address families, for a given source/destination ATM address pair, must therefore be distinct.

Table 4 Egress Cache With Tags

Keys			Contents		
Internetwork Layer Destination Address	Source/Dest. ATM Addresses	Tag	LEC	DLL header	Other information needed for control (e.g. Holding Time)

If an entry matching a packet received on a shortcut is not found in the egress cache, the packet is discarded, the error is counted, and the egress MPC initiates the Data Plane Purge process described in Section 4.7.2.3. If there is a cache hit, but the MAC destination address (from the DLL header) is not present in the LEC's C6, C8, C27, C30 variables, the egress MPC may continue to forward packets to the bridge as if they came from the LEC interface for up to 30 seconds to allow normal bridge flooding and learning procedures to occur. If the condition does not change within the 30 seconds, the egress MPC must invalidate the cache entry and send an MPOA Egress Cache Purge Request (See Section 4.7.1.6) to the MPS that imposed that egress cache entry.

If the VCC and internetwork layer destination address are in the cache, and the indicated LEC is fully operational, then the DLL header in the egress cache is attached to the internetwork layer packet, and the resultant frame is passed to the MPC service interface as if it arrived from the LEC. This DLL header comprises all octets preceding the start of the internetwork layer packet. The egress MPC may receive both padded and unpadded internetwork layer PDUs; therefore, the egress MPC may have to add padding to comply with the minimum frame size of the egress ELAN type.

The DLL header for the 802.3 format requires a length field indicating the length of all fields, starting with the LLC field, and including the IP packet. Therefore, the egress cache DLL header type is not transparent to the egress MPC. The egress MPC must parse the egress cache DLL header at least once, to determine that the length field is present, and must insert the correct value for the length of each outbound packet. Similarly, for inbound packets, the MPC must parse the frame to find the IP packet and check the validity of the length field, and then must perform the transformation to the shortcut format.

4.4.4 Cache Management

The ingress and egress caches are completely separate. Creation, deletion, or alteration of an entry in one cache does not imply any consequences for the other cache.

4.4.4.1 Ingress Cache Entry Creation and Management

An ingress MPC examines all packets destined for MAC addresses that belong to MPSs. When it detects a packet destined for an internetwork layer destination for which it does not already have a cache entry, it creates a new ingress cache entry for that internetwork layer destination. When the MPC detects a flow to a given internetwork layer destination, it sends an MPOA Resolution Request. When an MPOA Resolution Reply is received, the internetwork layer destination address, destination ATM address, source holding time, and MPOA Egress Cache Tag Extension (if present) are used to complete the ingress cache entry. If a reply is not received, the MPC should use the retry procedure defined in Section 4.3.

Any existing VCC may be used for data forwarding if its source and destination ATM addresses match those in the MPOA Resolution Reply, and the VCC signaling parameters are suitable. Otherwise, the ingress MPC must signal the creation of a new VCC before the shortcut can be used.

Ingress cache entries are aged using the source holding time from the latest MPOA Resolution Reply received for the corresponding internetwork layer destination address. Ingress cache entries may be withdrawn by the ingress MPS or deleted by the ingress MPC at any time for local reasons.

When an ingress MPC receives an NHRP Purge Request it must stop using the shortcut for packets destined to the specified internetwork layer destination address. It may issue a new MPOA Resolution Request immediately, or it may wait and use local information to determine when to query again.

To prevent active cache entries from aging out, ingress MPCs should issue new MPOA resolution requests to refresh these active cache entries at some time prior to the expiration of the holding time. For example, an MPC may choose to refresh active cache entries by sending a new resolution request after two thirds of the holding time has expired.

4.4.4.2 Egress Cache Entry Creation and Management

When an MPS determines that it must impose an egress cache entry on an MPC, the MPS sends an MPOA Cache Imposition Request to the MPC. The MPC uses the cache ID (in the MPOA DLL Header Extension) and the egress MPS control ATM address as the key to find and/or create an egress cache entry.

Egress cache entries are created with a holding time provided by the egress MPS. The entry must not be used beyond the egress holding time. If an egress cache imposition with a non-zero holding time is received for an existing cache entry, the egress MPC must update the egress cache entry. If an egress cache imposition with a zero holding time is received for an existing cache entry, the egress MPC must stop using the entry.

An egress MPC may find that it must discard packets received over a shortcut because the egress cache entry has become invalidated. The details of why a LEC no longer serves a LAN destination, or why an MPC views an egress cache entry as invalidated is a matter local to the MPC. An example would be an edge device that incorporates a bridge running the 802.1D spanning tree protocol, that finds that a packet received over a shortcut is due to be sent over a port that is not in the forwarding state, or is due to be sent back out the LEC port associated with the shortcut that the packet arrived on. This situation could occur due to a bridge topology change. Such a change might result in an edge device no longer being the correct edge device for a given internetwork layer destination address.

The definitions of the configuration and run-time variables controlling a LEC provide the means for specifying these conditions exactly. The LAN destinations served by a LEC are the union of the LAN destinations in the LEC's variables Local Unicast MAC Address(es) C6, Local Route Descriptor(s) C8, Remote Unicast MAC Address(es) C27, and Remote Route Descriptor(s) C30. Whenever a packet is received over a shortcut, the egress cache entry for that packet specifies the receiving LEC, and supplies the DLL encapsulation. If this destination MAC Address or route descriptor is not present in the LEC's C6, C8, C27, and C30 variables, then the egress cache entry is invalid and the MPC should initiate a purge.

Note that this follows normal LANE usage for answering LE_ARP_REQUESTs. No egress cache entry can be created if the destination MAC Address or destination Route Descriptor is not in the four LEC variables listed above, because the LEC would not answer an LE_ARP_REQUEST for that LAN destination. If the LEC would not answer an LE_ARP_REQUEST for the LAN destination, the router would not send that packet to that edge device via LANE, and hence the MPC has no business forwarding the packet.

If an egress MPC detects that an egress cache entry is invalid, it must inform the MPS that imposed the egress cache entry as described in Section 4.7.1.6. If the MPC has lost contact with the MPS, it should initiate a Data Plane Purge as described in Section 4.7.2.3.

4.4.5 LAN-to-LAN Flows Within the Same MPOA Device

An MPOA device must be able to handle the case where it is both the ingress MPC and the egress MPC for a given data flow. A simple implementation of this specification would cause the device to set up a VCC to itself. Although this simple implementation will work correctly, a more efficient implementation may build an internal shortcut and bypass the ATM network.

4.4.6 Control Information in MPC Caches

4.4.6.1 Ingress Cache

The control information required for maintaining ingress cache entries can be further divided into sub groups based on the function they are serving. The main control functions are State, Connection, Aging, Retry, Usage and Purge.

4.4.6.1.1 State Information

The following is a list of information that is needed to maintain the state of an ingress cache entry. The shortcut state information is separated from the VCC state information.

- Shortcut Entry State - Indicates if this entry is in Resolving/Resolved/Invalid states.
- Shortcut VCC State - Indicates if the shortcut VCC is Open/Closed.
- Ingress MPS Control ATM Address - This is the address of the ingress MPS to which the MPOA Resolution Request is sent.
- Last NHRP CIE Code – This is the last CIE code received in a MPOA Resolution Reply. Tracking the last error aids in debugging.
- Last Q.2931 Cause Value – This is the last cause value received in a Q.2931 Release. Tracking the last error aids in debugging.

4.4.6.1.2 Connection Information

The packet forwarding function generally uses the VPI/VCI information, but there are other pieces of information that need to be maintained.

- ATM Address of Egress-MPC - This is the address used as the Called Party Address while setting up a shortcut VCC.
- Service Category - If an ATM Service Category Extension was received in a MPOA Resolution Reply, it should be saved to be used while setting up shortcut VCCs.

4.4.6.1.3 Aging Information

The MPOA Resolution Response returns a Holding Time that is the time for which the information returned in the response is valid. This information is used to age the entry once it is Resolved.

This field can be implemented in different ways; for example, it could be counted down to 0 based on a periodic timer or it could be set to the time at which this entry should be removed.

4.4.6.1.4 MPOA Resolution Request Retry

MPOA Resolution Requests may be retried if a response is not received within a reasonable amount of time. These retries must use the MPOA retry mechanism described in section 4.3. In addition, if the request fails, a new resolution request for the same flow must not be issued for the duration of the Hold Down Time (MPC-p6).

The following information pertains to retries of MPOA Resolution Requests:

- Request Id - The Request Id that was used in an outstanding request, this must be reused for subsequent retries.
- RetryTime - This variable tracks the number of seconds that an MPC must wait before retrying a resolution request.

While attempting to resolve an address, an MPC may decide at any time that a shortcut is no longer needed and terminate the retry procedure.

4.4.6.1.5 Usage

There needs to be a count of the usage of a particular ingress cache entry for forwarding packets. This count is used primarily to do MPOA flow detection, but can also be useful for management and for maintaining lists for recycling cache entries when the system runs out of memory resources.

- Packets Forwarded - Number of packets that were forwarded using this cache entry.

4.4.6.2 Egress Cache

The control information contents of an egress cache entry can also be grouped based on the functions they serve. The different functions are State, Connection, Aging, Usage and Purge.

4.4.6.2.1 State Information

The state information consists of fields that are used to maintain the state of a shortcut. The state information of the entry is kept distinct from the actual VCC state.

- Shortcut Entry State - This field can be Resolved/Purge/Invalid. Packets received over shortcut VCCs are forwarded only when the entry is in the Resolved state
- Egress MPS Control ATM Address - This field is used to identify the MPS that imposed this cache entry. It is used for subsequent communication with the egress MPS for sending Egress Cache Purges.
- Cache Id - This information is used as a key in combination with the previous field to identify a unique egress cache entry for cache updates from the egress MPS.

4.4.6.2.2 Connection Information

The information related to shortcut VCCs are stored in these fields.

- Ingress MPC/NHC data ATM Address - The ATM address of the ingress MPC that issued the MPOA Resolution request for this entry. This address will be the Calling Party Address of an incoming shortcut VCC Setup Request. This may be used to verify the identity of the sender of packets over the shortcut VCC.

4.4.6.2.3 Aging Information

Every egress cache entry has a holding time associated with it that was provided in the MPOA Imposition Request message. This time is used to keep an entry in the Resolved State

4.4.6.2.4 Usage Information

It is useful for the management of the device to keep count of the number of packets that were received and processed using an egress cache entry. In addition this count would be useful to build a list of entries that need to be purged in case the system is running out of memory resources.

- Packets Received - This counter is incremented each time a packet is received on a shortcut VCC and this egress cache entry is used for encapsulating the packet before passing it to the outbound MPC Service Interface.

4.4.6.2.5 Purge Information

There are different reasons for purging entry/entries in an egress cache. The contents of the purge message vary depending on the purge reason.

- Egress MPS Control ATM Address - This information is needed if the MPC fails to receive an MPOA Keep-Alive message from the egress MPS within its life time. The egress MPC needs to inform the ingress MPC to purge all ingress cache entries that were imposed on the egress MPC by the failed egress MPS.

4.5 Detailed MPS Behavior

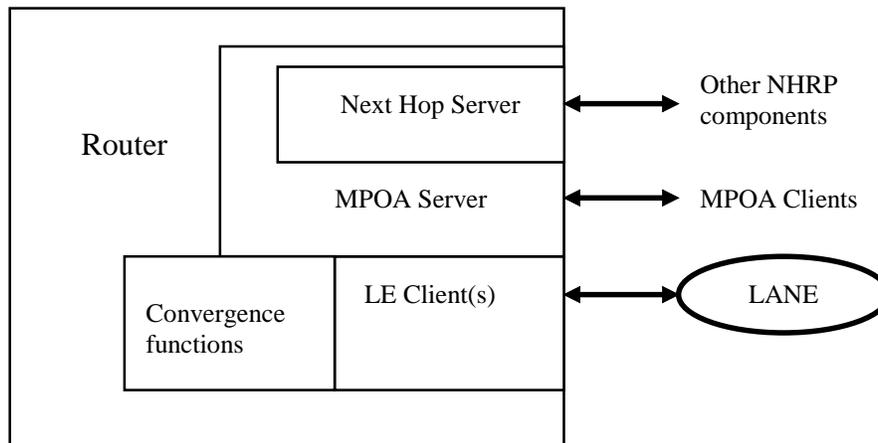


Figure 9 Router MPS Example

The MPS is a component of a router, as illustrated in Figure 9. An MPS is only useful in a router that has an NHS and interfaces to one or more LECs. The data and control path from the router through the LEC(s) to LANE is unaltered by MPOA. The MPS does, however, interact with the router, its LEC(s), the NHS, and other MPOA components. A LEC is associated with a single MPS.

The router engages in the operation of traditional routing protocols. One or more of the router's interfaces must use LANE.

The MPS must be aware of the router's configuration and forwarding tables to the extent of knowing (or being able to ask the router) whether a given internetwork layer destination address should be forwarded to a LEC, and which one. For each LEC and MAC Address on which the MPS is to be enabled, the MPS supplies the LEC with the MPS's control ATM address. Each LEC so notified includes the MPOA Device Type TLV in every LE_REGISTER_REQUEST, LE_ARP_REQUEST, and LE_ARP_RESPONSE that it sends to indicate to the recipient that an MPS is connected to the responding LEC and MAC Address, and to indicate the control ATM address of the MPS.

Having advertised its control ATM address via LE_ARP LANE control messages, an MPS may receive MPOA Resolution Requests from MPCs. In addition, in its capacity as an NHS, the MPS/NHS may receive NHRP queries from NHCs or peer NHSs. The MPS/NHS handles both types of queries.

If the routing and subnet convergence information available to the MPS/NHS indicates that the next hop is a directly connected MPC, then the Resolution Request is passed on to that LEC's MPC as an MPOA Cache Imposition Request. Otherwise, the request should be treated as a standard NHRP Resolution Request and forwarded or answered as indicated in [NHRP].

The MPS must maintain the status of all ingress cache entries (positive MPOA Resolution Replies) and egress cache entries (positive MPOA Cache Imposition Replies) that it has given to its MPCs. The MPS will generate the reply, and record the fact of the reply. If the information is later invalidated, a notification will go to the source of the Resolution Request. A destination may become invalid either because the actual host moved/expired, or due to a routing change.

4.5.1 MPS Configuration

MPSs must be capable of obtaining configuration information from an LECS. To obtain this configuration information, the MPS sends a Configuration Request with a MPOA Device Type TLV identifying the device as an MPS to the LECS. The LECS uses the same mechanism when mapping Configuration Requests with MPOA Device Type TLVs to configuration information as it does for standard LEC requests. The LECS may use the MPOA Device Type TLV to return only MPS specific TLVs. MPS TLVs may be returned by the LECS in any Configuration Response. MPS TLVs sent by the LECS override the default values of the corresponding MPS parameters. If no MPS TLVs are returned, the MPS must use default or locally configured values.

The MPS may be configured to bypass the configuration phase and use default or locally configured parameters. An MPS should retrieve configuration over only one of its LECs. MPS TLVs may be returned to a MPS's LEC during the LEC's initialization phase. The MPS may use this information without issuing another Configuration Request.

The TLV encodings of MPS configuration parameters information are specified in Section 5.2.1.

4.5.2 MPOA Resolution and NHRP Resolution

Ingress MPC-initiated MPOA Resolution includes a request phase and a reply phase, as shown in Figure 10. The role of the MPS in the Resolution process can be best described as that of a translator. An ingress MPC sends an MPOA Resolution Request to the appropriate ingress MPS. This ingress MPS translates the MPOA Resolution Request to an NHRP Resolution Request and forwards the Request on the routed path to the Internetwork-layer destination address (via its co-located NHS). When the NHRP Resolution Request arrives at the appropriate egress MPS, the egress MPS translates the NHRP Resolution Request to an MPOA Cache Imposition Request and sends it to the appropriate egress MPC. The egress MPC responds to the Cache Imposition Request by returning an MPOA Cache Imposition Reply to the egress MPS. The egress MPS then translates the MPOA Cache Imposition Reply to an NHRP Resolution Reply and forwards the reply on the routed path to the ingress MPS address in the NHRP Resolution Request (via its co-located NHS). When the ingress MPS receives the NHRP Resolution Reply, it translates the Reply to an MPOA Resolution Reply and returns it to the requesting ingress MPC. At the end of this process, the ingress MPC is prepared to establish and start using an MPOA shortcut and the egress MPC is prepared to receive data over the shortcut.

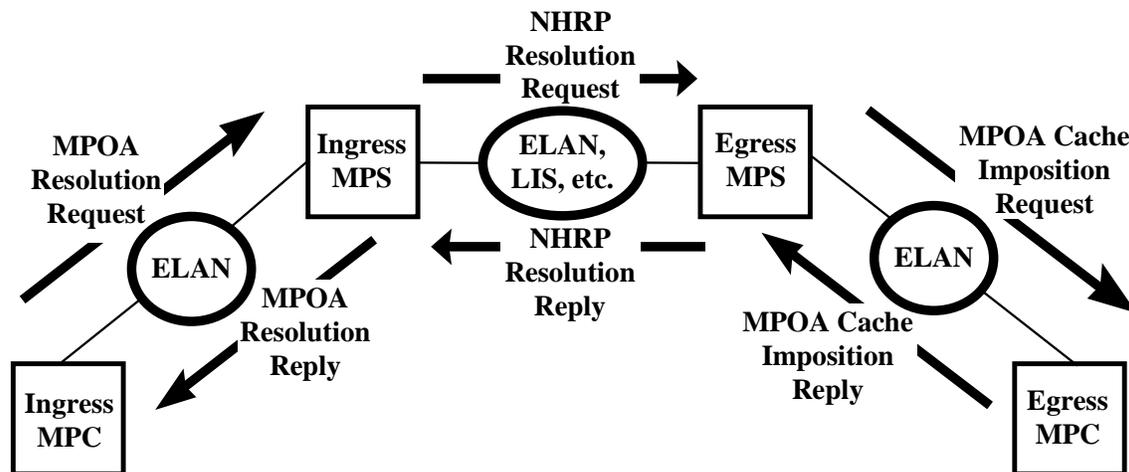


Figure 10 MPOA Resolution Process

MPOA Resolution Requests and MPOA Resolution Replies are identical in format to corresponding NHRP Resolution Requests and NHRP Resolution Replies except that a different Packet Type (`ar$op.type`) is used, as specified in Section 5.3.2.1. Distinction is required because MPCs are assumed to be associated with edge devices (i.e. bridges to LANs). This distinction results in protocol behavioral concerns not present with NHCs. Specifically, since the MPC does not necessarily have an internetwork layer address, the responding MPS/NHS may not be able to deliver the reply to the ingress MPC. Consequently, MPOA Resolution Requests are re-originated as NHRP Resolution Requests by the ingress MPS.

Re-origination ensures that corresponding NHRP Resolution Replies return to their point of origin. With MPS re-origination, MPOA models MPC-MPS relationships as distributed edge-routing: MPCs (as ATM addressed entities) are distributed forwarders associated with internetwork layer addressed (router co-resident) NHCs. Re-originated NHRP Resolution Requests contain the Source NBMA Address of the ingress MPC and the Source Protocol Address of the router co-resident with the ingress MPS. The original Source Protocol Address (if any) and Request ID are retained by the ingress MPS to be re-inserted in later conversion of NHRP Resolution Reply to MPOA Resolution Reply.

It is important that the association of MPC data ATM address and router (MPS) protocol address is not "learned" by down-stream Next Hop Servers; in particular, if the authoritative responder is itself a non-MPOA NHS, it may

misdelaiver protocol messages (e.g. the NHRP Resolution Reply) to the MPC. This learning by NHSs is prevented by setting the S and D bits in the NHRP Flags field to 0 in NHRP Resolution Requests and Replies respectively.

The translations required by the MPS are explained in the following subsections.

An MPS has both ingress functions and egress functions. It is possible for an MPS to serve as both the ingress MPS and egress MPS for establishing and maintaining a particular shortcut. In this case, the resolution process is still modeled in this specification as an ingress MPS exchanging NHRP messages with an egress MPS through an NHS. A strict implementation of this model would result in a double translation; however, an actual implementation is free to optimize this case as appropriate.

4.5.2.1 Translating MPOA Resolution Requests to NHRP Resolution Requests

When an ingress MPS re-originates an MPOA resolution request as an NHRP resolution request, it maps the tuple <source Control ATM address, request id> to a unique request id for the re-originated NHRP resolution request. This tuple is maintained until an NHRP Resolution Reply is received or the MPS Give Up Time (MPS-p6) has expired. When an ingress MPS receives an NHRP resolution reply, it must convert it to an MPOA resolution reply and forward it to the requesting MPC. The Source Control ATM address and destination protocol address are retained for the Holding Time specified in the NHRP resolution reply so that the ingress MPS can correctly direct NHRP Purges.

The MPS must forward the reply to the MPC at the source ATM address of the VCC over which the MPOA resolution request was received (i.e. the MPC control ATM address). The MPS must NOT forward the reply to the source ATM address contained in the NHRP resolution reply (i.e. the MPC data ATM address). This behavior is intended to support MPC's which use distinct ATM addresses for control and data.

4.5.2.2 Translating NHRP Resolution Requests to MPOA Cache Imposition Requests

Prior to responding to an NHRP Resolution Request for an MPC, the MPS must impose an egress cache entry in the egress MPC by sending an MPOA Cache Imposition Request and receiving an MPOA Cache Imposition Reply.

When an MPS receives an NHRP Resolution Request from its co-resident NHS, it checks to see if the router forwarding tables direct that internetwork layer destination address to one of the LECs known to the MPS. If so, the MPS communicates with the appropriate router convergence functions (e.g. IP ARP) to determine the DLL header for frames sent through the LEC to that destination. The MPS must then check with the LEC to see whether the LAN destination used to reach that internetwork layer destination is served by an MPC. This information, along with the ATM address of the MPC, is passed via LANE LE_ARP control frames in the Device Type TLV, and is returned by the LEC to the inquiring MPS. Once this information is obtained, the MPS converts the NHRP Resolution Request to an MPOA Cache Imposition Request.

The egress MPS does this conversion by copying all fields in the Fixed and Common Header except as follows:

- ar\$op.type is set to 0x80;
- Flags field is unused and must be set to zero;
- A new Request ID is generated.
- The Holding Time is set to at least twice the Holding Time the MPS will set in the corresponding NHRP Resolution Reply (MPS-p7 or as determined by local information).
- The prefix length is set or adjusted as appropriate.

In addition, message-specific portions and MPOA Extensions specified in Section 5.3.5 must be initialized. Included in this initialization is building and affixing to the message the MPOA DLL Header Extension. All NHRP Extensions included in the NHRP Resolution Request must be included in the MPOA Cache Imposition Request as well. The MPS must retain sufficient information from the original NHRP Resolution Request to allow subsequent mapping of MPOA Cache Imposition Reply to NHRP Resolution Reply.

4.5.2.3 Translating MPOA Cache Imposition Replies to NHRP Resolution Replies

If the MPS receives a negative Cache Imposition Reply (NAK), it is the co-located NHS's responsibility to decide whether to accept the shortcut itself and return one of its own ATM addresses, or return a negative reply (NAK).

If a successful reply is received by the MPS, state for detecting routing changes is saved, and the reply passed to the original source of the Resolution Request. The MPS converts successful MPOA Cache Imposition Replies to NHRP Resolution Replies by copying all fields in the Fixed and Common Headers and CIE except as follows:

- ar\$op.type is set to 2;
- Flags field is set as specified in [NHRP];
- Request ID is restored;
- Destination Protocol Address in the CIE is set to the egress MPS protocol address;
- Holding Time is restored to the value determined (i.e. less than or equal to half the holding time provided to the E-MPC).

Remaining NHRP Resolution Reply specific fields are filled in as specified in [NHRP].

If the NHRP Reply is generated with an MPC ATM address, the D (Destination Stability) bit must be zero to disable intermediate caching of the resolution.

The egress MPS must maintain state relative to all valid unexpired MPOA Cache Imposition Requests so that it may respond appropriately if the routing topology changes. If the cache imposition is successful, the egress MPS must maintain the mapping of internetwork layer address to DLL header and ATM address for the duration of the holding time it provides in the NHRP Resolution Reply as it would if it were forwarding the frames itself. In the case of IP, for example, the MPS must maintain its IP ARP cache entry in accordance with its locally configured ARP time-out parameter. If the holding time used in the NHRP Resolution Reply is greater than the IP ARP time-out, the MPS must re-verify the ARP when its time-out expires for the duration of the Holding Time. If a change is detected, the MPS must initiate the appropriate purge procedures.

4.5.2.4 Translating NHRP Resolution Replies to MPOA Resolution Replies

When the ingress MPS receives an NHRP Resolution Reply, the MPS converts this NHRP Resolution Reply to an MPOA Resolution Reply as follows. The MPS constructs an MPOA Resolution Reply containing the request ID and Source Protocol Address copied from the corresponding MPOA Resolution Request. It removes any TLVs it inserted. ar\$op.type is set to 0x87. All other fields are copied from the NHRP Resolution Reply.

When an I-MPS receives an MPOA resolution request for which it is the logical next hop, it has two choices of what to do:

1. NAK the request with an NHRP error code 12: "No Internetworking Layer Address to NBMA Address Binding Exists", and force the packets to be bridged via LANE.
2. Provide a reply using its own data ATM address.

Note that this will be a likely case when the MPS is in a router that is also the gateway to all off-campus destinations.

4.5.2.5 MPS to MPS NHRP

During the process of forwarding NHRP messages, an MPS/NHS may discover that the next hop is another MPS on a common ELAN as described in Section 4.2.

When an NHS co-resident with an MPS determines that a received NHRP message is to be forwarded to another NHS that is also co-resident with an MPS, the request shall be forwarded to the MPOA Control address of the peer NHS/MPS. The NHRP message shall be forwarded using AAL5 LLC/SNAP encapsulation as described in [NHRP].

When an MPS determines that the target of an NHRP Resolution Request is another MPS, rather than an MPC, it must forward the NHRP Resolution Request to that MPS as described above. This may occur when multiple MPSs share a common ELAN, and one MPS receives an NHRP Resolution Request for an internetwork layer address belonging to another MPSs on that ELAN.

4.6 Keep-Alive Protocol

MPCs need to know that MPSs that have supplied cache entries are alive and able to maintain those cache entries. As such, the MPS is required to periodically transmit an MPOA Keep-Alive message to all MPCs for which it has supplied and is maintaining ingress or egress cache entries. These must be sent every MPS-p1 seconds (subject to

jitter). The MPOA Keep-Alive may be sent over any LLC/SNAP VCC between the MPS and the MPC. Specifically, it may be sent over a point-to-multipoint VCC, including one established specifically for the purpose of transmitting the Keep-Alive.

The Keep-Alive message contains the control ATM address of the MPS, a Keep-Alive Lifetime value, and a sequence number. The Control ATM address of the MPS is used to correlate cache entries with a particular MPS. The Keep-Alive Lifetime, specified as MPS-p2, is the length of time that a Keep-Alive message is to be considered valid. If a Keep-Alive message is not received within Keep-Alive Lifetime seconds (specified in the previous Keep-Alive message), the MPC must consider the MPS to have failed. If subsequent Keep-Alive messages received by an MPC do not have sequence numbers that increase in value, the MPC must assume that the MPS has rebooted and, therefore, also has failed.

If an MPC detects that an MPS has failed, it must invalidate all cache entries provided by that MPS. Note that there is no requirement that a VCC (either point to point or point to multipoint) be maintained between an MPS and MPC. In practice, the keep-alive traffic itself will typically prevent the VCC from idling out, but this depends on the relative time-out values. In particular no inference about the state of an MPS should be drawn by an MPC if a VCC is released over which keep-alive messages were being received. The inference that an MPS is down should only be drawn after the Keep-Alive Lifetime has expired.

4.7 Cache Maintenance

Ingress and egress cache entries are created through the MPOA resolution process. Once created, these cache entries must be updated or removed as appropriate. This section describes the mechanisms provided to perform this cache maintenance.

The following purge mechanisms are used by MPOA components:

1. MPOA Cache Imposition Request from egress MPS to egress MPC to either refresh or purge a cache entry
2. MPC-Initiated Egress Cache Purge from egress MPC to egress MPS
3. NHRP Purge Request from ingress MPS to ingress MPC
4. NHRP Purge Request sent on the data plane from egress MPC to ingress MPC

4.7.1 Egress Cache Maintenance

An MPS must maintain state for all the MPOA and NHRP Resolution Replies and successful MPOA Cache Imposition Requests that it sources for the duration of the holding time it provides. The holding time provided by the MPS is viewed as a contract in that the MPS guarantees that, for the duration of the holding time, if the information it gave to another party changes, it will send a notification (update or purge) to that party. The recipient of the information is then free to use the information provided by the MPS for the duration of the Holding Time (unless it detects a change). The items for which an MPS maintains state are called “active cache entries”.

From the perspective of an egress MPS, active cache entries are those for which it has performed a successful MPOA Cache Imposition Request and answered an NHRP Resolution Request.

4.7.1.1 Egress MPS Purges and Cache Updates

When an egress MPS detects a change for a destination internetnetwork layer address affecting one of its active egress cache entries, it must:

1. Send NHRP Purge Requests to the set of affected sources of relevant resolution requests, and
2. Send an MPOA Cache Imposition Request with a holding time of zero to the egress MPCs with affected egress cache entries.

Or, it must:

1. Send an MPOA Cache Imposition Request with an updated DLL header to the egress MPCs to update the affected egress cache entries. If a DLL header update is sent, and the corresponding MPOA Cache Imposition Reply contains any new information (e.g., a new tag, a new data ATM, or different TLVs), then the MPS must send a purge as described above because the ingress MPC cannot be updated with an unsolicited resolution response.

The reasons that a relevant change may occur include:

- Routing has changed such that:
 - The egress MPS/NHS is no longer the NHRP Authoritative Responder, so that a received NHRP Resolution Request would be forwarded to another NHS;
 - The next internetwork layer hop has changed so that a received NHRP Resolution Request would cause an MPOA Cache Imposition Request to be forwarded to different MPC;
 - The internetwork layer next hop has changed so that a received NHRP Resolution Request would cause an MPOA Cache Imposition Request to be forwarded to the same MPC with a different DLL header.
- Bridging over LANE has changed such that:
 - The egress LEC has changed so that the shortcut needs to go to a different MPC than previously given (generally detected by LE_ARP).
 - An Egress Cache Purge Request has been received from an egress MPC

To update an egress cache entry, the egress MPS sends an MPOA Cache Imposition Request with the same egress Cache ID that was used on the original Cache Imposition Request and a non-zero holding time.

When an egress MPC receives an MPOA Cache Imposition Request with a Cache ID matching an active egress cache entry received from the same MPS, it must replace the fields in the current egress cache entry with corresponding fields in the new MPOA Cache Imposition Request.

To purge an egress cache entry, an egress MPS sends an MPOA Cache Imposition Request with the same egress Cache ID that was used on the original Cache Imposition Request and a zero holding time. The egress MPS may purge all egress cache entries in an MPC for a given destination protocol address by including the protocol address in the CIE and omitting the MPOA DLL Header Extension (which would contain the cache ID).

To ensure that correct updates are made in either the case of an update or a purge, the egress MPS must send the MPOA Cache Imposition Request to the egress MPC using the same VCC (or a VCC originating from the same egress MPS control ATM address and terminating at the same MPC ATM address) as was used for the original cache imposition.

Egress cache updates must be sent reliably using the retry mechanism described in Section 4.3.

4.7.1.2 Egress MPC Invalidation of Imposed Cache Entries

An egress MPC must invalidate any imposed egress cache entry for which the holding time has expired.

An egress MPC must invalidate all egress cache entries that originated from an egress MPS with which the MPC has lost communication, as described in Section 4.6.

If an egress MPC receives a packet on a shortcut, and the corresponding egress cache entry specifies a MAC destination address or destination Route Descriptor that is no longer in any of the associated LEC's variables (Local Unicast MAC Address(es) C6, Local Route Descriptor(s) C8, Remote Unicast MAC Address(es) C27, and Remote Route Descriptor(s) C30), it takes the following actions. The egress MPC may continue to forward packets to the bridge as if they came from the LEC interface for up to 30 seconds to allow normal bridge flooding and learning procedures to occur. If the condition does not change within the 30 seconds, the egress MPC must invalidate the cache entry and send an MPOA Egress Cache Purge Request (See Section 4.7.1.6) to the MPS that imposed that egress cache entry.

4.7.1.3 Invalidation of State Information Relative to Imposed Cache

An MPS must assume that all egress cache entries imposed by it to an egress MPC with which it has lost all communication may continue to be used until the Holding Time expires, or until it has sent egress MPS-initiated cache purges as described in section 4.7.1.1 and must not remove state information relative to these impositions. The MPS must expire this state information normally and may re-impose egress cache entries associated with remaining state information on restoration of the connection to the egress MPC.

4.7.1.4 Recovery From Receipt of Invalid Data Packets

An egress MPC that continues to receive data on a shortcut for which it does not have a valid egress cache entry must periodically send an NHRP Data Plane Purge Request on that shortcut to the ingress MPC as defined in Section 4.7.2.3. The frequency of these purges should not exceed one per second per source ATM address/destination internetwork layer address pair. This is required to recover from situations that may arise as a result of a lost cache imposition or incorrect shortcut usage by the remote end.

4.7.1.5 Egress Encapsulation

Cache impositions contain DLL encapsulation information as defined in an appropriate Annex to this document (e.g. - Annex A describes protocol specific encapsulation used for IP and IPX).

4.7.1.6 MPC-Initiated Egress Cache Purge

The MPC-Initiated Egress Cache Purge protocol provides the capability for an egress MPC to notify the egress MPS when it discovers an invalid egress cache entry. This notification is used by the egress MPS to issue associated NHRP Purge Requests. The MPC-Initiated Egress Cache Purge Request is most likely used when either the bridge topology has changed and a destination is no longer behind the same edge device, or the destination has aged out of the bridge forwarding table for lack of communication.

Information which must be included in an MPOA Egress Cache Purge Request is:

- Request ID
- Egress MPS Protocol Address
- Egress MPC Protocol Address or NULL
- Egress MPC Data ATM Address
- Destination Protocol Address (to purge)
- Destination Prefix Length
- MPOA DLL Header Extension
- no-reply flag

Upon receiving an MPOA Egress Cache Purge Request, the egress MPS must generate the appropriate NHRP Purge Request for the entry indicated in the Egress Cache Purge Request.

The no-reply flag (N-bit) is used to indicate whether the egress MPC wishes to receive an MPOA Egress Cache Purge Reply. It is recommended that the no-reply flag is always set by the egress MPC.

If the no-reply flag is cleared, an MPOA Egress Cache Purge Reply is expected and the MPS must clear the no-reply field in the associated NHRP Purge Request. When the egress MPS receives the associated NHRP Purge Reply, it issues an MPOA Egress Cache Purge Reply to the egress MPC. In the Egress Cache Purge Reply, the egress MPS returns all information provided by the egress MPC in the request.

If the no-reply flag is set in the MPOA Egress Cache Purge Request, the egress MPC does not expect to get an MPOA Purge Reply. If an egress MPC does not request an MPOA Egress Cache Purge Reply, it is a local matter to the egress MPS/NHS whether to request an NHRP Purge Reply.

If the shortcut is between an ingress MPC and an egress MPC, the NHRP Purge Request is sent to the ingress MPS (identified by its internetwork layer address) that re-originated the NHRP Resolution Request after receiving the original MPOA Resolution Request from the ingress MPC. The ingress MPS then forwards the NHRP Purge Request to the ingress MPC. Note that multiple ingress cache entries may be invalidated as a result of a single MPOA Egress Cache Purge Request. This is because the scope of the NHRP Purge Request includes all entries covered by the source, destination and internetwork layer destination addresses in the NHRP Purge Request and is not restricted to the source and destination ATM addresses of the shortcut.

4.7.2 Ingress Cache Maintenance

4.7.2.1 MPOA Trigger

An MPC must be able to detect inbound data flows and establish shortcuts. In addition, an ingress MPS may detect inbound data flows and request that ingress MPCs establish shortcuts for them. A trigger mechanism is used such that the rest of the protocol remains consistent with the MPC-initiated mechanism.

In the event that an ingress MPS determines the need for a shortcut for an inbound data flow, the ingress MPS may trigger the appropriate ingress MPC into initiating an MPOA Resolution Request for that flow. This is done using an MPOA Trigger that describes the inbound data flow to be shortcut. The ingress MPC must create an ingress cache entry for the flow, if one does not already exist and if it has the resources to establish another shortcut, and must respond by initiating MPOA Resolution Requests for the target indicated in the MPOA Trigger.

An ingress MPS must use the MPOA retry procedure defined in Section 4.3 to control the sending of MPOA Trigger messages (note that the receipt of a corresponding MPOA Resolution Request by the triggering MPS is considered to be the reply for a given MPOA Trigger message).

Information provided in an MPOA Trigger is:

- Ingress MPS Control ATM Address
- Destination Internetwork Layer Address and Address Prefix

The meaning and use of these information fields is given in the sections below.

Ingress MPS Control ATM Address

This address is required for an ingress MPC to build an ingress cache entry and identify the inbound datagrams that will be sent on the shortcut to be established as a result of an MPOA Resolution Request.

Destination Internetwork Layer Address and Address Prefix

The protocol address to be used in the triggered MPOA Resolution Request. This address is also required for an ingress MPC to build an ingress cache entry and identify the inbound datagrams that will be sent on the shortcut to be established as a result of a successful receipt of a corresponding NHRP Resolution Reply. The MPS may indicate that an address prefix should be requested by adding a CIE with the prefix set appropriately. The use of prefixes at the MPC is optional, however, and the MPC may issue the resolution request for just the single protocol address.

4.7.2.2 Ingress MPSs and NHRP Purges

When an ingress MPS receives an NHRP purge request, it must send NHRP ingress cache purge requests to all relevant MPC's for which it is maintaining ingress state for the purged destination address(es).

If the no-reply flag is clear in the received NHRP purge request (meaning an NHRP purge reply is requested), the ingress MPS must respond to the NHRP purge request. The ingress MPS should not use the retry procedure, defined in Section 4.3, to ensure reliable delivery of NHRP ingress cache purge requests to relevant MPCs. If the ingress MPS expects a response from an MPC, it must clear the no-reply flag in its generated NHRP ingress cache purge request, and use the retry procedure. The ingress MPS may send an NHRP purge reply without waiting for NHRP ingress cache purge replies from all MPC's.

When processing an NHRP ingress cache purge request, the ingress MPC should not use the received source address information to decide which entries to purge (see also Appendix VII.2).

4.7.2.3 Data Plane Purge Protocol

Under certain circumstances it is necessary to send an NHRP Purge Request on the data plane to tell the ingress MPC or NHC that ingress cache entries are no longer valid.

The different conditions under which an egress MPC is required to send NHRP Purge Requests over the shortcut are described below:

Egress MPS Dies: If an egress MPC fails to receive an MPOA Keep-Alive message from an MPS that has imposed egress cache entries within the MPOA Keep-Alive Lifetime (as specified in the last received MPOA Keep-Alive message) then it may send NHRP Purge Requests which invalidate all the cache entries imposed by the failed MPS, which are currently associated with a shortcut VCC. Alternatively, the egress MPC may invalidate these cache entries locally and send NHRP Purge Requests only when data is received that uses these entries. If there is no open VCC to the source ATM address as specified in an egress cache entry, it is not necessary to establish a VCC for the purpose of sending an NHRP Purge Request.

Egress Cache Miss: If an MPC receives a packet over a shortcut, but the egress cache lookup fails, the MPC must send an NHRP Purge Request over that shortcut to inform the ingress MPC to remove the appropriate ingress cache entry. In this case, the egress MPC does not know to which MPS to send an Egress Cache Purge request.

Note: An egress cache Miss can occur for several reasons:

1. Destination internetwork layer address not found.
2. Invalid tag.
3. IP/tag/source ATM address not consistent (e.g. tag hit, but wrong destination IP address or destination IP hit, but wrong source ATM).

The Data Plane Purge mechanism uses the NHRP Purge Request frame format as described in Section 5.3.11. When an ingress MPC receives the NHRP Purge Request on the shortcut, it must do the following in this order: authenticate the NHRP Purge Request if authentication was used, process any vendor private Extensions, process MPOA-specific Extensions, and purge the ingress cache as appropriate. In the case of conflicting information in Extensions, the previously specified order also specifies the priority for conflict resolution (e.g., do nothing if authentication fails.)

4.8 Connection Management

This section specifies the procedures for MPOA connection management in their entirety. These procedures are derived from, and are intended to be compatible with, those described in [RFC 1755], which describes procedures for establishing and clearing VCCs in a multiprotocol environment. In some cases text from [RFC 1755] is reproduced here in this specification either unmodified or slightly modified, without an explicit reference that the text has been taken from that source, but the work done by the authors of that RFC is hereby acknowledged.

4.8.1 Generic VCC Management Procedures

MPOA components must support the use of LLC/SNAP encapsulation for all PDUs. By default VCCs must be signaled to use LLC encapsulation. The negotiation of other encapsulations, such as the null encapsulation, is not precluded, but an MPOA component is not required to support any encapsulation other than LLC/SNAP.

Because the connection management procedures use VCCs with LLC encapsulation, many of the procedures are generic procedures that can be used by any protocol. The same VCC may be used to carry both MPOA control and data traffic. Also the same VCC may be used to carry both MPOA and non-MPOA traffic, as long as the non-MPOA traffic uses LLC encapsulation. Where the MPOA protocols require specific rules or default values these are explicitly indicated. Apart from that, the text can be read as applying to any protocol that uses LLC encapsulation in an MPOA-capable device.

Although LLC encapsulation allows the sharing of a VCC by multiple protocols, it does not require it. Stations can control the ATM addresses that they advertise for different protocols (using a different ATM selector for example), forcing separation. Also, when a station is transmitting traffic, it may establish multiple VCCs between the same two endpoints, each carrying a different protocol, for example.

Communication between multiple local and remote protocol entities may use a single VCC if the local entities all are sharing an ATM address and the remote entities are all sharing an ATM address. Such sharing is facilitated by using LLC multiplexing on the VCC.

The MPOA specification imposes rules on the allocation of ATM addresses within an MPOA device and, as a result, on what entities may share a VCC. For example each MPC in a device must use a distinct ATM control address. Also, the ATM address assignment for LANE Data Direct VCCs is constrained as described in Section 4.2.3. The sharing of VCCs is thus always constrained by the overarching ATM address assignment rules. Within those constraints an implementation is free to allocate the same, or different, ATM addresses to different protocol entities. For example an MPC may choose to use one ATM address for MPOA shortcut VCCs, and another for LANE LLC Data Direct VCCs, or it may choose to use a single ATM address for both.

Note that there is no necessity to have one particular protocol or protocol suite as a primary owner of a VCC. Whether there is one protocol that "owns" a VCC and allows sharing, or a separate VCC management entity to which all protocols make requests as peers, is purely an implementation issue and as such is outside the scope of this specification. Note that because the VCCs are potentially shared, it is not possible to deduce status information about a particular protocol based on status information of a particular VCC. In particular, it is not possible to deduce that a protocol entity is operational just because a VCC has been established, as the process or task implementing that protocol could be non-operational.

4.8.2 Scope of MPOA VCCs

PDU's are sent between MPOA components and also between MPOA components and non-MPOA components, in particular between MPCs and NHCs that do not also include MPOA functionality. Note that packets transmitted between MPOA capable routers use NHRP between the co-located NHSs. It is possible to mix routers with MPSs and routers with NHSs in a network. An NHRP Resolution Request issued by an MPS may be answered by a router with an NHS; for example, if the router has directly attached Ethernet hosts, and is the egress router for those hosts, it may answer the Resolution Request.

An MPOA component must be capable of establishing, receiving and maintaining a VCC to any entity that conforms to the connection management procedures specified in this document, whether or not that entity is an MPOA component.

4.8.3 Initiating VCCs

VCCs are established when needed. If an MPOA component has a datagram to send and there is no existing VCC that it can use, or it chooses not to use an existing VCC, then it establishes a VCC to transfer the datagram. Both MPCs and MPSs may initiate calls. For example an MPC may initiate a call to transfer an MPOA Resolution Request, and an MPS may initiate a call to transfer an MPOA Trigger or NHRP Purge Request.

When an MPOA component has a datagram to send, it should first look to see if there is an existing VCC that it can use. There may be an existing VCC to the correct ATM address that it chooses not to use, for example, due to a mismatch in AAL5-SDU sizes or because the additional traffic could violate the Traffic Descriptor used when the VCC was first established. In some cases, an MPOA component may choose not to use an existing VCC for its own local purposes, such as to achieve protocol separation. If an MPOA component chooses not to use an existing VCC, then it must attempt to establish a new VCC.

4.8.4 Receiving Incoming VCCs

When an incoming VCC is established that indicates the use of LLC encapsulation, there is no information conveyed by the UNI signaling protocol about what protocols will subsequently be used over the VCC. For example, the originator may use the VCC for control or data traffic or both. Unless limited by resources, MPOA components should accept all incoming VCCs that indicate the use of LLC encapsulation.

Any form of security checking before incoming call acceptance (e.g. by determining if the calling ATM address belongs to a set of valid addresses) is outside the scope of this specification. An MPOA component must be prepared to receive incoming calls originated in the same ELAN or in a different ELAN, if it has any mechanism to differentiate between the two based on the calling party ATM address. Any such mechanism is outside the scope of

this specification. An MPOA component must not make any assumptions at call setup time about the type of traffic (e.g. Control or Data) to be used on a VCC based on information as to whether the Calling Party is in the same or a different ELAN.

4.8.5 Support for Multiple VCCs

MPOA components must be able to support multiple VCCs between peer systems, without regard to which peer system initiated each VCC. When an incoming call is accepted, an MPOA component must be prepared to receive incoming PDUs on that VCC. It may also transmit PDUs on that VCC. It must not accept and immediately clear the incoming call, or ignore PDUs received on the VCC.

Allowing multiple VCCs is primarily intended for cases where the VCCs have different attributes, such as Traffic Descriptor, Quality of Service requested from the network, or AAL5-CPCS-SDU size. It is recognized that independently of these considerations two MPOA components may simultaneously initiate calls to each other, leading to duplicate identical VCCs. To avoid the wasted resources of unintentional duplicate VCCs, the following mechanism is defined to allow one VCC to be cleared due to inactivity, with all traffic carried on the other VCC.

When an MPOA component has a datagram to send, and it detects that there are more than one VCC that are capable of conveying the packet, and that the MPOA component does not wish to use more than one VCC, it should send the packet on the VCC initiated by the party that has the numerically lower ATM address. In this way both parties will use a single VCC, allowing the other VCC to time out due to inactivity. If an MPOA component switches over to use a VCC in this way, it may set the inactivity timer to a small value for the VCC that it does not intend to use, provided that it was the initiator of the VCC that it does not intend to use. If an MPOA component finds during this process that there are multiple VCCs initiated by the party with the lower value of ATM address, it may choose to use any one of them.

Specific protocols that use LLC encapsulation may impose more restrictive rules. In particular, a protocol may mandate that only one VCC be used between a pair of end stations to transfer traffic of that protocol between those two end stations. Such a per-protocol restriction does not affect the use of multiple VCCs by other protocols in the same box, and may coexist in a device that uses protocols that allow multiple VCCs.

4.8.6 Internetwork Layer-to-ATM Address Mapping

MPOA components are not required to support the InATMARP protocol as defined in [RFC 1577]. One reason for this is that MPCs are not required to have any internetwork layer addresses. An MPOA component is not permitted to learn Internetwork Layer Address-to-ATM Address mappings as a result of using the InATMARP protocol on VCCs.

An MPOA component must not learn Internetwork Layer-to-ATM Address mappings by learning from the Source Protocol and Source NBMA Address fields in an MPOA/NHRP Resolution Request. The only method used to learn Internetwork Layer Address-to-ATM Address mappings, apart from static configuration, is to issue an MPOA/NHRP Resolution Request and learn from the MPOA/NHRP Resolution Reply. One reason for this is that there may be asymmetrical routes in the network. Another is that the Source Protocol address in an NHRP Resolution Request may be that of an MPS, and the Source NBMA Address that of a physically separate MPC.

4.8.7 Establishment of Bi-directional Data Flow

When a VCC is established by an ingress MPC to an egress MPC, traffic in the reverse direction (egress MPC to ingress MPC) may use that same VCC, as described in 4.4.4.1, if the signalling parameters in that direction are suitable. If desired, an MPOA component can establish a dedicated unidirectional VCC by specifying a return PCR of zero.

4.8.8 VCC Termination

There is no requirement that a VCC be permanently maintained between two MPOA components. Either the Calling Party or the Called Party may clear the VCC. If a VCC is terminated by the remote party, a station should not immediately re-establish the VCC unless it has some PDUs to transfer.

In general, an MPOA component should allow VCCs to idle out based on inactivity. Additionally, an MPOA component may decide to release the least recently used VCC to free up resources for a new VCC, or as a reaction to certain error conditions, such as persistent protocol errors due to traffic on a certain VCC.

4.8.9 Use of UNI Signaling Information Elements

MPOA control and data PDUs may be transferred on any suitable VCC. Such a VCC may have previously been set up to transfer a PDU for a non-MPOA protocol, but that has not yet timed out. If an MPOA component wishes to setup a new VCC to transfer an MPOA PDU, then the following rules apply with regard to the encoding of UNI signaling information elements. It is not a requirement that an MPOA PDU be transferred over a VCC that was established in accordance with these rules. It is a requirement that an MPOA component be capable of establishing a VCC according to these rules.

Shortcut data flows, as specified in this version of MPOA, are always carried over point-to-point VCCs. In general, Control PDUs will also be carried over point-to-point VCCs, but point-to-multipoint VCCs may also be used in specific cases. An MPOA component must be able to be added as the first or subsequent party of a point-to-multipoint call. The use of the signaling IEs is the same as that outlined here for point-to-point calls, subject to the constraints imposed by [UNI 3.0, UNI 3.1 or UNI 4.0]. The main difference is that call characteristics, such as Traffic Descriptor, QoS, or AAL5 Parameters, can only be negotiated between the Calling Party and the first Called Party. The second and subsequent Called Parties cannot engage in any negotiation. An MPOA component is not required to be able to initiate point-to-multipoint calls for Control PDUs. Note that the intra-ELAN multicast and broadcast data flows are handled by the LANE BUS, and are transparent to MPOA. The use of shortcuts for multicast data flows, i.e. bypassing multicast routers, is not supported in this specification.

To enhance interoperability, an MPOA component must treat as equivalent the presence of a parameter set to a null or zero value, and the absence of that parameter. All unused and reserved fields must be set to zero on transmission and ignored on receipt. A received packet that has non-zero values in these fields must not be treated as an error.

4.8.9.1 Traffic Descriptor

All MPOA components must be capable of initiating and accepting VCCs with the UBR service category. The support of VCCs with other than the UBR service category is allowed, but not required.

For transferring control messages, an MPOA component should initiate a VCC with the UBR service category. If an MPOA component attempts to set up a VCC using anything other than the UBR service category, for the purposes of transferring control messages, and the VCC establishment fails as a result of either the network or the remote party being unable to support a non-UBR service category, the MPOA component must retry using the UBR service category.

For shortcuts, an MPOA component must be capable of initiating a VCC that proposes the UBR service category. It may propose any service category, but must be prepared to deal with either the network or the remote party rejecting the call due to being unable to support the proposed non-UBR service category. In this case the MPOA component should retry using the UBR service category.

The mechanisms by which an MPOA component decides to initiate a VCC that uses anything other than the UBR service category, are outside the scope of this specification. Different mechanisms are possible such as monitoring the data flow, or monitoring or participation in a resource reservation mechanism like RSVP [RSVP].

UNI 3.x Signaling does not provide for ATM Traffic Descriptor or Quality of Service negotiation. UNI 4.0 Signaling does provide for the negotiation of these parameters within an ATM service category, but does not permit the negotiation of the ATM Service Categories themselves. To determine what ATM Service Categories a target station supports, an MPOA component should use the ATM Service Category Extension in advance of VCC establishment. This capability reduces the probability that a VCC will be rejected because the Service Category cannot be supported, with the result that the calling party has to try again.

When an ingress MPC sends an MPOA Resolution Request, it should add a single ATM Service Category Extension, as specified in Section 4.4.2, to identify the Service Categories it supports. If only UBR is supported the extension should still be added, and will take a zero value. When an egress MPC responds to an MPOA Cache

Imposition Request with an MPOA Cache Imposition Reply, it should fill in the ATM Service Category Extension to identify the Service Categories it supports, if the extension was included in the request, as specified in Section 4.4.3.

When an ingress MPC knows, through receipt of an ATM Service Category Extension in an NHRP Resolution Reply, that the desired Service Category is supported on the target MPC, it may attempt to set up the shortcut with that Service Category. If the first attempt for the call setup does not succeed, the MPC may attempt with another Service Category that both ends support. The MPC may attempt to setup a shortcut with the UBR Service Category at any time.

The PCR used in the ATM Traffic Descriptor should be set to line rate. It may be set to less than line rate as a local configuration option. This may be useful if it is known that there are slower speed links in the network, and that traffic shaping to the slower speed by a device may reduce frame loss. As specified in section 3.6.2.4 of [UNI 3.1], for best effort traffic, a user need not conform to the signaled PCR, and the network may enforce a PCR different than the signaled PCR. It is recommended, however, that if a user signals a PCR less than line rate that it conform to the PCR.

The valid combinations of values of the Broadband Bearer Capability, Traffic Descriptor and QoS Class IEs are defined in Appendix F of [UNI 3.1].

Use

This IE must be included in a SETUP message.

Format

Field	Value
Forward Peak Cell Rate (CLP=0+1) ID	132
Forward Peak Cell Rate (CLP=0+1)	line rate
Backward Peak Cell Rate (CLP=0+1) ID	133
Backward Peak Cell Rate (CLP=0+1)	line rate
Best Effort Indication	190

UNI 3.0 considerations

In UNI 3.0, issues of traffic management were less well understood than in UNI 3.1. UNI 3.0 does not contain a guide to coordinating the use of the User Cell Rate IE (Traffic Descriptor in UNI 3.1), Broadband Capability IE, and QoS Parameter IE. It is recommended that the use of these IEs in UNI 3.0 should be the same as for UNI 3.1.

The value for the cause "User Cell Rate" is 51 in UNI 3.0, and 37 in UNI 3.1

UNI 4.0 considerations

An MPOA component must support the ability to set both the forward and backward frame discard bits. An MPOA component must use the frame discard capability. This capability may be used with any ATM service category. The use of frame discard increases throughput under network congestion. An MPOA component must not treat the reception of an ATM Traffic Descriptor IE which does not indicate frame discard as an error.

Traffic parameter negotiation using the Minimum Acceptable ATM Traffic Descriptor IE is an optional UNI 4.0 feature. If available an MPOA component should use this capability. This capability may be used with any ATM service category. The use of traffic parameter negotiation allows the calling and called parties to discover the smallest bandwidth limitation along the path of the connection. With this information a source may then take measures to reduce the possibility of network congestion. For example a source may then perform traffic shaping down to the smallest bandwidth, or may perform congestion control at a higher layer on an end to end basis. Even if the calling party is not capable of using the information about the smallest bandwidth the Minimum Acceptable ATM Traffic Descriptor IE should be used, as the called party may be capable of doing so.

When using traffic parameter negotiation with the UBR service category the PCR should be set to link rate in the ATM Traffic Descriptor IE, and the PCR should be set to zero in the Minimum Acceptable ATM Traffic Descriptor

IE. Note that the value of zero in the Minimum Acceptable ATM Traffic Descriptor IE will be treated by the network as meaning "unspecified", not that zero bandwidth is acceptable.

When progressing a call the network will adjust the ATM Traffic Descriptor IE to reflect the actual bandwidth available. The network may drop the Minimum Acceptable ATM Traffic Descriptor IE from the SETUP message, if it and ATM Traffic Descriptor IE are equivalent. An MPOA component must not treat the reception of a SETUP message without the Minimum Acceptable ATM Traffic Descriptor IE as an error.

4.8.9.2 QoS Parameter

Class 0 is the only class allowed when using the UBR Service Category. The use of Classes other than Class 0, is outside the scope of this specification.

Use

This IE must be included in a SETUP message.

Format

Field	Value
QoS Class Forward	0
QoS Class Backward	0

UNI 3.0 considerations

In UNI 3.0, the two-bit coding standard field is set to "00". In UNI 3.1 and later, this field is set to "11" as the ITU-T has now standardized QoS class 0. MPOA components should treat both values as equivalent.

UNI 4.0 considerations

UNI 4.0 allows, for some ATM service categories, the signalling of individual QoS parameters for the purpose of giving the network and called party a more exact description of the desired delay and cell loss characteristics. This capability uses the End to End Transit Delay IE and the Extended QoS Parameters IE. However UNI 4.0 does not allow these IEs to be used with the UBR service category, and the only QoS Class allowed for UBR is Class 0. See Table A9-2 in [UNI4.0] for the allowed combinations of QoS parameters and ATM service categories.

4.8.9.3 AAL5 Parameters

The only AAL5 parameter that may be negotiated is the CPCS-SDU size. This is the maximum number of octets that may be transferred in an AAL5 frame on that VCC. This includes any Data-Link Layer octets (e.g. MAC and LLC sub-layers). The term "MTU size" refers to the size of the internetwork layer PDU (e.g. NHRP packet starting with a\$afn byte, or IP packet starting with the version number field). The MTU size does not include any MAC or LLC octets.

For MPOA components, the default CPCS-SDU size is 1536 octets. All MPOA components must support initiating and receiving VCCs that use a CPCS-SDU size of the default size in both the forward and backward directions.

An MPOA component may attempt to negotiate a larger CPCS-SDU size, in accordance with the procedures specified in Annex F of [UNI 3.1]. When doing so, the proposed value of CPCS-SDU must not be less than the default size. When an MPOA component proposes the use of a larger size than the default, it must be prepared to accept the remote party negotiating the value downwards. If an MPOA component receives an incoming call that proposes a larger value than the default, it must not treat this an error. Instead, it may accept the proposed value if it can support that, or use a smaller value that must not be less than the default value. The negotiated value is not restricted to being one of the LANE maximum frame sizes. For example two MPOA hosts could negotiate a size of 65535 bytes. If a remote party negotiates the CPCS-SDU down to a value lower than the default size, an MPOA component may use choose to use the VCC (performing any necessary fragmentation) or to release the VCC.

If an MPOA/NHRP Resolution exchange was used to obtain the remote ATM address to which an MPOA component wishes to set up an SVC, then the desired MTU of the remote party should be known in advance of SVC establishment, as an MPOA component is required to include a non-zero value for desired MTU size in an MPOA Cache Imposition Reply. If there is no explicit indication of MTU size in an MPOA/NHRP Resolution Reply then a value of 9180 should be assumed as the desired MTU size of the remote party. If LLC/SNAP is to be used on the SVC then the CPCS-SDU size should be at least 8 bytes greater than the MTU size, to allow space for the LLC/SNAP header. If the SVC is to be used for data packets using the MPOA tagged encapsulation the CPCS-SDU size should be at least 12 bytes greater than the MTU size, to allow space for the MPOA tag and LLC/SNAP header.

If an MPOA/ NHRP Resolution exchange was not used to obtain the remote ATM address to which an MPOA component wishes to set up an SVC, then the default CPCS-SDU size appropriate to the type of interface (LANE or Native ATM) over which the destination is reached should be used. Typically this will be the case for SVCs that are setup to adjacent Components on the same ELAN/LIS for the purpose of transferring Control PDUs. For an ELAN the MPOA default value of 1536 should be proposed, independent of the maximum frame size or emulation type in use on that ELAN. One reason for this is that many LECs, with different parameters, may be served by a single MPC. This implies that an edge device that only supports Token-Ring, for example, may be required to send its control flows over SVCs that use the MPOA default size. For Native IP over ATM the CPCS-SDU default size is 9188 (9180 MTU + 8 LLC/SNAP). Thus for example if an MPS is setting up an SVC to a next hop NHS reached over a Native ATM interface it should propose a CPCS-SDU size of 9188.

The CPCS-SDU sizes that an MPOA edge device will wish to use are determined by the maximum frame sizes of the LAN media attached to the edge device. An MPC is not required to do fragmentation of internetwork layer packets. If after an exchange of NHRP packets an MPC determines that to meet the MTU capabilities of the remote party it must perform fragmentation, it may choose to keep using the default hop-by-hop LANE path and not establish a shortcut. An intermediate router will perform any necessary fragmentation. Note that if possible all hosts in an MPOA network, both LAN and ATM attached, should use mechanisms such as Path MTU Discovery to reduce or eliminate the requirement for the network to fragment packets, as the inefficiencies due to performing fragmentation may be significant.

For MPOA, the AAL5 adaptation layer IE must always contain values for both the Forward and Backward CPCS-SDU sizes, and the IE itself must be included in both SETUP and CONNECT messages. The requirement to include the AAL5 adaptation layer IE in a CONNECT message is an additional requirement over the Annex F of [UNI 3.1] procedures, but is required by [RFC 1755].

The SSCS type parameter should be omitted. If present it must be coded to indicate the null SSCS (a value of zero). An MPOA component must treat the presence of this parameter with a zero value, and the absence of the parameter as equivalent.

Use

This IE must be present in both SETUP and CONNECT messages.

Format

Field	Value
AAL Type	5
Forward Max SDU Size ID	140
Forward Max SDU Size	as desired
Backward Max SDU Size ID	129
Backward Max SDU Size	as desired

UNI 3.0 considerations

For UNI 3.0, the mode parameter should be omitted. If included it should be set to 1 to indicate message mode. This parameter must be ignored by an MPOA component. For UNI 3.1 and later the mode parameter is illegal.

The value for the cause "AAL Parameter Cannot be Supported" is 93 in UNI 3.0 and 78 in UNI 3.1. (Note that the value of 78 for this error indication in UNI 3.1 is an error and the correct value should have been 93).

MPOA & NHRP-only interoperability considerations

If a NHRP-only Component does not support CPCS-SDU size negotiation then it will not be possible to establish an SVC between it and an MPOA component that does not support the NHRP default size. This situation may be recognized if an MPOA component initiates an SVC that gets rejected with the error code for “AAL5 parameters cannot be supported”, or an MPOA component accepts an incoming call but negotiates the CPCS-SDU value down, and then finds that the call is immediately released by the remote party, with the same error code. An MPOA component should recognize this situation and stop attempting to establish a VCC to those Components that exhibit this behavior.

4.8.9.4 B-LLI

The purpose of the B-LLI IE is to provide a means to be used for compatibility checking by an addressed entity. It specifies the layer 2 and/or layer 3 protocol, and thus the encapsulation, that the calling party plans to use on the VCC being established.

LLC/SNAP encapsulation is the default encapsulation for MPOA, and this encapsulation must be implemented by all MPOA components. The use of B-LLI negotiation is allowed, but is not required. An MPOA component must include a B-LLI IE in a SETUP message encoded as shown in the table below. The layer 2 information indicates LLC and there is no layer 3 information.

An MPOA component may implement B-LLI negotiation procedures as defined in Annex C of [UNI 3.1]. In this way an encapsulation other than the default may be negotiated and used. An example of an alternative encapsulation is the null encapsulation (VCC-multiplexing), as described in [RFC 1483], where an internetwork layer packet is carried in the AAL5 CPCS-PDU payload, with no Data-Link Layer information included. Up to three instances of the B-LLI IE may be included in a SETUP message. The order of appearance of the B-LLI IEs indicates the order of preference, i.e. most favored B-LLI is first. If this negotiation is used, one of the B-LLI IEs must indicate the use of LLC, as described above. An MPOA component receiving a SETUP must not assume that the B-LLI IE indicating LLC is the first or only B-LLI IE in the SETUP message. This applies even if the MPOA component does not support the use of any encapsulation other than LLC/SNAP. This allows a station that proposes negotiation to interoperate with one that does not.

A single B-LLI IE may be included in the CONNECT message. If it is not included, it means that the called party accepts the first (or only) B-LLI IE in the SETUP message. If it is included, it means that the called party is explicitly indicating the B-LLI IE it is accepting.

Use

This IE must be used in a SETUP message. It may be used in a CONNECT message.

Format

Field	Value
Layer 2 ID	2
User Information Layer 2 Protocol	12 LAN LLC (ISO 8802/2)

4.8.9.5 Broadband Bearer Capability

An MPOA component must support the use of both BCOB-C and BCOB-X. It must be able to signal either capability, and receive an incoming call that uses either capability. If the call is rejected due a problem with the bearer capability (causes 57 - not authorized, 58 - not presently available, or 65 - not implemented), it should retry using the other value.

When a user specifies BCOB-C, the user is requesting more than an ATM-only service. The network may look at the AAL and provide service based on it. When the user specifies BCOB-X, the user is requesting an ATM-only service from the network, and the network shall not process any higher layer protocols (e.g. AAL protocols). Note that in UNI 4.0 the frame discard feature can be used with both BCOB-C and BCOB-X.

This specification recommends that BCOB-X be the default, with BCOB-C used when configured to do so, or when a call using BCOB-X failed due to the causes listed above.

Appendix F of [UNI 3.1] specifies additional rules for the encoding of this IE, that supplement those specified in section 5.4.5.7 of [UNI 3.1]. Note that octet 5a must be absent if BCOB-C is used.

Use

This IE must be present in a SETUP message.

Format

Note: the table shows the default value, not the only valid one

Field	Value
Bearer Class	16 (BCOB-X)
Traffic Type	0 (no indication)
Timing Requirements	0 (no indication)
Susceptibility to Clipping	0 (not susceptible)
User Plane Connection Configuration	0 (point to point)

4.8.9.6 ATM Addressing information

Addressing information is conveyed using the following IEs:

- Calling Party Number
- Called Party Number
- Calling Party Subaddress
- Called Party Subaddress

An MPOA component must support the use of private ATM addresses, and must support the use of all three formats of private ATM addresses (DCC format, ICD format, E.164 format). An MPOA component may support the use of Native E.164 ATM addresses (for which there is a single format). The control ATM address of an MPOA component must be a private ATM address.

An MPOA component is not required to support the use of the Calling Party Subaddress or Called Party Subaddress IEs. These IEs are used to carry a private ATM address across a public network that supports only Native E.164 Addresses. LANE does not require the use of the Calling Party Subaddress or Called Party Subaddress IEs and MPOA makes no additional requirements in this regard. An MPOA component may support the use of the Calling Party Subaddress or Called Party Subaddress IEs.

Use

The Calling Party Number and Called Party Number IEs must be used in a SETUP message. The Calling Party Subaddress and Called Party Subaddress IEs may be used in a SETUP message.

Format

The formats for private ATM addresses are shown. The coding of Calling/Called Party Number using Native E.164 ATM addresses and the coding of the Subaddress IEs are as defined in [UNI 3.1].

Calling Party Number

Field	Value
Type of Number	0 (unknown)
Addressing/Numbering Plan Identification	2 (ATM endsystem address)

Presentation Indicator	0 (presentation allowed)
Screening Indicator	1 (user provided verified & passed)
Address Octets	address value

Called Party Number

Field	Value
Type of Number	0 (unknown)
Addressing/Numbering Plan Identification	2 (ATM endsystem address)
Address Octets	address value

UNI 3.0 considerations

In UNI 3.1, the ATM Endsystem Address type was introduced to differentiate ATM addresses from OSI NSAPs. In UNI 3.0, 'ATM Endsystem Address' is not a valid type. Therefore, in the Called and Calling Party Subaddress IEs, the three-bit 'type of subaddress' field must specify 'NSAP' (value = 001) when using the Subaddress IE to carry ATM addresses.

5. MPOA Frame Formats

[Normative]

5.1 Encapsulation

5.1.1 Data Frame Encapsulation

By default, MPOA uses LLC encapsulation for all data flows in accordance with the rules defined in [RFC 1483]. The default shortcut data encapsulation is the RFC 1483 LLC Encapsulation for Routed Protocols, shown in Figure 11. MPOA allows the negotiation of alternative encapsulations (e.g. Null Encapsulation) using the B-LLI negotiation procedures defined in ATM Signalling [UNI 3.0, UNI 3.1, UNI 4.0]. All MPOA devices must be able to use the default encapsulation for all data flows on all VCCs. Negotiation of other encapsulations is optional.

MPOA also allows the optional use of the MPOA Tagged Encapsulation shown in Figure 12 for data flows. The format for tagged packets consists of an 8-byte LLC/SNAP header, a 4-byte tag field, and the Internetwork layer packet. The tag field must be used to determine the Internetwork layer protocol type of the packet which follows. There may be a mix of tagged and non-tagged packets on a VCC.

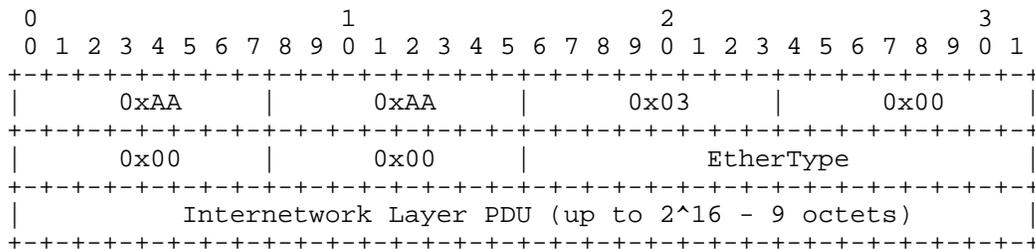


Figure 11 RFC 1483 LLC Encapsulation for Routed Protocols

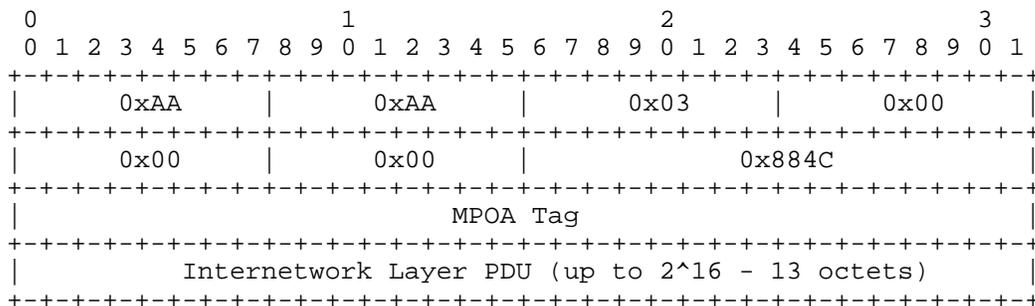


Figure 12 MPOA Tagged Encapsulation for Routed Protocols

5.1.2 Control Frame Encapsulation

By default, MPOA uses LLC encapsulation for all control flows as defined in [NHRP] and shown in Figure 13. MPOA allows the negotiation of alternative encapsulations (e.g. Null Encapsulation) using the B-LLI negotiation procedures defined in ATM Signalling [UNI 3.0, UNI 3.1, UNI 4.0].

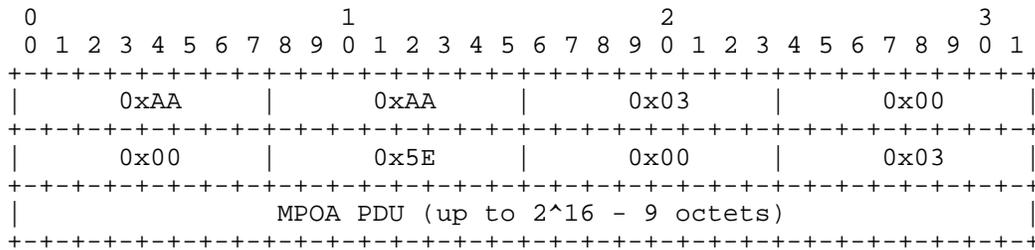


Figure 13 MPOA Control Frame Encapsulation

5.2 LANE TLVs

5.2.1 MPS Configuration TLVs

The following TLVs may be used to change the default values for MPS parameters.

TLV Name	Type	Length	Value
Keep-Alive Time	00-A0-3E-1D	2	MPS-p1 in seconds
Keep-Alive Lifetime	00-A0-3E-1E	2	MPS-p2 in seconds
Internetnetwork-layer Protocols	00-A0-3E-1F	8	MPS-p3 encoding: Control (1 octet): 0x00=DISABLE MPOA Resolution support for the protocol. 0x01=ENABLE MPOA Resolution support for the protocol. Short protocol (2 octets): (Encoded as specified in [NHRP]). Long protocol (5 octets): (Encoded as specified in [NHRP]). <i>Note: Multiple Internetnetwork-layer Protocol TLVs may be present.</i>
MPS Initial Retry Time	00-A0-3E-20	2	MPS-p4, in seconds.
MPS Retry Time Maximum	00-A0-3E-21	2	MPS-p5, in seconds.
MPS Give-up Time	00-A0-3E-22	2	MPS-p6 in seconds
Default Holding Time	00-A0-3E-23	2	MPS-p7, in seconds.

5.2.2 MPC Configuration TLVs

The following TLVs may be used to change the default values for MPC Parameters.

TLV Name	Type	Length	Value
SC-Setup Frame Count	00-A0-3E-24	2	MPC-p1
SC-Setup Frame Time	00-A0-3E-25	2	MPC-p2 in seconds
Flow-detection Protocols	00-A0-3E-26	8	<p>MPC-p3 encoding:</p> <p>Control (1 octet):</p> <p style="padding-left: 40px;">0x00=DISABLE flow detection for the protocol</p> <p style="padding-left: 40px;">0x01=ENABLE flow detection for the protocol.</p> <p>Short protocol (2 octets):</p> <p style="padding-left: 40px;">(Encoded as specified in [NHRP]).</p> <p>Long protocol (5 octets):</p> <p style="padding-left: 40px;">(Encoded as specified in [NHRP]).</p> <p><i>Note: Multiple Flow-detection Protocols TLVs may be present.</i></p>
MPC Initial Retry Time	00-A0-3E-27	2	MPC-p4, in seconds.
MPC Retry Time Maximum	00-A0-3E-28	2	MPC-p5, in seconds.
Hold Down Time	00-A0-3E-29	2	MPC-p6, in seconds.

5.2.3 Device Type TLV

The MPOA Device Type TLV contains two to five sub-fields, depending on the device, within the value field, as shown below. One sub-field identifies the type of MPOA device (server or client). The other sub-field specifies the ATM address of the MPOA device using UNI 4.0 encoding.

TLV Name	Type	Length	Value										
MPOA Device Type	00-A0-3E-2A	N	<p>MPOA DEVICE TYPE (1 octet)</p> <p>0 == Non-MPOA device 1 == MPOA Server 2 == MPOA Client 3 == MPS and MPC 4-255 undefined (reserved for future use)</p> <p>Number of MPS MAC Addresses (1 octet)</p> <p>This field is always present.</p> <p>Usage depends on device type, as follows:</p> <table border="1"> <thead> <tr> <th>device type</th> <th>use</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>must be zero (i)</td> </tr> <tr> <td>1</td> <td>may be zero or non-zero (ii)</td> </tr> <tr> <td>2</td> <td>must be zero (iii)</td> </tr> <tr> <td>3</td> <td>must be non-zero (iv)</td> </tr> </tbody> </table> <p>MPS Control ATM ADDRESS (20 octets)</p> <p>Present only for device types 1 and 3. (Private ATM address format. See UNI 4.0 sec. 3.0 for encoding)</p> <p>MPC Control ATM ADDRESS (20 octets)</p> <p>Present only for device types 2 and 3. (Private ATM address format. See UNI 4.0 sec. 3.0 for encoding)</p> <p>MPS MAC Addresses (Variable length)</p>	device type	use	0	must be zero (i)	1	may be zero or non-zero (ii)	2	must be zero (iii)	3	must be non-zero (iv)
device type	use												
0	must be zero (i)												
1	may be zero or non-zero (ii)												
2	must be zero (iii)												
3	must be non-zero (iv)												

- (i) all MAC addresses served by the ATM address are non-MPOA MAC addresses.
- (ii) if zero, then all MAC addresses served by the ATM address are MPS MAC addresses; if non-zero all MAC addresses served by the ATM address are non-MPOA MAC addresses except for those enumerated in the MAC address list, which are MPS MAC addresses.
- (iii) all MAC addresses served by the ATM address are MPC MAC addresses.
- (iv) all MAC addresses served by the ATM address are MPC MAC addresses except for those enumerated in the MAC address list, which are MPS MAC addresses.

5.3 Frame Formats

The encapsulation used for MPOA control messages is the same as is defined for NHRP control messages in [NHRP].

MPOA specifies the following control messages:

1. MPOA Resolution Request
2. MPOA Resolution Reply
3. MPOA Cache Imposition Request
4. MPOA Cache Imposition Reply
5. MPOA Egress Cache Purge Request

- 6. MPOA Egress Cache Purge Reply
- 7. MPOA Keep-Alive
- 8. MPOA Trigger

These messages reuse the NHRP packet formats. The NHRP packet type values 0x80 - 0x100 are administered by IANA. MPOA control messages use the values 0x80-0x87.

MPOA also uses the following NHRP control messages between MPCs and MPSs:

- 1. NHRP Purge Request
- 2. NHRP Purge Reply

5.3.1 MPOA CIE Codes

The following codes are defined for use in MPOA messages (in addition to those defined in NHRP). These codes may be used as deemed appropriate by MPOA components.

Table 5 MPOA CIE Codes

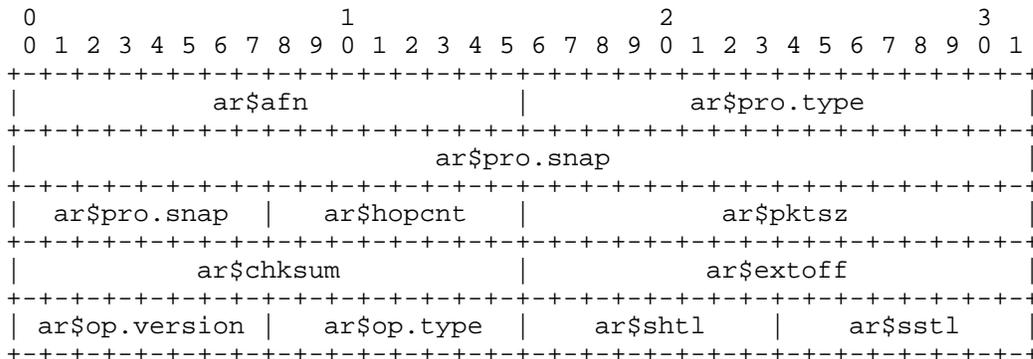
Code	Meaning
0x00	Success
0x81	Insufficient resources to accept egress cache entry.
0x82	Insufficient resources to accept shortcut
0x83	Insufficient resources to accept either shortcut or egress cache entry.
0x84	Unsupported Internetwork Layer protocol
0x85	Unsupported MAC layer encapsulation
0x86	Not an MPC
0x87	Not an MPS
0x88	Unspecified/other

5.3.2 Control Message Format

Each MPOA control message is conveyed using the NHRP packet format. NHRP Extensions may be included in each MPOA control message to convey additional information.

5.3.2.1 Fixed Header

Each MPOA control message has the same Fixed Header as an NHRP packet.



ar\$op.version is set to 0x01 (NHRP).

Packet type values (ar\$op.type) are assigned for MPOA control messages as follows.

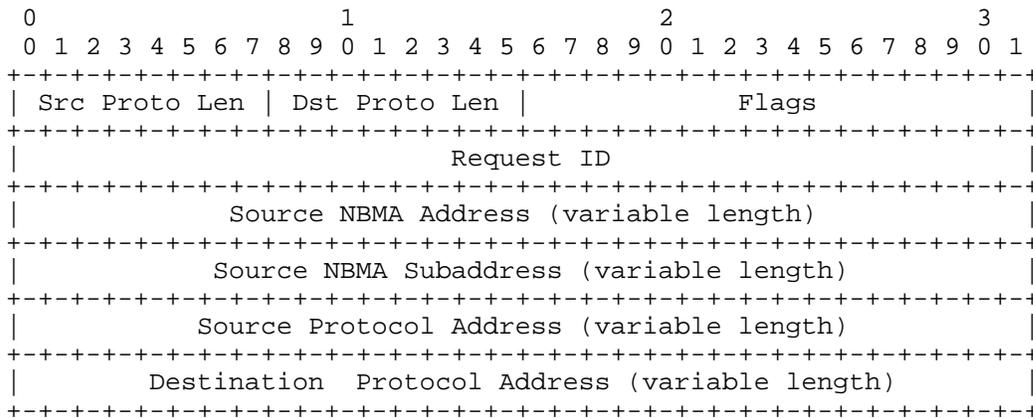
op.type	MPOA Control Message
0x80	MPOA Cache Imposition Request
0x81	MPOA Cache Imposition Reply
0x82	MPOA Egress Cache Purge Request
0x83	MPOA Egress Cache Purge Reply
0x84	MPOA Keep-Alive
0x85	MPOA Trigger
0x86	MPOA Resolution Request
0x87	MPOA Resolution Reply
0x88	MPOA Error Indication

Other fields in the Fixed Header must be set in conformance with NHRP.

5.3.2.2 Common Header

Each MPOA control message has the same Common Header as an NHRP packet.

The Common Header is as follows:



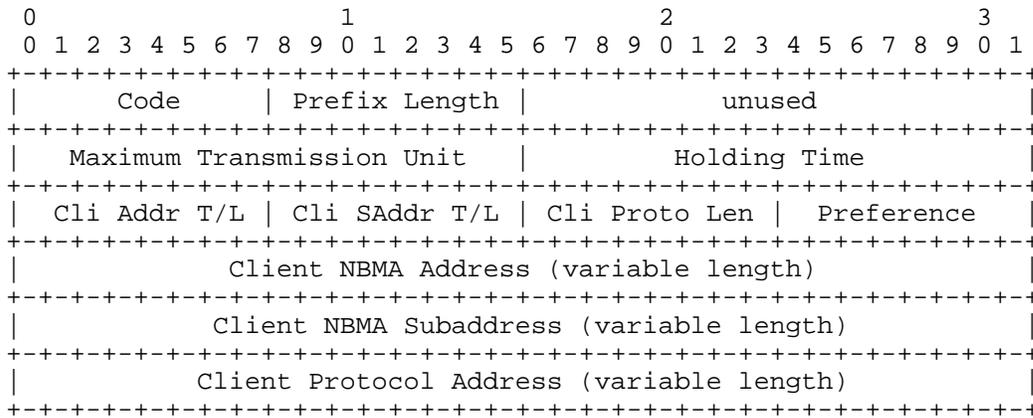
The Common Header specifies the sender’s NBMA (ATM) and internetwork layer address and the receiver’s internetwork layer address. Some exceptions exist in the context of certain MPOA control messages as described in the section for each message.

For consistency, all fields of the Common Header should be filled in as specified by NHRP, even though in some cases the receiving station may ignore some of them.

5.3.2.3 Client Information Element

MPOA control messages may have the same Client Information Elements as an NHRP packet.

The CIE has the following format:



The usage of the CIE for each message-specific part is described in the section for each message.

5.3.2.4 Extensions

MPOA control messages may have the same Extensions as an NHRP packet, such as Route Record, NHRP Authentication and Vendor Private Extensions.

All MPOA Extensions, summarized in the following table, use the NHRP Extension format.

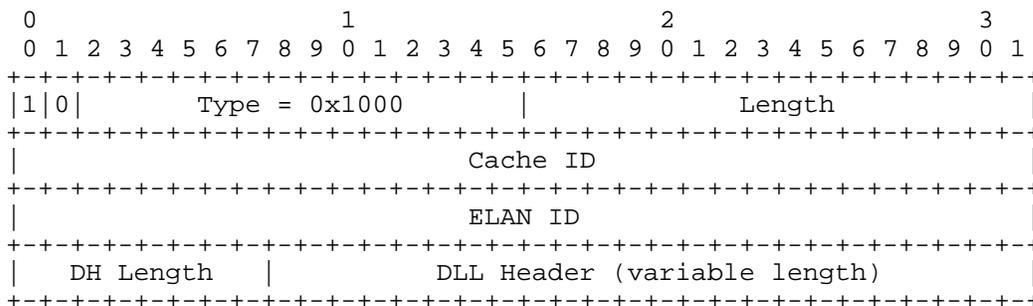
Table 6 MPOA Extensions

Type	MPOA Extension
0x1000	MPOA DLL Header Extension
0x1001	MPOA Egress Cache Tag Extension
0x1002	MPOA ATM Service Category Extension
0x1003	MPOA Keep-Alive Lifetime Extension
0x1004	MPOA Hop Count Extension
0x1005	MPOA Original Error Code Extension
0x1006	MPOA Authentication Extension

5.3.2.4.1 MPOA DLL Header Extension

The MPOA DLL Header Extension is used to convey Data-Link Layer Header information [including MAC destination address, MAC Source Address, Route Information Field (in the case of 802.5 Token Ring), Ethernet Type (in the case of Ethernet), or LLC Header (in the case of IEEE 802)].

The MPOA DLL Header Extension format is as follows:



Cache ID specifies the egress MPS's ID for an egress cache entry. A value of zero for the cache ID is not allowed.

ELAN ID specifies a LANE ELAN ID.

DH Length specifies the length of the DLL header.

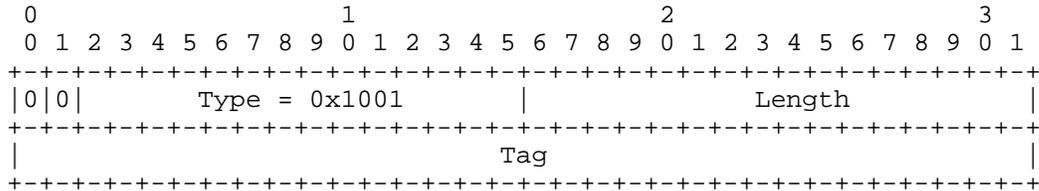
DLL header is used to specify Data-Link Layer Header information.

The specific usage of this Extension is described in the appropriate section for each message.

5.3.2.4.2 MPOA Egress Cache Tag Extension

The MPOA Egress Cache Tag Extension is used to convey egress cache tags.

The MPOA Egress Cache Tag Extension format is as follows:



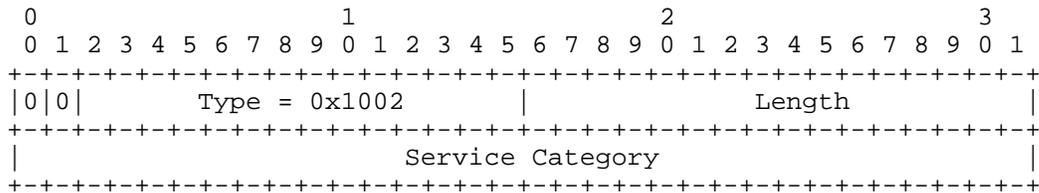
The tag is a 32 bit value chosen by the egress MPC. A value of zero for the tag is not allowed

The specific usage of this Extension is described in the appropriate section for each message.

5.3.2.4.3 MPOA ATM Service Category Extension

The MPOA ATM Service Category Extension is used to convey the set of Service Categories supported by the sender.

The MPOA ATM Service Category Extension format is as follows:



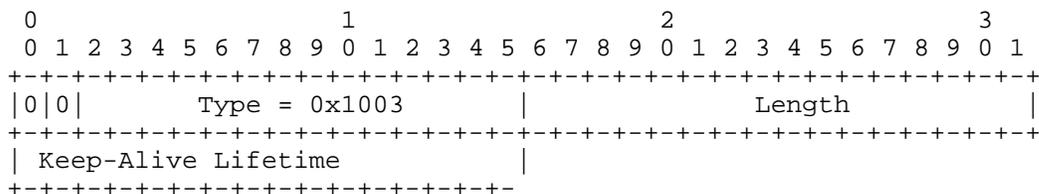
Service Category has precisely the same syntax and semantics defined in the corresponding LANE TLV [LANE].

The specific usage of this Extension is described in the appropriate section for each message.

5.3.2.4.4 MPOA Keep-Alive Lifetime Extension

The MPOA Keep-Alive Lifetime is used to convey the duration of time that a Keep-Alive message may be considered valid.

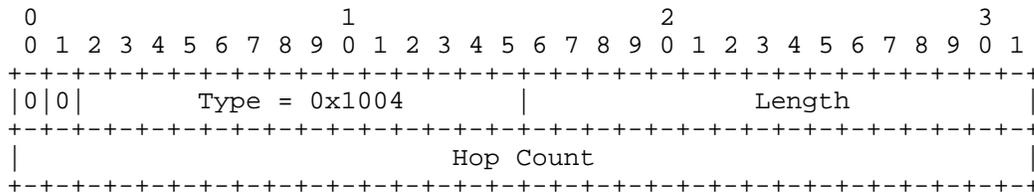
The MPOA Keep-Alive Lifetime Extension format is as follows:



5.3.2.4.5 MPOA Hop Count Extension

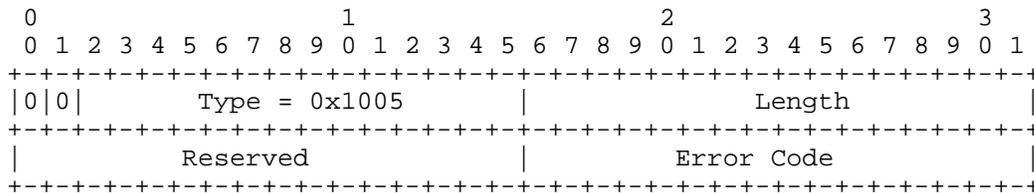
This extension is used to convey the hop count limit for forwarding internetworking packets. It is used for internetworking protocols that use a hop count field that counts up (e.g. the transport control field in IPX) and not down (e.g. the TTL field in IP). In order to be forwarded over a shortcut VCC the internetworking layer packet must contain a hop count that is less than the value specified in the hop count extension.

The MPOA Hop Count Extension format is as follows:



5.3.2.4.6 MPOA Original Error Code Extension

This extension may be included in any MPOA message. Its purpose is to preserve the original MPOA error code through conversions to and from NHRP messages. It may only be included in a Reply if the extension was included in a Request. An ingress MPC may include this extension with a null value in an MPOA Resolution Request, to allow its use in an MPOA Cache Imposition Reply. It may also be included in an MPOA Egress Cache Purge Request message so that it may be included with the subsequent NHRP Purges that the egress MPS may generate.



There are no MPOA-specific error codes currently defined.

5.3.2.4.7 MPOA Authentication Extension

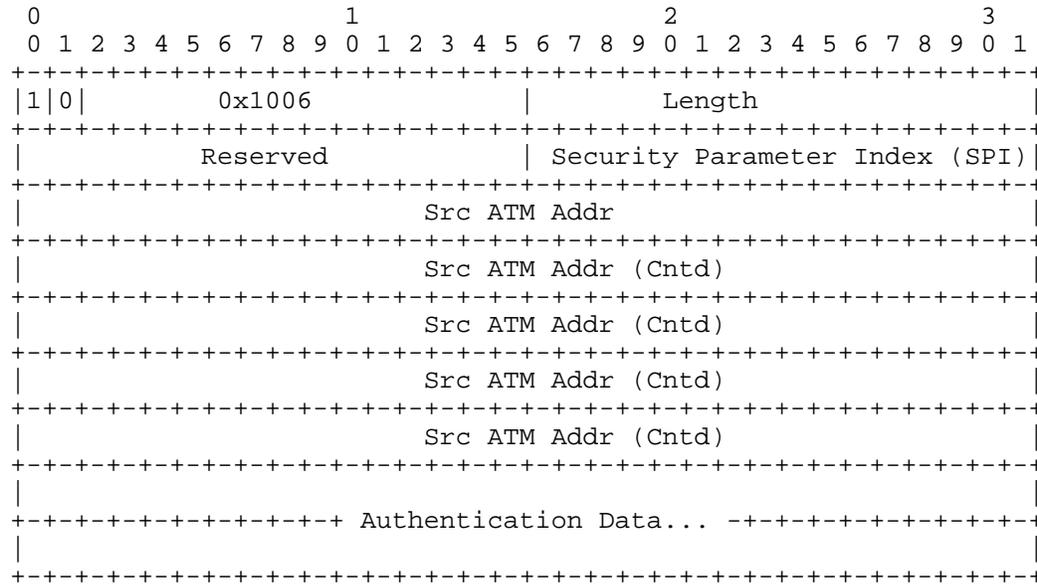
The MPOA Authentication Extension is carried in MPOA packets to convey authentication information between an MPC and an MPS. The MPOA Authentication Extension may be included in any message of MPOA used between MPC and MPS in either direction (i.e. from MPC to MPS or from MPS to MPC).

Whether an MPOA Authentication Extension is included in a message is a local matter, and the Extension is only used locally between MPC and MPS.

It should be noted that the MPOA Authentication Extension uses a type code different from the type code of the NHRP Authentication Extension. Authenticated exchanges between MPCs and MPSs must use the MPOA Authentication extension.

If a received packet fails the authentication test, the packet is either silently dropped or an Error Indication of type Authentication Failure is sent. Note that one possible authentication failure is the absence of an MPOA Authentication Extension.

The MPOA Authentication Extension format is as follows: :



Length

The length in octets of the remainder of the extension starting immediately after the length field.

Security Parameter Index (SPI)

Can be thought of as an index into a table that resides in both the MPC and MPS. This table contains the keys and other information such as the security algorithm. The MPC and MPS populate this table either offline using manual keying or online using a key management protocol.

Src ATM Addr

The MPOA Control ATM address of the source MPOA device. Between MPS/NHSs, NHRP Authentication is used.

The tuple <spi, src addr> uniquely identifies the key and other parameters that are used in authentication.

The length of the authentication data field is dependent on the security algorithm used.

The data field contains the keyed hash calculated over the entire MPOA payload. The authentication data field is zeroed out before the hash is calculated.

5.3.3 MPOA Resolution Request Format

An MPOA Resolution Request is sent from an ingress MPC to an ingress MPS to request the egress ATM address corresponding to an internetwork layer destination address. Upon receipt of an MPOA Resolution Request, an ingress MPS must send a new NHRP Resolution Request towards the egress MPS.

Fixed Header

ar\$op.type is set to 0x86 (MPOA Resolution Request).

Common Header

The Common Header is coded as follows:

Field	Usage
Flags	Unused.
Source NBMA Address	The ATM address of the ingress MPC from which internetwork layer datagrams will be sent.
Source NBMA Subaddress	The ATM Subaddress of the ingress MPC from which internetwork layer datagrams will be sent.
Source Protocol Address	Optional. Source Protocol address of the MPC if used. If not used, Src Proto Len field must be set to zero and no storage is allocated for the Source Protocol address..
Destination Protocol Address	The internetwork layer address of the final destination to which the internetwork layer datagrams will be sent.

Other fields in the Common Header must be set in conformance with NHRP.

Client Information Element

One CIE may be added in conformance with NHRP.

Field	Usage
Prefix Length	Widest acceptable prefix length.
MTU	Unused and must be set to zero (0).

Extensions

An MPOA Egress Cache Tag Extension must be added as follows:

Field	Usage
Type	This field must be set to 0x1001 for an MPOA Egress Cache Tag Extension.
Length	The length is set to zero (0) and no storage is allocated for the tag.

An MPOA ATM Service Category Extension should be added as follows:

Field	Usage
Type	This field must be set to 0x1002 for an MPOA ATM Service Category Extension.
Length	This field is coded as specified by NHRP.
Service Category	The Service Category is set to indicate the Service Categories supported by the ingress MPC.

5.3.4 MPOA Resolution Reply Format

An MPOA Resolution Reply is sent from an ingress MPS to an ingress MPC in reply to a corresponding MPOA Resolution Request upon receiving an NHRP Resolution Reply from the egress MPS.

Fixed Header

ar\$op.type is set to 0x87 (MPOA Resolution Reply).

Common Header

The Common Header is coded as follows:

Field	Usage
Flags	Unused.
Source NBMA Address	The ATM address of the ingress MPC from which internetwork layer datagrams will be sent.
Source NBMA Subaddress	The ATM Subaddress of the ingress MPC from which internetwork layer datagrams will be sent.
Source Protocol Address	Copied from corresponding MPOA Resolution Request.
Destination Protocol Address	The internetwork layer address of the final destination to which the internetwork layer datagrams will be sent.

Other fields in the Common Header must be set in conformance with NHRP.

Client Information Element

A CIE must always be present in the MPOA resolution reply.

For a normal resolution reply, all CIEs must be copied from the corresponding NHRP Resolution Reply.

To report an error condition the ingress MPS must include a CIE as follows:

Field	Usage
Code	Selected CIE Code
Prefix Length	Unused.
Maximum Transmission Unit	Unused.
Holding Time	Unused.
Cli Addr T/L	This field must be set to zero and no storage is allocated for the Client NBMA Address.
Cli SAddr T/L	This field must be set to zero and no storage is allocated for the Client NBMA Subaddress.
Cli Proto Len	This field must be set to zero and no storage is allocated for the Client Protocol Address.

Extensions

All Extensions must be copied from the corresponding NHRP Resolution Reply, except for those extensions the ingress MPS added to the initial request.

5.3.5 MPOA Cache Imposition Request Format

An MPOA Cache Imposition Request is sent from an egress MPS to an egress MPC to impose an egress cache entry upon receipt of an NHRP Resolution Request from the ingress MPS.

Fixed Header

ar\$op.type is set to 0x80 (MPOA Cache Imposition Request).

Common Header

The Common Header is coded as follows:

Field	Usage
Flags	Unused
Request ID	This field is a Request ID for the imposition transaction. The MPS should assign a Request ID for the MPOA Cache Imposition Request that is unique over all unacknowledged impositions.
Source NBMA Address	This field must be copied from the NHRP Resolution Request. This is the ATM address of the ingress MPC (or NHC) from which internetwork layer datagrams will be sent.
Source NBMA Subaddress	This field must be copied from the NHRP Resolution Request. This is the ATM Subaddress of the ingress MPC (or NHC) from which internetwork layer datagrams will be sent.
Source Protocol Address	This field is set to the internetwork layer address of the egress MPS.
Destination Protocol Address	This field must be copied from the NHRP Resolution Request. This address refers to the internetwork layer address of the final destination to which the internetwork layer datagrams will be sent.

Other fields in the Common Header must be set in conformance with NHRP.

Client Information Element

The CIE is coded as follows:

Field	Usage
Code	Unused.
Prefix Length	Prefix length in bits for the Destination Protocol Address, as known to the egress MPS.
Maximum Transmission Unit	The egress MPS must set this field based on either local information, or copy it from the NHRP Resolution Request.
Holding Time	The number of seconds for which this entry should be considered to be valid in the egress cache. The value given here must be at least twice as large as the one that will be returned in the corresponding NHRP Resolution Reply.
Cli Addr T/L	This field must be set to zero and no storage is allocated for the Client NBMA Address.
Cli Saddr T/L	This field must be set to zero and no storage is allocated for the Client NBMA Subaddress.
Cli Proto Len	This field must be set to zero and no storage is allocated for the Client Protocol Address at all.
Preference	Unused.

Extensions

All NHRP Extensions must be copied from the original NHRP Resolution Request.

If the holding time in the CIE is non-zero, an MPOA DLL Header Extension must be included as follows:

Field	Usage
Type	This field must be set to 0x1000 for an MPOA DLL Header Extension.
Length	This field is coded as specified by NHRP.
Cache ID	This field is used to convey cache ID to be set in the egress cache. Cache ID value must be selected by the egress MPS so that a different ID is assigned for each egress cache entry.
ELAN ID	This field is set to the LANE ELAN Id.
DH Length	The length of the DLL Header.
DLL Header	The DLL header to be used for encapsulating internetwork layer datagrams when the egress MPC receives the datagrams from the shortcut before sending them to the higher layers.

The MPS sends the egress MPC the DLL header that the egress router would use to transmit frames along the default routed path via LANE to the destination specified in the NHRP common header.

For a given LAN type, the DLL headers are self-describing. It is the responsibility of the egress MPC to parse the DLL header provided by the MPS to determine whether the given encapsulation is supported for the given protocol. If the MPC does not support the encapsulation or protocol provided in the MPOA Cache Imposition Request, the MPC must return a status value in the MPOA Cache Imposition Reply indicating that either the DLL encapsulation or protocol is not supported.

For DLL encapsulations that contain a length field, such as 802.3 LLC SNAP, the length field in the DLL header section of the MPOA Cache Imposition Request must be filled in with the correct length for an empty frame. In the 802.3 LLC SNAP case, for example, the length field is set to 8.

The MPOA Imposition Request message is also used to purge egress cache entries in the egress MPC. In this case the source NBMA Address field must be NULL, the holding time field must be set to zero and MPOA DLL header extension is optional.

5.3.6 MPOA Cache Imposition Reply Format

An MPOA Cache Imposition Reply is sent from an egress MPC to an egress MPS in reply to an MPOA Cache Imposition Request.

Fixed Header

ar\$op.type is set to 0x81 (MPOA Cache Imposition Reply).

Common Header

The Common Header must be copied from the corresponding MPOA Cache Imposition Request.

Client Information Element

CIEs are coded as follows:

Field	Usage
Code	Selected CIE Code
Prefix Length	Actual prefix length imposed in bits for the Client Protocol Address. The prefix length may be set to indicate a host route.
Maximum Transmission Unit	The MTU size should be set to the maximum value allowed by the egress MPC. This value must be non-zero.
Holding Time	Unused.
Cli Addr T/L	This field must be set to zero (and no Client ATM Address included) unless the status is Success.
Cli Saddr T/L	This field must be set to zero (and no Client ATM Subaddress included) unless the status is Success.
Cli Proto Len	This field must be set to zero and no storage is allocated for the Client Protocol Address at all.
Client NBMA Address	ATM address of the egress MPC that will receive internetwork layer datagrams via the shortcut.
Client NBMA Subaddress	ATM subaddress of the egress MPC that will receive internetwork layer datagrams via the shortcut (if any).

Extensions

All Extensions must be copied from the corresponding MPOA Cache Imposition Request.

If an MPOA Egress Cache Tag Extension is included, it must be set as follows:

Field	Usage
Length	If a valid tag exists, the length is set to four (4).
Tag	Set to the value of the tag chosen by the egress MPC.

If an MPOA Service Category Extension is included, it must be set as follows:

Field	Usage
Service Category	The Service Category is set to indicate the Service Categories supported by the egress MPC.

5.3.7 MPOA Egress Cache Purge Request Format

An MPOA Egress Cache Purge Request is sent from an egress MPC to an egress MPS to purge an egress cache entry.

Fixed Header

ar\$op.type is 0x82 (MPOA Egress Cache Purge Request).

Common Header

The Common Header is coded as follows:

Field	Usage
Flags	The Flags field is coded as follows: <pre> 0 1 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+ N +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+ N: No-Reply Flag (recommended to be set)</pre>
Request ID	The semantics of this field are the same as that of NHRP. This value must be selected by the egress MPC, so that it may recognize the corresponding MPOA Egress Cache Purge Reply.
Source NBMA Address	Data ATM Address of the egress MPC.
Source NBMA Subaddress	Data ATM Subaddress of the egress MPC (if any).
Source Protocol Address	Internetwork layer address of the MPC (if any). If the MPC has no internetwork layer address, Src Proto Len must be set to zero and no storage is allocated for the Source Protocol Address.
Destination Protocol Address	Internetwork layer address of the egress MPS.

Other fields in the Common Header must be set in conformance with NHRP.

Client Information Element

CIEs are coded as follows:

Field	Usage
Prefix Length	Destination Prefix Length.
Client Protocol Address	Destination Protocol Address (to purge)
Client NBMA Address	Egress MPC Data ATM Address
Client NBMA Subaddress	Egress MPC Data subaddress (if any)

Extensions

An MPOA DLL Header Extension must be included as follows:

Field	Usage
Cache ID	This field is used to convey a cache ID of the egress cache entry being invalidated.
ELAN ID	Unused.
DH Length	This field must be set to zero and no storage is allocated for the DLL Header.

5.3.8 MPOA Egress Cache Purge Reply Format

An MPOA Egress Cache Purge Reply is sent from an egress MPS to an egress MPC in reply to an MPOA Egress Cache Purge Request.

An MPOA Egress Cache Purge Reply is formed from an MPOA Egress Cache Purge Request by changing the ar\$op.type to 0x83.

Fixed Header

ar\$op.type is set to 0x83 (MPOA Egress Cache Purge Reply).

Common Header

The Common Header must be copied from the corresponding MPOA Egress Cache Purge Request.

Client Information Element

All CIEs must be copied from the corresponding MPOA Egress Cache Purge Request.

Extensions

All Extensions must be copied from the corresponding MPOA Egress Cache Purge Request.

5.3.9 MPOA Keep-Alive Format

An MPOA Keep-Alive is periodically sent from an MPS to an MPC(s).

Fixed Header

ar\$op.type is set to 0x84 (MPOA Keep-Alive).

Common Header

The Common Header is coded as follows:

Field	Usage
Flags	Unused.
Request ID	The sequence number of the Keep-Alive Message. This value must be set to zero on the first transmission and must be incremented by at least one each time the MPS sends this message to a given MPC.
Source NBMA Address	Control ATM Address of the MPS.
Source NBMA Subaddress	NULL
Source Protocol Address	NULL
Destination Protocol Address	NULL

Client Information Element

There are no CIEs for this message.

Extensions

An MPOA Keep-Alive Lifetime Extension must be added as follows:

Field	Usage
Type	This field must be set to 0x1003 for an MPOA Keep-Alive Lifetime Extension.
Length	Two (2) Octets
Keep-Alive Lifetime	Set to the duration of time that a Keep-Alive message may be considered valid

5.3.10 MPOA Trigger Format

An MPOA Trigger is sent from an ingress MPS to an ingress MPC to request the ingress MPC to issue MPOA Resolution Requests.

Fixed Header

ar\$op.type is set to 0x85 (MPOA Trigger).

Common Header

The Common Header is coded as follows:

Field	Usage
Flags	Unused.
Request ID	Unused (there is no reply/ACK packet for this message).
Source NBMA Address	Control ATM address of the ingress MPS.
Source NBMA Subaddress	NULL
Source Protocol Address	Internetwork layer address of the ingress MPS. This must be the internetwork layer address that would be returned in an NHRP Responder Address Extension or NHRP Forward/Reverse Transit Record Extension included by that MPS (if such Extensions are included) in an NHRP Resolution Reply.
Destination Protocol Address	Destination Protocol Address (to trigger)

Client Information Element

One CIE may be added in conformance with NHRP.

Field	Usage
Prefix Length	Widest Acceptable prefix length.
MTU	Unused and must be set to zero (0).

Extensions

No extensions are required.

5.3.11 NHRP Purge When Used on the Data Plane

An NHRP Purge message may be sent on the data plane by an egress MPC to an ingress MPC or NHC to purge ingress cache entries.

Fixed Header

ar\$op.type is set to 0x05 (NHRP Purge Request).

Common Header

The Common Header is coded as follows:

Field	Usage
Flags	The Flags field is coded as follows: <pre> 0 1 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 +-----+ N unused +-----+ N: No-Reply Flag The N bit must be set to one.</pre>
Request ID	Unused. Must be set to zero (0).
Source NBMA Address	Egress MPC data ATM address.
Source NBMA Subaddress	Egress MPC data ATM subaddress (if any).
Source Protocol Address	1. Set to the protocol address of the egress MPS (if known) or NULL (Src Proto Len=0 and no storage is allocated for the Source Protocol Address).
Destination Protocol Address	NULL

Other fields in the Common Header must be set in conformance with NHRP.

Client Information Element

The purge request includes one or more CIEs as follows:

Field	Usage
Code	Set to 0x00.
Prefix Length	Prefix Length in bits for the Client Protocol Address(es) being purged. At least the ingress cache entries associated with the shortcut over which the MPOA Data Plane Purge is received are purged.
Maximum Transmission Unit	Unused.
Holding Time	Unused.
Cli Addr T/L	This field must be set to zero and no storage is allocated for the Client NBMA address at all.
Cli SAddr T/L	This field must be set to zero and no storage is allocated for the Client NBMA Subaddress at all.
Cli Proto Len	The length in octets of the Client Protocol Address.
Client NBMA Address	Egress MPC data ATM address.
Client Protocol Address	The internetwork layer address that is being purged from the ingress MPC's cache.

Extensions

NHRP Authentication and Vendor Private Extensions may be added as desired.

5.3.12 NHRP Ingress Cache Purge Request Format

An NHRP Ingress Cache Purge Request is sent from an ingress MPS to an ingress MPC to purge an ingress cache entry.

Fixed Header

ar\$op.type is 0x05 (NHRP Purge Request).

Common Header

The Common Header is coded as follows:

Field	Usage
Flags	<p>The Flags field is coded as follows:</p> <pre> 0 1 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 +-----+-----+-----+-----+ N unused +-----+-----+-----+-----+ N: No-Reply Flag (No-Reply Flag is recommended to be set, i.e. no reply expected) </pre>
Request ID	The semantics of this field are specified in [NHRP]. This value must be selected by the ingress MPS, so that it may recognize the corresponding NHRP Ingress Cache Purge Reply. The same Request ID must be used by the Ingress MPS for the retry procedure.
Source NBMA Address	Data ATM address of the ingress MPS.
Source NBMA Subaddress	Data ATM subaddress of the ingress MPS (if any).
Source Protocol Address	Internetwork layer address of the ingress MPS (Note).
Destination Protocol Address	Internetwork layer address of the ingress MPC or NULL (Note).

Other fields in the Common Header must be set in conformance with NHRP.

Note: See also Appendix VII, section 2, for further information.

Client Information Element

The NHRP ingress cache purge request includes one or more CIEs as follows:

Field	Usage
Code	Set to 0x00 in NHRP purge requests.
Prefix Length	Destination prefix length in bits for the Client Protocol Address(es) being purged. The ingress MPS takes this value from the NHRP purge request which it had received from its peer-NHS.
Maximum Transmission Unit	Unused and coded as zero.
Holding Time	Unused and coded as zero.
Cli Proto Len	Length of the purged client's protocol address
Preference	Unused and coded as zero.
Client Protocol Address	The internetwork layer address that is being purged from the ingress MPC's cache.

Other fields in the CIE must be set as received in the NHRP purge request from the peer-NHS.

Extensions

NHRP extensions may be added as desired, particularly if received in the NHRP purge request from the peer-NHS. An MPOA DLL header extension shall not be added.

5.3.13 NHRP Ingress Cache Purge Reply Format

An NHRP Ingress Cache Purge Reply is sent from an ingress MPC to an ingress MPS in reply to an NHRP Ingress Cache Purge Request, depending on the value of the received no-reply flag.

An NHRP Ingress Cache Purge Reply is formed from an NHRP Ingress Cache Purge Request by changing the ar\$op.type to 0x06.

Fixed Header

ar\$op.type is set to 0x06 (NHRP Purge Reply).

Common Header

The Common Header must be copied from the corresponding NHRP Ingress Cache Purge Request.

Client Information Element

All CIEs must be copied from the corresponding NHRP Ingress Cache Purge Request.

Extensions

All Extensions must be copied from the corresponding NHRP Ingress Cache Purge Request.

5.3.14 MPOA Error Indication Format

The MPOA Error Indication is used to convey error indications to the sender of an MPOA packet.

Fixed Header

ar\$op.type is set to 0x88 (MPOA Error Indication).

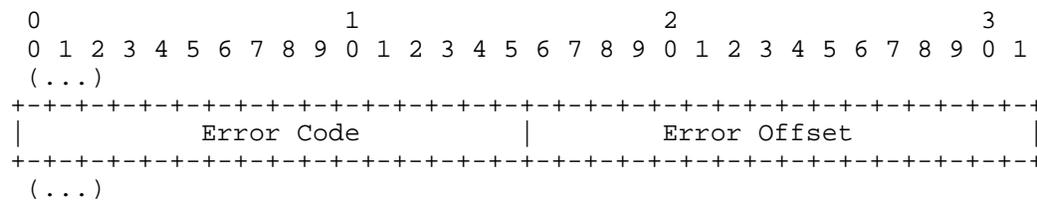
Common Header

The Common Header is coded as follows:

Field	Usage
Flags	Unused
Request ID	Coded as concatenation of error code and error offset as shown below.
Source NBMA Address	ATM Address of the MPOA component which observed the error.
Source NBMA Subaddress	ATM Subaddress of the MPOA component which observed the error (if any).
Source Protocol Address	Optional. Protocol address of the MPOA component (if used) which issues the MPOA error indication. If not used, the Src Proto Len field must be set to zero and no storage is allocated for the Source Protocol address.
Destination Protocol Address	Optional. Protocol address of the MPOA component (if used) which sent the MPOA packet which was found to be in error. If not used, the Dest Proto Len field must be set to zero and no storage is allocated for the Destination Protocol address.

Other fields in the Common Header must be set in conformance with NHRP.

The Request ID field of the Common Header is used to indicate the error code and error offset as follows:



The Error Code indicates the type of error detected. The Error Code is chosen from the following list:

- 0x0000 - 0x000A reserved for future use
- 0x000B Authentication Failure
- 0x000C - 0xFFFF reserved for future use

0x000B - Authentication Failure: This error code is returned if a received packet fails an authentication test.

The Error Offset indicates the error offset in octets into the original MPOA packet in which an error was detected. This offset is calculated starting from the MPOA Fixed Header.

An MPOA Error Indication shall never be generated in response to another MPOA Error Indication packet. When an MPOA Error Indication packet is generated, the offending MPOA packet shall be discarded. In no case shall more than one MPOA Error Indication be generated for a single MPOA packet.

Client Information element

There are no CIEs for this message.

Extensions

No extensions may be added to an MPOA Error Indication.

6. References

- [AIW] APPN Implementers' Workshop, Document ATM-09, HPR Extensions for ATM Networks.
- [IEEE 802.1d] ISO/IEC 10038; ANSI/IEEE Std. 802.1d, "Information Processing Systems - Local Area Networks - (MAC Bridges).
- [IPOA] M. Laubach / J. Halpern, RFC 2225, Classical IP and ARP over ATM, April 1998.
- [ISO 9646-1] ISO/IEC 9646-1:1994, Information technology - Open systems interconnection - Conformance testing methodology and framework - Part 1: General Concepts. (See also ITU Recommendation X.290(1995)).
- [ISO 9646-7] ISO/IEC 9646-7:1994, Information technology - Open systems interconnection - Conformance testing methodology and interconnection - Part 7: Implementation Conformance Statements - Requirements and Guidance on ICS and ICS Proformas.
- [LANE] Keene, J. (editor), "LAN Emulation Over ATM Version 2 - LUNI Specification," ATM Forum (AF-LANE-0084.000), 1997.
- [MARS] G.J. Armitage, RFC 2022, Support for Multicast over UNI 3.0/3.1 based ATM Networks, November 1996.
- [MPOA 1.0] A. Fredette (editor), "Multi-Protocol Over ATM, Version 1.0", ATM Forum (AF-MPOA-0087.000), July 1997
- [MPOA 1.0 MIB] J. Cucchiara (editor), "Multi-Protocol Over ATM Version 1.0 MIB", ATM Forum (AF-MPOA-0092.000), July 1998
- [MTU] R. Atkinson, RFC 1626, Default IP MTU for use over ATM AAL5, May 1994
- [NHRP] J. Luciani, et al., RFC 2332, NBMA Next Hop Resolution Protocol (NRHP), April 1998.
- [PATH MTU] J. Mogul and S. Deering, RFC 1191, Path MTU Discovery, November, 1990.
- [RFC 791] J. Postel, RFC 791, Internet Protocol, September 1, 1981.
- [RFC 1483] J. Heinanen, RFC 1483, Multiprotocol encapsulation over ATM Adaptation Layer 5.
- [RFC 1755] M. Perez, F. Liaw, A. Mankin, E. Hoffman, D. Grossman, A. Malis, RFC 1755 ATM Signaling support for IP over ATM.
- [ROUTER REQ] RFC 1812, Fred Baker, Requirements for IP Version 4 Routers, June 22, 1995.
- [RSVP] B. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, Internet Draft, draft-ietf-rsvp-spec-14.txt.
- [UNI 3.0] "ATM User-Network Interface (UNI) Specification, Version 3.0", The ATM Forum Technical Committee.
- [UNI 3.1] "User-Network Interface (UNI) Specification, Version 3.1", The ATM Forum Technical Committee.
- [UNI 4.0] "ATM User-Network Interface (UNI) Signaling Specification, Version 4.0", The ATM Forum Technical Committee.

Annex A. Protocol-Specific Considerations

[Normative]

Each internetwork layer protocol provides addressing and forwarding functions in a different way, and uses different encapsulations and different demultiplexing points (e.g. LSAPs, OUIs, PIDs, and Ethernet Types) on LANs. Many internetwork layer protocols even have multiple encapsulations on a single LAN. Because of these differences, MPOA components require internetwork layer protocol-specific knowledge to perform flow detection, address resolution, and shortcut data transformations. Flow detection and address resolution require at least a minimal understanding of internetwork layer addresses. Ingress MPC and egress MPC shortcut transformations must be defined on a protocol-specific basis. Each of these functions must be specified individually for each internetwork layer protocol supported.

A.1 IP Packet Handling in MPOA

This section describes the processing of IP packets in MPOA. Note that MPOA does not define the handling of IP Packets that are not sent over shortcuts.

A.1.1 Requirements

The MPOA System must support the IP version 4 Router Requirements [ROUTER REQ]. MPOA distributes this responsibility across MPOA components.

A.1.2 Encapsulation

There are three standard formats for IP packets carried over Ethernet and Token Ring (shown in Figure 14-Figure 16), and three standard formats for IP packets carried over an MPOA shortcut (shown in Figure 17-Figure 19).

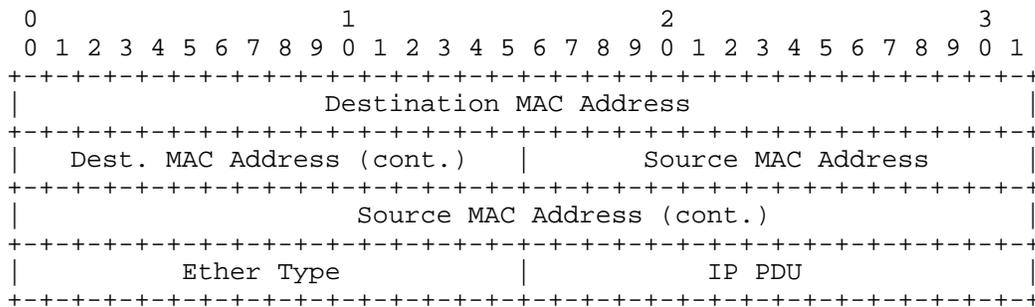


Figure 14 Ethernet IP Encapsulation

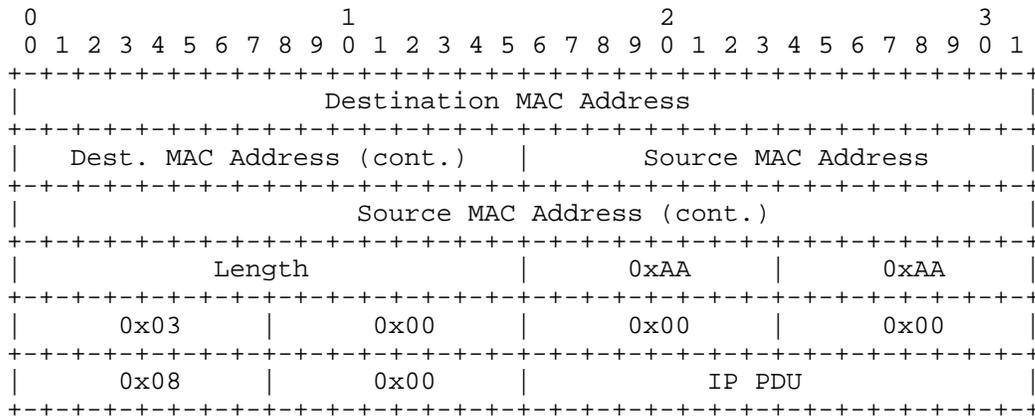


Figure 15 802.3 IP Encapsulation

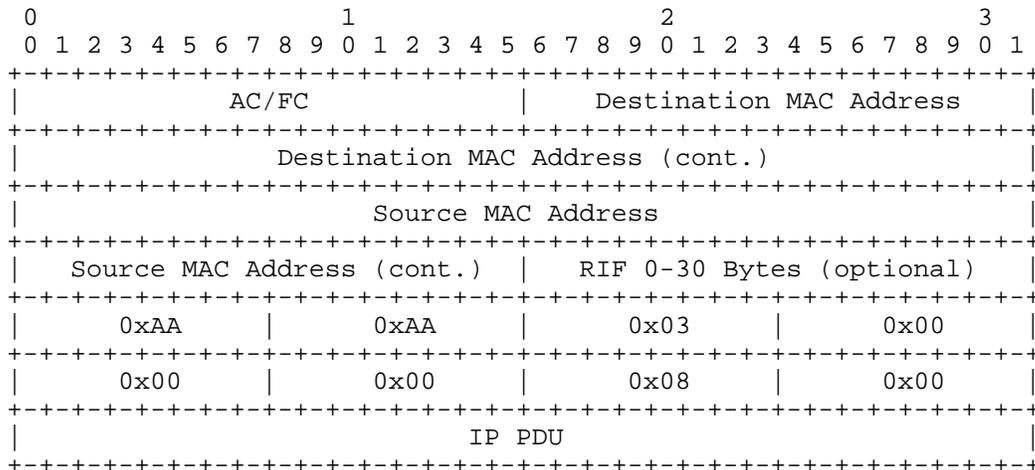


Figure 16 802.5 IP Encapsulation

The DLL header supplied in the egress cache entry consists of the entire media specific encapsulation through Ethernet Type. Note that the LECID is not included in this DLL header. Note that the 802.3 DLL header contains a length field that must be updated by the egress MPC for each packet received on a shortcut.

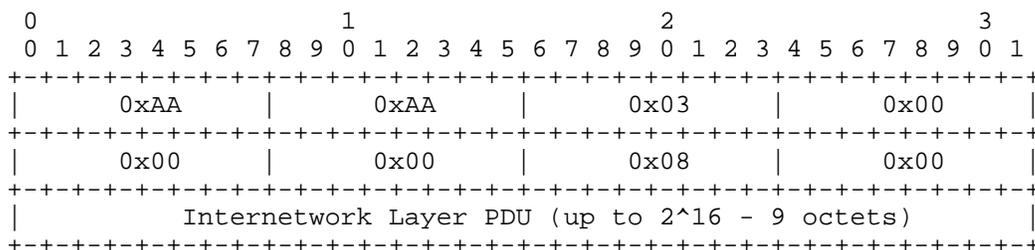


Figure 17 RFC 1483 LLC/SNAP Encapsulation for Routed IP PDUs

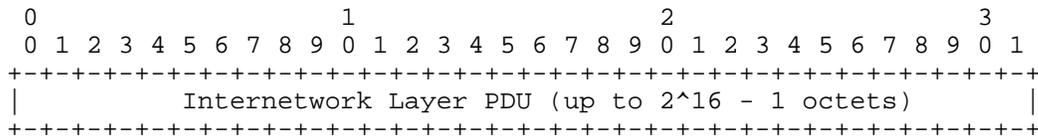


Figure 18 RFC 1483 "Null" Encapsulation for Routed IP PDUs

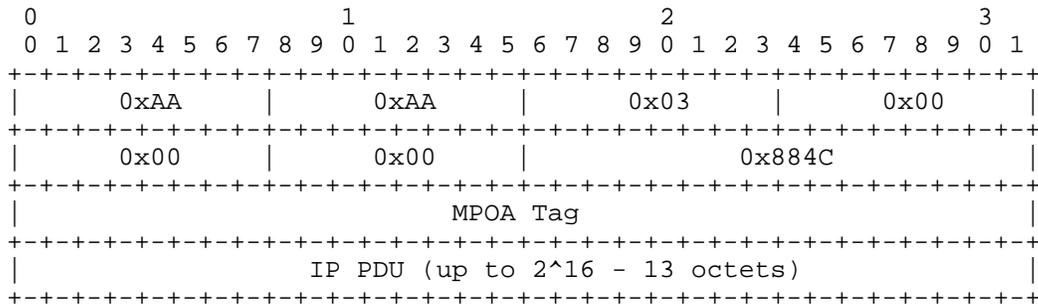


Figure 19 MPOA Tagged Encapsulation for IP

A.1.3 MPS Role

An MPS is co-located with a router and must process IP packets in conformance with [ROUTER REQ].

A.1.4 Ingress MPC Role

An MPC must perform basic IP forwarding. Additional router features may be implemented in an MPC, as described in the following subsections.

IP Options

There are some cases when the ingress MPC does not have all of the information that is required to meet the router requirements. For example, when the ingress MPC receives a Strict Source Route option in an IP packet, it does not have access to the routing table to determine whether the next hop given in the option is indeed the current next hop to the destination. Thus, the ingress MPC cannot determine on its own whether the packet is to be forwarded or dropped.

If the ingress MPC cannot correctly process an IP option, it must send the packet unmodified to the MPS via LANE. An ingress MPC may send all packets with IP options to the MPS. Of course, if an MPC has full support for an IP option, it should process the option on its own and send the packet over a shortcut.

TTL

When sending a packet via LANE, the MPC must not modify the TTL field in an IP packet.

When sending a packet via a shortcut, the ingress MPC must decrement the TTL by at least one in conformance with [Router Requirements]. There is no requirement to decrement the TTL by the actual number of router hops that are bypassed by the shortcut. If the TTL of the received packet is less than or equal to the value by which the MPC intends to decrement the TTL, the MPC must either send the packet unmodified to the MPS via LANE, or must discard the packet and send an ICMP Time Exceeded to the source of the packet.

Checksum

The IP checksum must be modified to reflect all changes to the IP header.

ICMP

The ingress MPC may encounter a variety of error conditions when forwarding packets and must ensure that the appropriate ICMP Error is generated. The ingress MPC must either send the packet unmodified to the MPS via LANE, or must send the applicable ICMP message.

An ingress MPC may generate the following ICMP messages:

- Destination Unreachable (Fragmentation needed but DF bit set)
- Time Exceeded (during transit)
- Parameter Problem

When sending the Destination Unreachable ICMP due to fragmentation, the Path MTU Discovery technique should be used as defined in [PATH MTU].

MTU

An ingress MPC may, but is not required to, fragment.

If the MTU returned in the MPOA Resolution Reply is smaller than the MTU on the inbound interface, the ingress MPC must make a decision on how to handle the shortcut and fragmentation. To avoid unnecessary packet mis-ordering, the ingress MPC must not set up the shortcut, and then send frames for a given flow on different paths based on whether fragmentation is required. The following are acceptable options:

1. Ingress MPC may establish the shortcut and fragment packets when necessary.
2. Ingress MPC may choose not to establish the shortcut.

A.1.5 Egress MPC Role

MTU

The egress MPC may advertise a large MTU and fragment the packets itself.

TTL

The egress MPC is not required to decrement TTL.

A.2 IPX Packet Handling in MPOA

A.2.1 Requirements

The MPOA System must support the IPX router requirements. MPOA distributes this responsibility across MPOA components.

A.2.2 Encapsulation

Four common IPX encapsulations are in use over Ethernet: Raw Ethernet (Novell proprietary), Ethernet_II, LLC, and LLC/SNAP. Other media types have their own associated encapsulations. Three IPX encapsulations, shown in Figure 20-Figure 22, may be used over a shortcut: Tagged, RFC 1483 Routed, and RFC 1483 NULL.

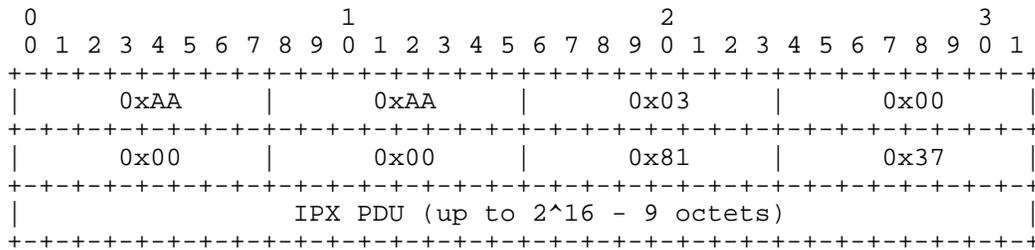


Figure 20 RFC 1483 LLC/SNAP Encapsulation for Routed IPX PDUs

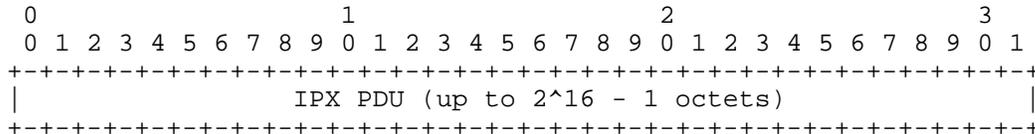


Figure 21 RFC 1483 "Null" Encapsulation for Routed IPX PDUs

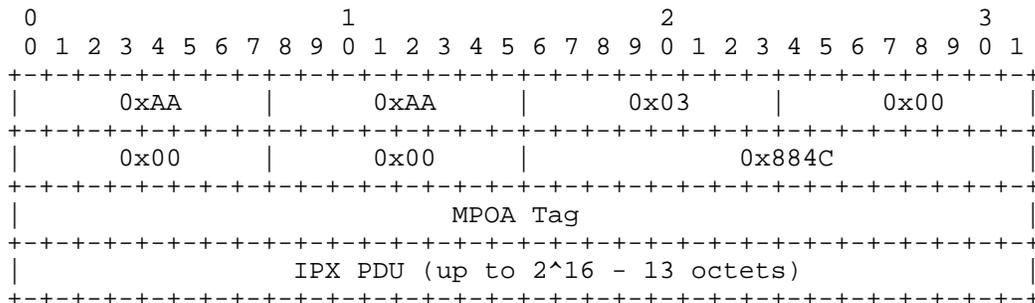


Figure 22 MPOA Tagged Encapsulation for IPX

A.2.3 MPS Role

An MPS is co-located with an IPX router and must process IPX packets in conformance with IPX router requirements.

The ingress MPS must include a hop count extension in MPOA Resolution Replies with a value as determined by the IPX routing protocol in use, (e.g. 16 for RIP).

A.2.4 Ingress MPC Role

IPX Options

There is no equivalent to the IP Options feature with IPX.

Transport Control

Unlike the IP TTL counts down, the IPX TC (Transport Control) field counts up. IPX hosts transmit packets with a TC of zero. When an upper bound is reached, the packet is discarded. An ingress MPC must increment the TC before sending an IPX packet on a shortcut.

Originally, the upper bound with the IPX-RIP routing protocol was 16. With NLSP, this number is configurable and may be as high as 128. Some other IPX routing protocols have an upper bound of 255. Packets that hit this limit may

be dropped or forwarded unmodified to the router to be dropped. The checksum field is not updated when the TC is modified, as the TC field is not included in the set of fields the checksum covers.

An ingress MPC must only forward an IPX packet over a shortcut if the transport control field in the packet has a value that is less than the hop count extension value in the associated MPOA Resolution Reply. If the value of the transport control field is greater than or equal to the hop count extension value, the MPC must send the packet unmodified to the MPS via LANE.

In certain cases, unicast IPX packets may have a Source Network Number of zero. When a Netware client first transmits a packet after booting, it will not know its own network number and will insert a zero in the Source Network Number field. After the initial SAP/RIP exchange it will know its own network number and should use this from that point on. However, some clients do not do this and continue to use a zero Source Network Number. IPX routers are required to accept these packets and insert the correct Source Network Number as they forward a packet. If a packet with a source network number of zero is received by an MPC, the MPC must send the packet unmodified to the MPS via LANE.

MTU

There is no fragmentation in IPX. Netware applications try different packet sizes until communication is achieved.

Encapsulation

The ingress MPC removes the LAN encapsulation and adds the shortcut encapsulation.

A.2.5 Egress MPC Role

Checksum

When forwarding an IPX packet from a network where checksums are in use to a network using the Novell proprietary Raw Ethernet encapsulation, it is necessary to set the checksum field to 0xFFFF. Packets using the Raw Ethernet encapsulation can only be distinguished from other encapsulations if the checksum field is 0xFFFF. This processing must be carried out by the egress MPC.

Encapsulation

The egress MPC removes the shortcut encapsulation and adds the LAN encapsulation.

The egress MPC does no further processing of IPX packets and simply delivers the packets to the higher layers.

Annex B. MPOA Request/Reply Packet Contents

[Normative]

B.1 Ingress MPC-Initiated MPOA Resolution

Ingress MPC-Initiated MPOA Resolution includes a request phase and a reply phase. The request phase proceeds from left to right as follows:

	Ingress MPC	Ingress MPS	Egress MPS	Egress MPC
Packet Type	MPOA Resolution Request	NHRP Resolution Request	MPOA Cache Imposition Request	
Request ID	Request ID 1	Request ID 2	Request ID 3	
Source Protocol Address	NULL or MPC Protocol Address	I-MPS Protocol Address	E-MPS Protocol Address	
Destination Protocol Address	Destination Protocol Address	Destination Protocol Address	Destination Protocol Address	
Source NBMA Address	I-MPC Data ATM Address	I-MPC Data ATM Address	I-MPC Data ATM Address	
Client Protocol Address (1)	NULL	NULL	NULL	
Prefix Length (1)	Widest Acceptable Prefix Length	Widest Acceptable Prefix Length	Requested Prefix Length	
Holding Time			$\geq 2 * \text{Holding Time}$	
Client NBMA Address (1)	NULL	NULL	NULL	
Extensions	Empty MPOA Egress Cache Tag Extension MPOA ATM Service Category Extension (1)	Received Extensions	Received Extensions MPOA DLL Header Extension (Cache ID, ELAN ID, DLL Header)	

The reply phase proceeds from right to left as follows:

	Ingress MPC	Ingress MPS	Egress MPS	Egress MPC
Packet Type		MPOA Resolution Reply	NHRP Resolution Reply	MPOA Cache Imposition Reply
Request ID		Request ID 1	Request ID 2	Request ID 3
Source Protocol Address		Restore to NULL or I-MPC address (from original MPOA Resolution Request)	I-MPS Protocol Address	E-MPS Protocol Address
Destination Protocol Address		Destination Protocol Address	Destination Protocol Address	Destination Protocol Address
Source NBMA Address		I-MPC Data ATM Address	I-MPC Data ATM Address	I-MPC Data ATM Address
Client Protocol		E-MPS Protocol Address	E-MPS Protocol Address	NULL

Address				
Prefix Length		Actual Prefix Length	Actual Prefix Length	Actual Prefix Length (2)
Holding Time		Holding Time	Holding Time	NULL
Client NBMA Address		E-MPC Data ATM Address	E-MPC Data ATM Address	E-MPC Data ATM Address
Extensions		Received Extensions	Received Extensions	Received Extensions

Legend:

NULL: zero length, no space allocated in packet

Notes:

(1) Optional

(2) An E-MPC can modify the Prefix Length to make it a host entry if a CIE was included in the request.

B.2 Egress MPC-Initiated Egress Cache Purge

Egress MPC-Initiated Egress Cache Purge includes a request phase and a reply phase. The request phase proceeds from right to left as follows:

	Ingress MPC	Ingress MPS	Egress MPS	Egress MPC
Packet Type		NHRP Purge Request	NHRP Purge Request	MPOA Egress Cache Purge Request
Request ID		Request ID 3	Request ID 2	Request ID 1
Source Protocol Address		I-MPS Protocol Address	E-MPS Protocol Address	E-MPC Protocol Address or NULL
Destination Protocol Address		I-MPC Protocol Address or NULL	I-MPS Protocol Address	E-MPS Protocol Address
Source NBMA Address		I-MPS Data ATM Address	E-MPC Data ATM Address	E-MPC Data ATM Address
Client Protocol Address		Destination Protocol Address (to purge)	Destination Protocol Address (to purge)	Destination Protocol Address (to purge)
Prefix Length		Destination Prefix Length	Destination Prefix Length	Destination Prefix Length
Client NBMA Address		E-MPC Data ATM Address	E-MPC Data ATM Address	E-MPC Data ATM Address
Extensions		Received Extensions	Received Extensions (except for MPOA DLL Header Extension)	MPOA DLL Header Extension (Cache ID) Local Extensions

The reply phase proceeds from left to right as follows:

	Ingress MPC	Ingress MPS	Egress MPS	Egress MPC
Packet Type	NHRP Purge Reply	NHRP Purge Reply	MPOA Egress Cache Purge Reply	
Request ID	Request ID 3	Request ID 2	Request ID 1	

Source Protocol Address	I-MPS Protocol Address	E-MPS Protocol Address	NULL or E-MPC Protocol Address	
Destination Protocol Address	I-MPC Protocol Address or NULL	I-MPS Protocol Address	E-MPS Protocol Address	
Source NBMA Address	I-MPS Data ATM Address	E-MPC Data ATM Address	E-MPC Data ATM Address	
Client Protocol Address	Destination Protocol Address (to purge)	Destination Protocol Address (to purge)	Destination Protocol Address (to purge)	
Prefix Length	Destination Prefix Length	Destination Prefix Length	Destination Prefix Length	
Client NBMA Address	E-MPC Data ATM Address	E-MPC Data ATM Address	E-MPC Data ATM Address	
Extensions	Received Extensions	Received Extensions	Received Extensions	

B.3 Egress MPS-Initiated Egress Cache Purge

Egress MPS-Initiated Egress Cache Purges are transacted with the ingress MPC and the egress MPC simultaneously. Each transaction includes a request phase and a reply phase. The request phase proceeds as follows:

	Ingress MPC	Ingress MPS	Egress MPS	Egress MPS	Egress MPC
Packet Type		NHRP Purge Request	NHRP Purge Request	MPOA Cache Imposition Request	
Direction		←	←	→	
Request ID		Request ID 3	Request ID 2	Request ID 1	
Source Protocol Address		I-MPS Protocol Address	E-MPS Protocol Address	E-MPS Protocol Address	
Destination Protocol Address		I-MPC Protocol Address or NULL	I-MPS Protocol Address	Destination Protocol Address (to purge)	
Source NBMA Address		I-MPS Data ATM Address	E-MPC Data ATM Address	NULL	
Client Protocol Address		Destination Protocol Address (to purge)	Destination Protocol Address (to purge)	NULL	
Prefix Length		Destination Prefix Length	Destination Prefix Length	Destination Prefix Length	
Holding Time				0	
Client NBMA Address		E-MPC Data ATM Address	E-MPC Data ATM Address	NULL	
Extension				MPOA DLL Header Extension (Cache ID) (4)	

The reply phase proceeds as follows:

	Ingress MPC	Ingress MPS	Egress MPS	Egress MPS	Egress MPC
Packet Type	NHRP Purge	NHRP Purge			MPOA Cache

	Reply	Reply			Imposition Reply
Direction	→	→			←
Request ID	Request ID 3	Request ID 2			Request ID 1
Source Protocol Address	I-MPS Protocol Address	E-MPS Protocol Address			E-MPS Protocol Address
Destination Protocol Address	I-MPC Protocol Address or NULL	I-MPS Protocol Address			Destination Protocol Address (to purge)
Source NBMA Address	I-MPS Data ATM Address	E-MPC Data ATM Address			NULL
Client Protocol Address	Destination Protocol Address (to purge)	Destination Protocol Address (to purge)			NULL
Prefix Length	Destination Prefix Length	Destination Prefix Length			Destination Prefix Length
Client NBMA Address	E-MPC Data ATM Address	E-MPC Data ATM Address			NULL
Extensions					Received Extensions

(4) Optional

B.4 Data-Plane Purge

Data-Plane Purges operate in a single phase from right to left as follows:

	Ingress MPC	Egress MPC
Packet Type		NHRP Purge Request
Request ID		Unused. Set to zero
Source Protocol Address		E-MPS Protocol Address or NULL (5)
Destination Protocol Address		NULL
Source NBMA Address		E-MPC Data ATM Address
Client Protocol Address		Destination Protocol Address (to purge)
Prefix Length		Destination Prefix Length
Client NBMA Address		E-MPC Data ATM Address
Extensions		

(5) Use E-MPS Protocol address if purge results from an MPS dying, and NULL if purge results from an egress cache miss.

B.5 MPOA Trigger

MPOA Triggers operate in a single phase from right to left as follows:

	Ingress MPC	Ingress MPS
Packet Type		MPOA Trigger
Request ID		unused
Source Protocol Address		I-MPS Protocol Address
Destination Protocol Address		Destination Protocol Address (to trigger)
Source NBMA Address		I-MPS Control ATM Address
Client Protocol Address (6)		Destination Protocol Address (to trigger)
Prefix Length (6)		Destination Prefix Length (to trigger)
Client NBMA Address (6)		NULL
Extensions		

(6) Optional

B.6 MPOA Keep-Alive

MPOA Keep-Alive messages are sent by both ingress and egress MPSs. MPOA Keep-Alives operate in a single phase from left to right as follows:

	MPS	MPC
Packet Type	MPOA Keep-Alive	
Request ID	Keep-Alive sequence number	
Source Protocol Address	NULL	
Destination Protocol Address	NULL	
Source NBMA Address	MPS Control ATM Address	
Extensions	MPOA Keep-Alive Lifetime Extension	

Annex C. NBMA Next Hop Resolution Protocol (NHRP)

[Normative]

The NBMA Next Hop Resolution Protocol (NRHP) [NHRP] as specified in IETF RFC 2332 is a mandatory part of this specification.

Note: At the time MPOA 1.0 was specified, a draft version of NHRP had been put into this Annex C as a placeholder. It was made clear that the intention was to replace the copy of that draft by a reference to the RFC version of NHRP as soon as it would be available.

Annex D: MPOA 1.1 PICS Proforma

D.1. Introduction

To evaluate conformance of a particular implementation, it is necessary to have a statement of which capabilities and options have been implemented. Such a statement is called a Protocol Implementation Conformance Statement (PICS). To provide a PICS, the questions of a PICS Proforma - as provided by this document - have to be answered for an implementation.

Further information on the purpose of a PICS Proforma is provided by the informative attachment at the end of this Annex D.

D.1.1.Scope

This Annex provides for the PICS proforma for the MPOA 1.1 (Multiprotocol over ATM, Version 1.1) specification of the ATM Forum. With regard to the errata to MPOA 1.0 (see Annex E below), the corrected text of MPOA 1.1 is taken as the basis for the PICS, but references to the previous [MPOA 1.0] statements are attached as notes in those cases where the PICS Proforma are affected.

Basic rules how to fill in a PICS proforma are summarized in section D.2 below.

General rules for the specification of PICS proforma are provided by [ISO 9646-1]. Detailed guidance for the specification of PICS proforma is provided by [ISO 9646-7]; in particular the structure of a PICS proforma, the questions to be asked, the syntax and notation to be used and the semantics of the questions and expected answers.

D.1.2.Normative references

The normative references of section 6 above apply for this PICS Proforma.

D.1.3.Definitions

In addition to the definitions of section 2.1 above, this Annex uses the following terms defined in [ISO 9646-1]:

- A **Protocol Implementation Conformance Statement (PICS)** is a statement made by the supplier of an implementation or system, stating which capabilities have been implemented for a given protocol.
- A **PICS proforma** is a document, in the form of a questionnaire, designed by the protocol specifier or conformance test suite specifier, which when completed for an implementation or system becomes the PICS.

D.1.4.Acronyms

The acronyms and abbreviations of section 2.2 above apply in this Annex.

D.1.5.Conformance

The supplier of a protocol implementation which is claimed to conform to the MPOA 1.1 Specification is required to complete a copy of the PICS proforma provided in this document and is required to provide the information necessary to identify both the supplier and the implementation.

D.2.Instructions

D.2.1.Instructions for completing the PICS proforma

The supplier of a protocol implementation which is claimed to conform to this ATM Forum MPOA 1.1 specification shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

The PICS proforma is a fixed format questionnaire. The supplier of the implementation shall complete this questionnaire, in particular identify the implementation, complete the global statement of conformance, and providing the answers in the rows of the tables in clauses D.5 and following. The structure of the tables is explained in subclauses D.2.4 and D.2.5. For each row in each table, the supplier shall enter an explicit answer (i.e. by ticking the appropriate "yes", "no", or "N/A" in each of the support column boxes provided. Where a support column box is left blank, or where it is marked "N/A" without any tick box, no answer is required. If a "prerequisite line" (see D.2.5 below) is used after a subclause heading or table title, and its predicate is false, no answer is required for the whole subclause or table, respectively.

A supplier may also provide - or be required to provide - further information, categorized as either Additional Information or Exception Information. When present, each kind of further information is to be provided in a further subclause of items labelled

"a.<i>" for additional information,

"x.<i>" for exceptional information

for cross-referencing purposes, where <i> is any unambiguous identification for the item (e.g., simply a numeral); there are no other restrictions on its format and presentation.

D.2.2. Additional Information

Items of Additional Information allow a supplier to provide further information intended to assist the interpretation of the PICS. It is not intended or expected that a large quantity will be supplied, and a PICS can be considered complete without any such information. Examples might be an outline of the ways in which a (single) implementation can be set up to operate in a variety of environments and configurations.

References to items of Additional Information may be entered next to any answer in the questionnaire, and may be included in items of Exception information.

D.2.3. Exception Information

It may occasionally happen that a supplier will wish to answer an item with mandatory or prohibited status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No pre-printed answer will be found in the Support column for this. Instead, the supplier is required to write into the support column an x.<i> reference to an item of Exception Information, and to provide the appropriate rationale in the Exception item itself.

An implementation for which an Exception item is required in this way does not fully conform to this Standard; and the answer to the global statement of conformance (see clause 4) cannot be "yes". A possible reason for the situation described above is that a defect in the Standards has been reported, a correction for which is expected to change the requirement not met by the implementation.

D.2.4 Legend for the columns of the PICS proforma tables

The questionnaire in clauses D.5 and beyond is structured as a set of tables in accordance with the guidelines presented in [ISO 9646-7]. The columns of the tables shall be interpreted as follows:

"Item"

The item column contains a unique reference (a mnemonic plus a number) for each item within the PICS proforma. Items need not always be numbered sequentially.

"Item Description"

The item description column contains a brief summary of the static requirement for which a support answer is required. This may be done by a question or a reference to a specific feature.

"Conditions for Status"

The conditions for status column contains a specification, if appropriate, of the predicate upon which a conditional status is based. The indication of an item reference in this column indicates a simple-predicate condition (support of this item is dependent on the support marked for the referenced item).

Within the "conditions for status" column, the following symbols are used:

- "|" the symbol "|" is used to indicate a logical negation ("NOT"),
- "&" the "&" symbol is used to indicate the logical AND,
- "v" the symbol "v" is used to indicate the logical "OR".

"Status"

The following notations, as defined in [ISO 9646-7], are used for the status column:

- I Irrelevant or out-of-scope - this capability is outside the scope of the standard to which this PICS proforma applies and is not subject to conformance testing in this context.
- M Mandatory - the support of this capability is required for full conformance to the standard.
- N/A Not Applicable - in the given context, it is impossible to use the capability. No answer in the support column is required.
- O Optional - the capability is not required for conformance to the protocol and may be supported or not. However, if the capability is implemented, it is required to conform to the protocol specifications.
- O.<n> Qualified optional - in this case, <n> is an integer that identifies a unique group of related optional items. If no additional qualification is indicated, the support of at least one of the optional items is required for conformance to the standard. Otherwise, the qualification and logic of the selection among the optional items is defined below the table explicitly.
- X eXcluded or prohibited - there is a requirement not to use this capability in a given context.

"Reference"

Except where explicitly stated, the reference column refers to the appropriate subclause(s) of the ATM Forum MPOA Vs. 1.1 specification describing the particular item. The reference merely indicates the place(s) where the core of a description of an item can be found; additional information on this item may be contained in other parts of the MPOA Vs. 1.1 specification, and has to be taken into account when making a statement about the conformance to that particular item.

"Support "

In the support column, the supplier of the implementation shall enter an explicit answer. The following notation is used:

- [] Yes [] No Tick "yes", if item is supported; tick "No", if item is not supported.
- [] N/A Tick "N/A", if the item is "not applicable".

In specific cases, the indication of explicit values may be requested. Where a support column box is left blank, or where it is marked "N/A" without any tick box, no answer is required.

D.2.5.Legend for further indications of the PICS proforma tables

In addition to the columns of a table, the following information may be indicated:

"Prerequisite line"

A prerequisite line after a subclause heading or table title indicates that the whole subclause or the whole table is not required to be completed if the predicate is false. The prerequisite line takes the form:

Prerequisite:<predicate>.

"Qualification"

At the end of a table, a detailed qualification for a group of optional items may be indicated, as specified in the description of the status "qualified optional" in subclause 2.4.

"Comments"

This box at the end of a table allows a supplier to enter any comments to that table. Comments may also be provided separately (without using this box).

D.3. Identification of the implementation

Identification of the implementation and the system in which it resides should be filled in to provide as much detail as possible regarding version numbers and configuration options.

Configuration options outlined in MPOA 1.1 have been incorporated into this PICS proforma. They are referred to by qualified options or prerequisite lines, in order to reflect that an implementation only needs to provide the addressed functions at an interface, if it is configured accordingly.

The contact person indicated (see subclause D.3.6) should be able to answer queries regarding information supplied in the PICS.

As specified in clause 5 of [ISO 9646-7], it is required for all implementations to at least provide the identification of the implementation (D.3.2), product supplier information (D.3.4), identification of a contact person (D.3.6), and detailed identification of the protocol for which the PICS applies (D.3.7). Identification of the system in which the implementation resides (D.3.3) is recommended in order to facilitate full identification of the system, and avoid possible problems during conformance testing. The customer information (D.3.5) only needs to be filled in if it is relevant and different from the product supplier information.

D.3.1. Date of statement

D.3.2. Identification of the implementation

The terms "name" and "version" should be interpreted appropriately to correspond with a suppliers terminology (e.g. Type, Series, Model).

Name of the implementation:

Implementation version:

D.3.3. Identification of the system in which it resides

Name of the system:

Hardware configuration:

Operating system:

D.3.4. Product supplier

Name:

Address:

Telephone number:

Facsimile number:

E-Mail address:

Additional information:

D.3.5.Customer

Name:

Address:

Telephone number:

Facsimile number:

E-Mail address:

Additional information:

D.3.6.PICS contact person

Name:

Address:

Telephone number:

Facsimile number:

E-Mail address:

Additional information:

D.3.7.Protocol for which this PICS applies

Protocol:

ATM Forum Multi-Protocol over ATM (MPOA) Specification

Protocol Version - please identify the specification unambiguously, including e.g. reference number, edition number (if applicable) and publication date:

Version 1.1 (=this specification)

D.5.Roles

The concept of roles is illustrated in subclause 8.5.2 of ISO/IEC 9646-7. For implementations, it is required to identify which roles are supported. Roles are used in PICS proformas as predicates for the "conditions for status" column if an item is conditional upon the role(s) supported.

Item	Roles Is the implementation capable of ...	Conditions for status	Status	Reference	Support
Role1	functioning as an MPOA client (MPC)?		O.1	2.1, 3.1	[]Yes []No
Role1.1	functioning as an Ingress MPC?	Role1] (Role1)	M N/A	2.1, 4.4.5	[]Yes []N/A
Role1.2	functioning as an Egress MPC?	Role1] (Role1)	M N/A	2.1, 4.4.5	[]Yes []N/A
Role2	functioning as an MPOA server (MPS)?		O.1	2.1, 3.1	[]Yes []No
Role2.1	functioning as an Ingress MPS?	Role2] (Role2)	M N/A	2.1	[]Yes []N/A
Role2.2	functioning as an Egress MPS?	Role2] (Role2)	M N/A	2.1	[]Yes []N/A
Role3	functioning as Co-Located MPOA Client and MPOA Server?	Role1 & Role2] (Role1 & Role2)	O N/A	4.2.3	[]Yes []No []N/A
Comments:					

Table D-1: Roles

D.6.Additional Specifications implemented to support ATM Forum MPOA Vs. 1.1

Note: It is strongly recommended that for the supported specifications, separate PICS are provided according to the PICS proforma specified for those specifications, and are attached to this PICS proforma. If not available, it is recommended to at least provide an informal sheet responding to the questions of clauses D.3 and D.4 above for those specifications.

Item	Additional Specifications Supported Does the implementation support ...	Conditions for status	Status	Reference	Support
AddS1.1	ATM Forum UNI 3.0 specification [UNI 3.0] ?		O.2	1.2	[]Yes []No
AddS1.2	ATM Forum UNI 3.1 specification [UNI 3.1] ?		O.2	1.2	[]Yes []No
AddS1.3	ATM Forum UNI 4.0 specification [UNI 4.0] ?		O.2	1.2	[]Yes []No
AddS2	ATM Forum LANE Vs. 2.0 LUNI interface specification [LANE] ?		M	1.2, 3	[]Yes
AddS3	the Next Hop Resolution Protocol (NHRP) as specified in RFC 2332 [NHRP]?	Role2 v Role3 Role 1	M O	1.2, 4, Annex C	[]Yes []Yes []No
AddS4	Multiprotocol encapsulation over ATM Adaptation Layer 5 (IETF RFC 1483) [RFC 1483] ?		M	5.1.1, 4.8.1	[]Yes
Comments:					

Table D-2: Additional Specifications Supported

D.7.MPC Generic Protocol Handling

Prerequisite: Role1

Item	MPC Generic Protocol Handling Is the implementation capable of...	Conditions for status	Status	Reference	Support
C-Kpalv1	supporting the keep-alive protocol and detect MPS failures by missing or out-of-sequence Keep-Alive messages from that MPS?		M	4.6, 5.3.2.4.4	[]Yes
C-Kpalv2	tolerating the release of a VCC over which keep- alive messages were received, without drawing any inference about the state of the related MPS?		O	4.6	[]Yes []No
Comments:					

Table D-3: MPC Generic Protocol Handling

D.8.MPC Configuration

Prerequisite: Role1

Item	MPC Configuration Is the implementation capable of...	Conditions for status	Status	Reference	Support
C-Conf1	obtaining configuration information from an LECS?		M	4.4.1	[]Yes
C-Conf1.1	bypassing the configuration phase and use default or locally configured parameters?		O	4.4.1	[]Yes []No
C-Conf1.2	being associated with more than one LE Client?		O	4.4.1, 3.1.4	[]Yes []No
C-Conf1.2.1	retrieving configuration information over more than one of its LE Clients?	C-Conf1.2] (C-Conf1.2)	O N/A	4.4.1	[]Yes []No []N/A
C-Conf1.3	using configuration information returned to an MPC's LE Client during the LE Client's initialization phase without issuing another Configuration Request?		O	4.4.1	[]Yes []No
C-Conf2	supporting all MPC configuration parameters?		M	4.1.2.1	[]Yes
C-Conf2.1	supporting which value or range of values of parameter MPC-p1 (number between 1 and 65535)?		M	4.1.2.1, 4.4.2	Value/Range: _____
C-Conf2.2	supporting which value or range of values of parameter MPC-p2 (in seconds between 1 and 60)?		M	4.1.2.1, 4.4.2	Value/Range: _____
C-Conf2.3	supporting which set of flow detection protocols as specified by parameter MPC-p3?		M	4.1.2.1	Protocols: _____
C-Conf2.4	supporting which value or range of values of parameter MPC-p4 (in seconds between 1 and 300)?		M	4.1.2.1	Value/Range: _____
C-Conf2.5	supporting which value or range of values of parameter MPC-p5 (in seconds between 10 and 300) ?		M	4.1.2.1	Value/Range: _____
C-Conf2.6	supporting which value or range of values of parameter MPC-p6 (in seconds between 30 and 1200)?		M	4.1.2.1	Value/Range: _____
C-Conf3	supporting the MPC constants?		M	4.1.2.2	[]Yes
Comments:					

Table D-4: MPC Configuration

D.9.MPC Device Discovery

Prerequisite: Role1

Item	MPC Device Discovery Is the implementation capable of...	Conditions for status	Status	Reference	Support
C-Ddisc1	MPS device discovery using LANE?		M	4.2	<input type="checkbox"/> Yes
C-Ddisc1.1	registering the MPOA Device Type TLV with each LE Client on which it is configured?		M	4.2	<input type="checkbox"/> Yes
C-Ddisc1.2	discovering and communicating with at least one MPS in an ELAN?		M	4.2, 3.1.1	<input type="checkbox"/> Yes
C-Ddisc1.2.1	discovering and communicating with more than one MPS in the same ELAN?	C-Ddisc1.2] C-Ddisc1.2	O N/A	4.2, 3.1.1	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-Ddisc1.2.2	discovering and communicating with how many MPSs in the same ELAN (value > 1)?	C-Ddisc1.2.1] C-Ddisc1.2.1	M N/A	4.2, 3.1.1	Value: ____ <input type="checkbox"/> N/A
C-Ddisc2	registering the MPOA Device Type TLV with each LE Client on which it is configured?		M	4.2, 4.2.1, 3.1.4	<input type="checkbox"/> Yes
C-Ddisc2.1	requesting each of its associated LE Clients to send an Unregister Request to the LES for each registered MAC address on the change of device status?	C-Ddisc2] C-Ddisc2	O N/A	4.2.1	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-Ddisc2.1.1	requesting each of its associated LE Clients to re-register at the LES with the new set of TLVs after unregistering?	C-Ddisc2.1] (C-Ddisc2.1)	O N/A	4.2.1	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-Ddisc2.2	requesting its LE Clients to send a targetless LE_ARP_REQUEST to the LES to indicate MPOA device changes?	C-Ddisc2] C-Ddisc2	O N/A	4.2.2, 4.2.4.2	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-Ddisc2.2.1	requesting each of the associated LE Clients, if the status of MPOA device changes, to send a Targetless_LE_ARP_REQUEST to the LES for MAC addresses previously included in an address resolution response with the new set of TLVs?	C-Ddisc2.2] (C-Ddisc2.2)	O N/A	4.2.2	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-Ddisc2.3	responding to an MPOA request with error code 0x86 to indicate an MPOA device change?	C-Ddisc2] C-Ddisc2	O N/A	4.2.4.3	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-Ddisc3	in case of a co-located MPC / non-MPOA device, separating the corresponding two sets of LE Client ATM addresses, and separating the related VCCs		M	4.2.3	<input type="checkbox"/> Yes
Comments:					

Table D-5: MPC Device Discovery

D.10.MPC Target Resolution

D.10.1.MPC Data Path Processing

Prerequisite: Role1

Item	MPC Data Path Processing Is the implementation capable of ...	Conditions for status	Status	Reference	Support
C-DataF11	supporting the detailed behavioural MPC diagram for data path processing?		M	Figure 8, 4.4	<input type="checkbox"/> Yes
C-DataF12	supporting the procedures for inbound data flow?	Role1.1] Role1.1	M N/A	4.4.2	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
C-DataF12.1	including a(n) (internetwork layer) source protocol address in the MPOA Resolution Request different from NULL?	Role1.1] Role1.1	O N/A	4.4.2	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-DataF12.2	including an MPOA Service Category extension in the MPOA Resolution Request?	Role1.1] Role1.1	O N/A	4.4.2, 4.8.9.1	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-DataF12.3	including additional extensions in the MPOA Resolution Request as specified in NHRP?	Role1.1] Role1.1	O N/A	4.4.2	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-DataF12.4	including the MPOA Egress Cache Tag Extension in the MPOA Resolution Request?	Role1.1] Role1.1	M N/A	4.4.2	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
C-DataF13	supporting the procedures for outbound data flow?	Role1.2] Role1.2	M N/A	4.4.3	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
C-DataF13.1	including the MPOA Egress Cache Tag Extension in the Cache Imposition Reply if it was included in the corresponding Request?	Role1.2] Role1.2	O N/A	4.4.3	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-DataF13.2	supporting unique keys for egress cache entries?	Role1.2] Role1.2	M N/A	4.4.3	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
C-DataF13.2.1	associating a distinct tag value with every egress cache entry?	C-DataF13.2] C-DataF13.2	O.4 N/A	4.4.3, 5.3.2.4.2	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-DataF13.2.2	using tags as part of the key for the egress cache entry?	C-DataF13.2] C-DataF13.2	O.4 N/A	4.4.3, 5.3.2.4.2	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-DataF13.2.3	always using different destination ATM addresses with different DLL headers?	C-DataF13.2] C-DataF13.2	O.4 N/A	4.4.3	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-DataF13.3	continuing forwarding packets from the LE Client interface to the bridge for up to 30 seconds in case of a cache hit, but no MAC destination address?	Role1.2] Role1.2	O N/A	4.4.3, 4.7.1.2	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-DataF13.4	adding padding, if necessary to comply with the minimum frame size of the egress ELAN?	Role1.2] Role1.2	M N/A	4.4.3	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
Comments:					

Table D-6: MPC Data Path Processing

D.10.2.MPC Cache Management

Prerequisite: Role1

Item	MPC Cache Management Is the implementation capable of ...	Conditions for status	Status	Reference	Support
C-CacheM1	supporting ingress cache entry creation and management?	Role1.1] Role1.1	M N/A	4.4.4.1	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
C-CacheM1.1	using the retry procedure if an MPOA Resolution Reply is not received?	Role1.1] Role1.1	O N/A	4.4.4.1, 4.3, 4.4.6.1.4	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-CacheM1.1.1	using the same Request Id. in the MPOA Resolution Retry as in the outstanding request(s)?	C-CacheM1.1] C-CacheM1.1	M (Note) N/A	4.3,4.4.6.1.4	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-CacheM1.1.2	autonomously terminating the retry procedure at any time, deciding a shortcut is no longer needed?	C-CacheM1.1] C-CacheM1.1	O N/A	4.4.6.1.4	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-CacheM1.2	withdrawing ingress cache entries autonomously at any time for local reasons?	Role1.1] Role1.1	O N/A	4.4.4.1	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-CacheM1.3	immediately re-issuing a new MPOA Resolution Request after a shortcut stop caused by a Purge?	Role1.1] Role1.1	O.5 N/A	4.4.4.1	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-CacheM1.4	waiting and using local information to determine when to issue a new MPOA Resolution Request after a shortcut stop caused by a Purge?	Role1.1] Role1.1	O.5 N/A	4.4.4.1	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-CacheM2	supporting egress cache entry creation and management?	Role1.2] Role1.2	M N/A	4.4.4.2	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
C-CacheM2.1	discarding packets received over a shortcut after the egress cache entry has become invalidated?	Role1.2] Role1.2	M N/A	4.4.4.2	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
C-CacheM3	supporting ingress cache control information: shortcut entry state, shortcut VCC state, Ingress MPS Control ATM address, Last NHRP CIE Code, Last Q.2931 Cause Value, ATM address of Egress-MPC, Holding Time, Request-Id, Retry Time, Packets Forwarded	Role1.1] Role1.1	M N/A	4.4.6.1	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
C-CacheM3.1	saving the received ATM Service Category Extension while setting up shortcut VCCs?	Role1.1] Role1.1	O N/A	4.4.6.1.2	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-CacheM4	supporting egress cache control information: shortcut entry state, egress MPS Control ATM Address, Cache Id., Ingress MPC/NHC data ATM Address, holding time?	Role1.2] Role1.2	M N/A	4.4.6.2	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
C-CacheM4.1	using the Ingress MPC/NHC data ATM Address to verify the sender identity of shortcut packets?	C-CacheM4] C-CacheM4	O N/A	4.4.6.2.2	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-CacheM4.2	supporting control information on "packets received" (via a shortcut VCC)?	Role1.2] Role1.2	O N/A	4.4.6.2.4	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
Comments: O.5: Exactly one of these options must be supported Note: It should be noted that section 4.4.6.1.4 of the original [MPOA 1.0] specification did refer to C-CacheM1.1.1 as being optional ("should"). This was inconsistent with section 4.3, and has been corrected.					

Table D-7: MPC Cache Management

D.10.3.MPC Cache Maintenance

Prerequisite: Role1

Item	MPC Cache Maintenance Is the implementation capable of ...	Conditions for status	Status	Reference	Support
C-CMaint1	invalidation of imposed egress cache entries?	Role1.2] Role1.2	M N/A	4.7.1.2	[]Yes [] N/A
C-CMaint2	the protocol for MPC-initiated Egress Cache Purge to notify the egress MPS on invalid cache entries?	Role1.2] Role1.2	O N/A	4.7.1.6	[]Yes []No [] N/A
C-CMaint2.1	including the following information in an MPOA Egress Cache Purge Request (Note 1): Request ID, Egress MPS Protocol Address, Egress MPC Data ATM Address, Destination Protocol Address, Destination Prefix Length, MPOA DLL Header Extension, no-reply flag	C-CMaint2] C-CMaint2	M (Note1) N/A	4.7.1.6	[]Yes [] N/A
C-CMaint3	supporting the procedures for handling an MPOA trigger from an MPS?	Role1.1] Role1.1	M N/A	4.7.2.1	[]Yes [] N/A
C-CMaint4	supporting the procedures for egress Data Plane Purge?	Role1.2] Role1.2	M N/A	4.7.2.3	[]Yes [] N/A
C-CMaint4.1.1	sending a data plane NHRP Purge Requests in case of missing Keep-Alive messages from the related MPS?	C-CMaint4] C-CMaint4	O.6 N/A	4.7.2.3 (Note 2)	[]Yes []No [] N/A
C-CMaint4.1.2	invalidate cache entries locally in case of missing Keep-Alive messages from the related MPS?	C-CMaint4] C-CMaint4	O.6 N/A	4.7.2.3 (Note 2)	[]Yes []No [] N/A
C-CMaint4.2	sending a data plane NHRP Purge Request over a shortcut in case of an egress cache lookup failure for a received packet over that shortcut?	C-CMaint4] C-CMaint4	M N/A	4.7.2.3	[]Yes [] N/A
C-CMaint4.2.1	periodically repeating the NHRP Data Plane Purge Request to the ingress MPC to recover from continuing receipt of invalid data packets?	C-CMaint4.2] C-CMaint4.2	M N/A	4.7.1.4, 4.7.2.3	[]Yes [] N/A
C-CMaint4.2.2	periodically repeating the NHRP Purge Request at which frequency?	C-CMaint4.2.1] C-CMaint4.2.1	M N/A	4.7.1.4, 4.7.2.3	_____ (value) [] N/A
C-CMaint5	supporting the procedures for receiving and processing an NHRP Purge Request on a shortcut?	Role1.1] Role1.1	M N/A	4.7.2.3	[]Yes [] N/A
C-CMaint6	ignoring source address information received in an NHRP ingress cache purge request for the decision which entries to purge?	Role1.1] Role1.1	O N/A	4.7.2.2	[]Yes []No [] N/A
<p>Comments:</p> <p>Note 1: The option to include the "Ingress MPC Data ATM Address", as allowed by the [MPOA 1.0] specification was considered to be inconsistent, and has been discarded within the MPOA 1.1. It should also be noted that MPOA 1.1 clarifies that all the items listed in C-CMaint2.1 are mandatory for the MPOA Egress Cache Purge Request.</p> <p>Note 2: C-CMaint4.1.1 was regarded as being mandatory in the original [MPOA 1.0] specification. MPOA 1.1 also allows C-CMaint4.1.2 as an alternative which is not specified in the original [MPOA 1.0] specification.</p> <p>O.6: Exactly one of these alternatives must be supported.</p>					

Table D-8: MPC Cache Maintenance

D.11.MPC Connection Management

Prerequisite: Role1

Item	MPC Connection Management Is the implementation capable of...	Conditions for status	Status	Reference	Support
C-Cman1	supporting LLC/SNAP encapsulation for all PDUs?		M	4.8.1, 5.1.1	<input type="checkbox"/> Yes
C-Cman1.1	supporting NULL encapsulation?		O	4.8.1, 5.1.1	<input type="checkbox"/> Yes <input type="checkbox"/> No
C-Cman2.1	supporting the sharing of a VCC by multiple protocols (LLC multiplexing)?		O	4.8.1	<input type="checkbox"/> Yes <input type="checkbox"/> No
C-Cman2.2	forcing separation of VCCs for different protocols between the same endpoints?	C-Cman2.1] C-Cman2.1	N/A O	4.8.1	<input type="checkbox"/> N/A <input type="checkbox"/> Yes <input type="checkbox"/> No
C-Cman3	establishing, receiving and maintaining a VCC to any entity that conforms to the connection management procedures as specified in [MPOA 1.0], whether or not the entity is an MPOA component?		M	4.8.2	<input type="checkbox"/> Yes
C-Cman4	initiating VCC establishing calls?		M	4.8.3	<input type="checkbox"/> Yes
C-Cman4.1	not immediately re-establishing VCCs terminated by the remote party, unless it has PDUs to transfer?		O	4.8.8	<input type="checkbox"/> Yes <input type="checkbox"/> No
C-Cman4.2	allowing VCCs to idle out based on inactivity?		O	4.8.8	<input type="checkbox"/> Yes <input type="checkbox"/> No
C-Cman5.1	setting up of additional VCCs to the same MPOA component?	C-Cman2.2] C-Cman2.2	M O	4.8.3 4.8.5	<input type="checkbox"/> Yes <input type="checkbox"/> Yes <input type="checkbox"/> No
C-Cman5.2	supporting multiple VCCs between peer systems?		M	4.8.5	<input type="checkbox"/> Yes
C-Cman5.3	supporting the procedures for reducing the number of redundant VCCs between peer systems?	C-Cman5.2] C-Cman5.2	O N/A	4.8.5	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-Cman5.4	accepting all incoming VCCs that indicate the use of LLC encapsulation?		O	4.8.4	<input type="checkbox"/> Yes <input type="checkbox"/> No
C-Cman6.1	learning Internetwork Layer Address-to-ATM Address mappings as a result of using the InATMARP protocol on VCCs?		X	4.8.6	<input type="checkbox"/> No
C-Cman6.2	deriving Internetwork Layer-to-ATM Address mappings by learning from the Source Protocol and Source NBMA Address fields in an MPOA/NHRP Resolution Request?		X	4.8.6	<input type="checkbox"/> No
C-Cman7	establishing VCCs according to the MPOA rules for usage of UNI signaling information elements?		M	4.8.9	<input type="checkbox"/> Yes
C-Cman7.1	establishing point-to-multipoint calls?		O	4.8.9	<input type="checkbox"/> Yes <input type="checkbox"/> No
C-Cman7.2	accepting VCCs with the UBR service category?		M	4.8.9.1	<input type="checkbox"/> Yes
C-Cman7.3	accepting VCCs with other than the UBR service category?		O	4.8.9.1	<input type="checkbox"/> Yes <input type="checkbox"/> No
C-Cman7.4	retrying a UBR-VCC for transferring control messages, if a non-UBR was not accepted?		M	4.8.9.1	<input type="checkbox"/> Yes
C-Cman7.5	supporting UNI 3.0 considerations for Traffic Description?	AddS1.1] AddS1.1	M N/A	4.8.9.1	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
C-Cman7.6	supporting UNI 4.0 considerations for Traffic Description?	AddS1.3] AddS1.3	M N/A	4.8.9.1	<input type="checkbox"/> Yes <input type="checkbox"/> N/A

C-Cman8	only using class 0 under the UBR Service Category?		M	4.8.9.2	<input type="checkbox"/> Yes
C-Cman8.1	treating coding standard field values "00" and "11" as equivalent for QoS class 0?		O	4.8.9.2	<input type="checkbox"/> Yes <input type="checkbox"/> No
C-Cman9	supporting the default CPCS-SDU size for both forward and backward directions?		M	4.8.9.3	<input type="checkbox"/> Yes
C-Cman9.1	initiating a negotiation of a larger CPCS-SDU size?		O	4.8.9.3	<input type="checkbox"/> Yes <input type="checkbox"/> No
C-Cman9.2	supporting fragmentation of internetwork layer packets to meet remote party MTU capabilities?		O	4.8.9.3	<input type="checkbox"/> Yes <input type="checkbox"/> No
C-Cman9.3	always including the AAL parameter IE in both SETUP and CONNECT messages?		M	4.8.9.3	<input type="checkbox"/> Yes
C-Cman9.3.1	omitting the SSCS type parameter of the AAL parameter IE, or coding it as null-SSCS?	C-Cman9.3] C-Cman9.3	M N/A	4.8.9.3	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
C-Cman9.4	supporting the UNI 3.0 considerations for AAL5 Parameters?	AddS1.1] AddS1.1	M N/A	4.8.9.3	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
C-Cman9.4.1	omitting the UNI 3.0 mode parameter, or setting it to 1 to indicate message mode?	C-Cman9.4] C-Cman9.4	O N/A	4.8.9.3	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-Cman9.5	recognizing NHRP-only interoperability situations with regard to CPCS-SDU sizes?		O	4.8.9.3	<input type="checkbox"/> Yes <input type="checkbox"/> No
C-Cman10	supporting the use of the B-LLI IE?		M	4.8.9.4	<input type="checkbox"/> Yes
C-Cman10.1	supporting B-LLI negotiation?	C-Cman10] C-Cman10	O N/A	4.8.9.4	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-Cman11	supporting the use of BCOB-C and BCOB-X?		M	4.8.9.5	<input type="checkbox"/> Yes
C-Cman11.1	retrying call establishment with the other value, if a call with one of the values BCOB-C or -X is rejected due to a Bearer Capability problem?	C-Cman11] C-Cman11	O N/A	4.8.9.5	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-Cman12	supporting all three formats of private ATM addresses?		M	4.8.9.6	<input type="checkbox"/> Yes
C-Cman12.1	supporting the use of native E.164 ATM addresses?	C-Cman12] C-Cman12	O N/A	4.8.9.6	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-Cman12.2	supporting the use of Calling Party Subaddress or Called Party Subaddress in the SETUP message?	C-Cman12] C-Cman12	O N/A	4.8.9.6	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
C-Cman12.3	supporting UNI3.0 considerations for ATM addressing?	AddS1.1] AddS1.1	M N/A	4.8.9.6	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
Comments:					

Table D-9: MPC Connection Management

D.12.MPC Data Transfer

Prerequisite: Role1

Ed. Note: Requirements for MPC Data Transfer have been included in other sections above.

D.13. MPS Configuration

Prerequisite: Role2

Item	MPS Configuration Is the implementation capable of...	Conditions for status	Status	Reference	Support
S-Conf1	obtaining configuration information from a LECS?		M	4.5.1	<input type="checkbox"/> Yes
S-Conf1.1	bypass the configuration phase and use default or locally configured parameters?		O	4.5.1	<input type="checkbox"/> Yes <input type="checkbox"/> No
S-Conf1.2	retrieving configuration information via an associated LE Client during the LE Client's initialization phase?		O	4.5.1	<input type="checkbox"/> Yes <input type="checkbox"/> No
S-Conf1.2.1	using configuration information returned by one associated LE Client without further issuing another configuration request?	S-Conf1.2] (S-Conf1.2)	O N/A	4.5.1	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
S-Conf2	supporting all MPS configuration parameters?		M	4.1.1.1	<input type="checkbox"/> Yes
S-Conf2.1	supporting which value or range of values for parameter MPS-p1 (number of seconds between 1 and 300)?		M	4.1.1.1	Value/Range: _____
S-Conf2.2	supporting which value or range of values for parameter MPS-p2 (number of seconds between 3 and 1000)?		M	4.1.1.1	Value/Range: _____
S-Conf2.3	supporting which set of Internetwork-layer Protocols MPS-p3?		M	4.1.1.1	Protocols: _____
S-Conf2.4	supporting which value or range of values for parameter MPS-p4 (number of seconds between 1 and 300)?		M	4.1.1.1	Value/Range: _____
S-Conf2.5	supporting which value or range of values for parameter MPS-p5 (number of seconds between 10 and 300)?		M	4.1.1.1	Value/Range: _____
S-Conf2.6	supporting which value or range of values for parameter MPS-p6 (number of seconds between 5 and 300)?		M	4.1.1.1	Value/Range: _____
S-Conf2.7	supporting parameter MPS-p7 ?		O	4.1.1.1	<input type="checkbox"/> Yes <input type="checkbox"/> No
S-Conf2.7.1	supporting which value or range of values for parameter MPS-p7 (number of minutes between 1 and 120)?	S-Conf2.7] (S-Conf 2.7)	M N/A	4.1.1.1	Value/Range: _____ <input type="checkbox"/> N/A
S-Conf3	supporting the MPS constant MPS-c1?		M	4.1.1.1	<input type="checkbox"/> Yes
Comments:					

Table D-10: MPS Configuration

D.14. MPS Device Discovery

Prerequisite: Role2

Item	MPS Device Discovery Is the implementation capable of...	Conditions for status	Status	Reference	Support
S-Ddisc1	MPC device discovery using LANE?		M	4.2	<input type="checkbox"/> Yes
S-Ddisc1.1	MPSs discovering each other if they are sharing an ELAN to facilitate the forwarding of NHRP messages?		M	4.2	<input type="checkbox"/> Yes
S-Ddisc1.2	registering the MPOA Device Type TLV with each LE Client on which it is configured?		M	4.2	<input type="checkbox"/> Yes
S-Ddisc2	supporting the association with more than one LE Client?		M	4.2, 3.1.4	<input type="checkbox"/> Yes
S-Ddisc3	registering the MPOA Device Type TLV with each LE Client on which it is configured?		M	4.2	<input type="checkbox"/> Yes
S-Ddisc3.1	requesting each of its associated LE Clients to send an Unregister Request to the LES for each registered MAC address on the change of device status?	S-Ddisc3] (S-Ddisc3)	O N/A	4.2.1	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
S-Ddisc3.1.1	requesting each of its associated LE Clients to re-register at the LES with the new set of TLVs after unregistering?	S-Ddisc3.1] (S-Ddisc3.1)	O N/A	4.2.1	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
S-Ddisc3.2	requesting its LE Clients to send a targetless LE_ARP_REQUEST to the LES to indicate MPOA device changes?	S-Ddisc3] S-Ddisc3	O N/A	4.2.2, 4.2.4.2	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
S-Ddisc 3.2.1	requesting each of the associated LE Clients, if the status of MPOA device changes, to send a Targetless_LE_ARP_REQUEST to the LES for MAC addresses previously included in an address resolution response with the new set of TLVs?	S-Ddisc3.2] (S-Ddisc3.2)	O N/A	4.2.2	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
S-Ddisc3.3	responding to an MPOA request with error code 0x87 to indicate an MPOA device change?	S-Ddisc2] S-Ddisc2	O N/A	4.2.4.3	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
S-Ddisc4	supporting dynamic address resolution to know if an NHRP request resolves to the ATM address of an MPOA Client?		M	4.2	<input type="checkbox"/> Yes
S-Ddisc5	in case of a co-located MPS / non-MPOA device, separating the corresponding two sets of LE Client ATM addresses, and separating the related VCCs		O	4.2.3	<input type="checkbox"/> Yes
Comments:					

Table D-11: MPS Device Discovery

D.15. MPS Connection Management

Prerequisite: Role2

Item	MPS Connection Management Is the implementation capable of...	Conditions for status	Status	Reference	Support
S-Cman1	supporting LLC/SNAP encapsulation for all PDUs?		M	4.8.1, .5.1.1	[]Yes
S-Cman1.1	supporting NULL encapsulation?		O	4.8.1, 5.1.1	[]Yes []No
S-Cman2	supporting the sharing of VCC by multiple protocols (LLC multiplexing)?		O	4.8.1	[]Yes []No
S-Cman3	establishing, receiving and maintaining a VCC to any entity that conforms to the connection management procedures as specified in MPOA V1.0, whether or not the entity is an MPOA component?		M	4.8.2	[]Yes
S-Cman5	setting up of multiple VCCs to the same MPOA component?		O	4.8.3, 4.8.5	[]Yes []No
S-Cman6	accepting all incoming VCCs that indicate the use of LLC encapsulation?		O	4.8.4	[]Yes []No
S-Cman7	accepting VCCs with the UBR service category?		M	4.8.9.1	[]Yes
S-Cman7.1	accepting VCCs with other than the UBR service category?		O	4.8.9.1	[]Yes []No
S-Cman8	only using class 0 under the UBR Service Category?		M	4.8.9.2	[]Yes
S-Cman9	supporting the default CPCS-SDU size for both forward and backward directions?		M	4.8.9.3	[]Yes
S-Cman10	supporting all three formats of private ATM addresses?		M	4.8.9.6	[]Yes
Comments:					

Table D-12: MPS Connection Management

D.16.MPS Generic Protocol Handling

D.16.1.Keep-Alive Protocol

Prerequisite: Role2

Item	MPS Keep-Alive Protocol Is the implementation capable of...	Conditions for status	Status	Reference	Support
S-Kpalv1	transmitting MPOA Keep-Alive messages to all MPCs for which ingress/egress cache entries are supplied and maintained?		M	4.6	<input type="checkbox"/> Yes
S-Kpalv1.1	transmitting MPOA Keep-Alive every MPS-p1 seconds?		M	4.6	<input type="checkbox"/> Yes
S-Kpalv1.2	maintaining a monotonically increasing sequence number?		M	4.6	<input type="checkbox"/> Yes
S-Kpalv1.3	transmitting Keep-Alive over a point-to-multipoint VCC specifically established for that purpose?		O	4.6	<input type="checkbox"/> Yes <input type="checkbox"/> No
Comments:					

Table D-13: MPS Keep-Alive Protocol

D.16.2. Cache Maintenance

Prerequisite: Role2

Item	MPS Cache Maintenance Is the implementation capable of...	Conditions for status	Status	Reference	Support
S-Cache1	maintaining the ingress cache entries for the positive MPOA resolution replies sent to the MPC's?	Role2.1] Role2.1	M N/A	4.5	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
S-Cache2	maintaining the egress cache entries for the positive MPOA cache imposition replies received from the MPC's?	Role2.2] Role2.2	M N/A	4.7.1, 4.5	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
S-Cache3	maintaining the egress cache for the duration of the cache holding time?	Role2.2] Role2.2	M N/A	4.7.1	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
S-Cache4	sending an NHRP Purge Requests to the set of affected sources of relevant resolution requests, after detecting a change for a destination internetwork layer address?	Role 2.2] Role2.2	O.7 N/A	4.7.1.1	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
S-Cache5	sending an MPOA Cache Imposition Request with a holding time of zero to the egress MPC's with affected egress cache entries, after detecting a change for a destination internetwork layer address?	Role2.2 & S-Cache4] (Role2.2 & S-Cache4)	M N/A	4.7.1.1	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
S-Cache6	sending an MPOA Cache Imposition Request with a an updated DLL header to the egress MPC's with affected egress cache entries, after detecting a change for a destination internetwork layer address?	Role 2.2] Role2.2	O.7 N/A	4.7.1.1	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
S-Cache7	purging all egress cache entries in an MPC for a given destination protocol address by including the protocol address in the CIE and omitting the MPOA DLL Header Extension?	Role 2.2]Role 2.2	O N/A	4.7.1.1	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
S-Cache8	sending the MPOA Cache Imposition Request to the egress MPC using the same VCC as was used for the original cache imposition?	Role 2.2] Role 2.2	M N/A	4.7.1.1	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
S-Cache9	sending egress cache updates reliably using the retry mechanism?	Role 2.2] Role 2.2	M N/A	4.7.1.1, 4.3	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
S-Cache10	holding the egress cache entries imposed by it on an egress MPC until the holding time expires or it has sent MPS cache purges?	Role 2.2] Role 2.2	M N/A	4.7.1.3	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
S-Cache11	generating an NHRP purge request for a received MPOA egress cache purge request?	Role 2.2] Role 2.2	M N/A	4.7.1.6	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
S-Cache12	sending NHRP ingress cache purge requests to all relevant MPCs, when ingress MPS receives an NHRP purge request for which it is maintaining ingress state for the purged destination address(es)?	Role2.1] Role2.1	M N/A	4.7.2.2	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
S-Cache13	always setting the no-reply flag to "1" in the NHRP ingress cache purge request (indicating that no reply is expected)?	S-Cache12] S-Cache12	O N/A	4.7.2.2	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
S-Cache14	ensuring reliable delivery of NHRP ingress cache purge requests to relevant MPCs by the retry procedures when the no-reply flag is clear in the received NHRP purge request?	S-Cache13] S-Cache13	N/A O (Note 1)	4.7.2.2, 4.3	<input type="checkbox"/> N/A <input type="checkbox"/> Yes <input type="checkbox"/> No

S-Cache15	indicating its own Protocol and ATM addresses as source address information within an NHRP ingress cache purge request ?	Role2.1] Role2.1	M N/A (Note 2)	4.7.2.2, 5.3.12	[] Yes [] N/A
Comments: O.7: exactly one of these alternatives must be supported Note 1: It should be noted that this capability S-Cache14 was qualified as being mandatory in [MPOA 1.0]. Correspondingly, S-Cache 13 was prohibited in [MPOA 1.0]. Note 2: It should be noted that this is a deviation from MPOA 1.0, based on the resolution of an inconsistency in [MPOA 1.0].					

Table D-14: MPS Cache Maintenance

D.16.3. Trigger

Prerequisite: Role2

Item	MPOA Trigger Is the implementation capable of...	Conditions for status	Status	Reference	Support
S-MTrigg0	supporting MPOA Trigger?		O	4.7.2.1	[]Yes []No
S-MTrigg1	generating an MPOA trigger message?	S-MTrigg0] (S-MTrigg0)	M N/A	4.7.2.1	[] Yes [] N/A
S-MTrigg2	using the MPOA retry procedure to control the sending of MPOA trigger messages?	S-MTrigg 1] (S-MTrigg 1)	M N/A	4.7.2.1, 4.3	[] Yes [] N/A
S-MTrigg3	adding a CIE with the prefix set appropriately to request an address-prefix based shortcut?	S-MTrigg 1] (S-MTrigg 1)	O (Note) N/A	4.7.2.1	[] Yes [] No [] N/A
Comments: Note: It should be noted that there was an inconsistency in the original [MPOA 1.0] specification. MPOA 1.1 specifies a consistent handling.					

Table D-15: MPOA Trigger

D.17. MPS and NHRP Relationship

Prerequisite: Role2

Item	MPS and NHRP Relationship Is the implementation capable of...	Conditions for status	Status	Reference	Support
S-Nhrp 1	reoriginating the MPOA resolution request as an NHRP resolution request?		M	4.5.2.1	<input type="checkbox"/> Yes
S-Nhrp 2	reoriginating the NHRP resolution reply as an MPOA resolution reply?		M	4.5.2.4	<input type="checkbox"/> Yes
S-Nhrp 3	translating NHRP resolution requests to MPOA cache imposition requests?		M	4.5.2.2	<input type="checkbox"/> Yes
S-Nhrp 4	translating MPOA cache imposition replies to NHRP resolution replies?		M	4.5.2.3	<input type="checkbox"/> Yes
Comments:					

Table D-16: MPS and NHRP Relationship

D.18. MPS Data Transfer

Prerequisite: Role2

Not applicable for MPS

D.19. Co-Located MPS and MPC

Prerequisite: Role3

Item	Co-Located MPS and MPC Is the implementation capable of...	Conditions for status	Status	Reference	Support
Col-Ddisc1	complying with the requirements for co-located MPC/MPSs with regard to device discovery?		M	4.2.3	<input type="checkbox"/> Yes
Col-Ddisc1.1	separating MPC- and MPS-related sets of LE Client ATM addresses, and separating the related VCCs?		O	4.2.3	<input type="checkbox"/> Yes <input type="checkbox"/> No
Comments:					

Table D-17: Co-Located MPS and MPC

D.20. Frame Formats

Item	Frame Formats Does the implementation support ...	Conditions for status	Status	Reference	Support
Frame-F1	MPOA control messages and their frame formats as specified for MPOA 1.0/NHRP?		M	5	<input type="checkbox"/> Yes
Frame-F1.1	NHRP Purge Request and NHRP Purge Reply control messages as specified for NHRP?		M	5.3	<input type="checkbox"/> Yes
Frame-F2	MPOA Tagged Encapsulation for Routed Protocols?		O	5.1.1, Figure 12	<input type="checkbox"/> Yes <input type="checkbox"/> No
Frame-F3	B-LLI negotiation of alternative encapsulations for control frames?		O	5.1.2	<input type="checkbox"/> Yes <input type="checkbox"/> No
Frame-F4	use of the MPOA Original Error Code Extension to preserve the original MPOA error code through conversions to and from NHRP messages?		O	5.3.2.4.6	<input type="checkbox"/> Yes <input type="checkbox"/> No
Frame-F5	adding a prefix length CIE to the MPOA Resolution Request Format in conformance with NHRP?		O	5.3.3	<input type="checkbox"/> Yes <input type="checkbox"/> No
Frame-F6.1	allowing extensions for NHRP Authentication in the data plane NHRP Purge?		O	5.3.11	<input type="checkbox"/> Yes <input type="checkbox"/> No
Frame-F6.2	allowing extensions for Vendor Private Extensions in the data plane NHRP Purge?		O	5.3.11	<input type="checkbox"/> Yes <input type="checkbox"/> No
Comments:					

Table D-18: Frame Formats

D.21. IP Packet Handling

Item	IP Packet Handling Is the implementation capable of ...	Conditions for status	Status	Reference	Support
IP1	performing basic IP forwarding?		M	A.1, A.1.4	<input type="checkbox"/> Yes
IP2	processing of IP packets in conformance with IETF RFC 1812?	Role1 & IP1 Role 2 v Role 3 <input type="checkbox"/> IP1	O M N/A	A.1, A.1.3, A.1.4	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> Yes <input type="checkbox"/> N/A
IP3	sending received IP packets to the MPS unmodified if indicated IP options cannot be processed correctly?	Role1.1 <input type="checkbox"/> Role1.1	M N/A	A.1.4	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
IP3.1	sending all packets with IP options to the MPS?	IP3 <input type="checkbox"/> IP3	O.8 N/A	A.1.4	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
IP3.2	processing IP packets with IP options it can fully support?	IP3 <input type="checkbox"/> IP3	O.8 N/A	A.1.4	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
IP4.1	handling of the TTL field in an IP packet?	Role1.1 <input type="checkbox"/> Role1.1	O N/A	A.1.4	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
IP4.2	modifying the IP checksum to reflect all changes to the IP header?	Role1.1 <input type="checkbox"/> Role1.1	O N/A	A.1.4	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
IP4.3	generating appropriate ICMP Error and sending applicable ICMP messages in error conditions?	Role1.1 <input type="checkbox"/> Role1.1	O N/A	A.1.4	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
IP5.1	establishing shortcuts and fragmenting IP packets where necessary, if the MTU returned in the MPOA Resolution Reply is smaller than the MTU on the inbound interface	C1* <input type="checkbox"/> C1*	O N/A	A.1.4	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
IP5.2	refraining from establishing shortcuts, if the MTU returned in the MPOA Resolution Reply is smaller than the MTU on the inbound interface?	IP5.1 <input type="checkbox"/> IP5.1	O M	A.1.4	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> Yes
IP6.1	advertising a large MTU and fragment received IP packets itself?	Role1.2 <input type="checkbox"/> Role1.2	O N/A	A.1.5	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
IP6.2	decrement TTL of received IP packets?	Role1.2 <input type="checkbox"/> Role1.2	O N/A	A.1.4	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
Comments: C1* := C-Cman9.2 & Role1.1					

Table D-19: IP Packet Handling

D.22. IPX Packet Handling

Item	IPX Packet Handling Is the implementation capable of ...	Conditions for status	Status	Reference	Support
IPX1	processing IPX packets in conformance with IPX router requirements?	Role 2 v Role3 Role1	M O	A.2.3	<input type="checkbox"/> Yes <input type="checkbox"/> Yes <input type="checkbox"/> No
IPX2	including a hop count extension in MPOA Resolution Replies as required by the IPX router requirements?	Role2.1] Role2.1	M N/A	A.2.3	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
IPX3	incrementing the IPX TC field before sending an IPX packet on a shortcut?	Role1.1] Role1.1	M N/A	A.2.3	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
IPX3.1	only forwarding an internetworking packet over a shortcut VCC, if the IPX transport control field value is less than the value specified in the hop count extension?	Role1.1] Role1.1	M N/A	A.2.3	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
IPX4	forwarding an IPX packet unmodified to the MPS, if the source network number is zero?	Role1.1] Role1.1	M N/A	A.2.3	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
IPX5	interworking with a Novell proprietary Raw Ethernet encapsulation?	Role1.2] Role1.2	O N/A	A.2.3	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
IPX5.1	setting the checksum field to 0xFFFF, if forwarding an IPX packet to the Novell proprietary Raw Ethernet encapsulation?	IP5] IP5	M N/A	A.2.3	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
IPX6	removing shortcut encapsulation and adding LAN encapsulation for IPX packets?	Role1.2] Role1.2	M N/A	A.2.3	<input type="checkbox"/> Yes <input type="checkbox"/> N/A
Comments:					

Table D-20: IPX Packet Handling

D.23. MPOA Authentication

Item	MPOA Authentication Is the implementation capable of ...	Conditions for status	Status	Reference	Support
Auth1	carrying an MPOA Authentication Extension in MPOA packets to convey authentication information.		O	5.3.2.4.7	<input type="checkbox"/> Yes <input type="checkbox"/> No
Auth2	sending back an MPOA Error Indication of type "Authentication Failure", if a received MPOA packet fails the authentication test?	Auth1] Auth1	O N/A	5.3.14, 5.3.2.4.7	<input type="checkbox"/> Yes <input type="checkbox"/> No <input type="checkbox"/> N/A
Comments:					

Table D-21: MPOA Authentication

D.24. MPOA MIB

Item	MPOA MIB Is the implementation capable of ...	Conditions for status	Status	Reference	Support
MIB1	supporting the MPOA 1.1 MIB?		O	Annex F	<input type="checkbox"/> Yes <input type="checkbox"/> No

Table D-22: MPOA MIB

Informative Attachment to Annex D: Purpose of a PICS Proforma

This informative Attachment provides further information on the purpose of a PICS Proforma:

To evaluate conformance of a particular implementation, it is necessary to have a statement of which capabilities and options have been implemented with regard to the corresponding specification or standard. Such a statement is called an Implementation Conformance Statement (ICS).

For protocol specifications like MPOA 1.1, this statement is called "Protocol Implementation Conformance Statement" (PICS). For the provision of this statement, a fixed format questionnaire called PICS proforma has to be used. A completed PICS proforma is the PICS for the implementation in question. It is an ICS (as defined in [ISO 9646-7]) for an implementation or system which claims to conform to a given specification.

The PICS can have a number of uses, including:

- by the protocol implementor, as a check list for implementations to reduce the risk of unintended non-conformance, e.g. through oversight;
- by the supplier and acquirer, or potential acquirer, of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the Standard's PICS proforma;
- by the user or potential user of the implementation, as a basis for initially checking the possibility of interworking and the level of interoperability with another implementation - while interoperability can never be guaranteed, failure to interoperate can often be predicted from incompatible PICS
- by a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

The PICS proforma contained in this specification therefore reflect a compromise between these different requirements.

Annex E. MPOA 1.0 Errata List

[Normative]

E.1 Introduction to MPOA Errata

The Errata are organized into sections corresponding to text in the MPOA v1.0 specification [MPOA 1.0] which is in error or is vague. The title of each section references the paragraph or diagram under scrutiny in the MPOA v1.0 specification [MPOA 1.0]. The body of each section includes the proposed corrected text underlining the difference from the original text, when possible.

Section E.3 summarizes various clarifications for MPOA 1.0; section E.4 provides a summary of the corrections to MPOA 1.0 which have been incorporated into MPOA 1.1. Section E.5 shows the corrections made to the [MPOA 1.0 MIB] in Annex F.

E.2. References

The references provided in section 6 above apply.

E.3. Clarifications

E.3.1. Section 2.1 (add an additional clarifying definition of the term "NHRP Purge")

NHRP Purge Within the context of MPOA, this term refers to an MPOA purge mechanism where the packet type values of NHRP purge are used (see sections 5.3.11, 5.3.12 and 5.3.13 below).

E.3.2. Sections 4.2 through 4.2.2 (various editorial clarifications, also on "every time they are sent")

4.2 Device Discovery

Discovery of control addresses of MPOA components is an essential part of the MPOA system. It is necessary for the MPOA Client to know the MAC and ATM addresses of local MPOA Servers so that MPOA Requests may be sent. The MPOA Server must know if an NHRP Request resolves to the ATM address of an MPOA Client so that a cache imposition may be sent. Finally, MPSs sharing an ELAN must be able to discover each other to facilitate the forwarding of NHRP messages. To this end, an MPOA Device Type TLV, defined in Section 5.2.3, is included in the following LANE messages every time they are sent:

- LE_REGISTER_REQUEST
- LE_REGISTER_RESPONSE
- LE_ARP_REQUEST
- LE_ARP_RESPONSE
- Targetless LE_ARP_REQUEST (see 7.1.30 [LANE])

The information carried in the MPOA Device Type TLV includes the type of device (MPS, MPC, MPS/MPC, or non-MPOA), MPS MAC addresses (if the type is MPS), and the respective control addresses. Each MPOA component must register the MPOA Device Type TLV with each LEC on which it is configured.

4.2.1 Register Protocol

If a LEC is being served by an MPOA Server or Client, it must include the MPOA Device Type TLV in the LE_REGISTER_REQUEST Register Request for the relevant MAC addresses. The LEC indicates the type of device in the TLV.

If the LES has no existing entry for the MAC-ATM binding, it must register the new MAC-ATM binding with MPOA Identification information in its address table. However, if the LES has an existing entry for the specified MAC-ATM binding, it must overwrite any existing MPOA Identification with the new MPOA Identification.

If the status of an MPOA device changes, it should request each of its served LECs to send an LE_UNREGISTER_REQUEST Unregister Request to the LES for each registered MAC address. After unregistering, the LEC should send an LE_REGISTER_REQUEST Register Request with the new set of TLVs to the LES for each MAC address.

4.2.2 Address Resolution Protocol

Address resolution requests and replies sent by a LEC with an ATM address that is associated with an MPOA device must include the MPOA Device Type TLV (as described in Section 5.2.3). The TLV will indicate whether the sending MPOA device is an MPOA Server, an MPOA client, or both. A LEC receiving an address resolution request or response must update its MAC-ATM binding entry to reflect the MPOA Identification TLV.

If the status of an MPOA device changes, it should request each of its served LECs to send a TARGETLESS Targetless LE_ARP_REQUEST to the LES for MAC addresses previously included in an address resolution response with the new set of TLVs.

E.3.3.Paragraph 5 of Section 4.2.3

Note that if sharing is used, the list of MPS MAC addresses on the given ELAN will be included in LANE control messages (as defined in Section 4.2) issued for or on behalf of an MPC or non-MPOA MAC address served by that shared LEC ATM address.

E.3.4. 7th Line of 8th (=second to last) Paragraph of Section 4.4.2

The MPOA Resolution Request also must include an MPOA Egress Cache Tag Extension~~TLV~~, should include an MPOA ATM Service Category extension, ...

E.3.5.Section 4.7.1.4, 2nd line (clarification on Data Plane Purge)

"An egress MPC ... must periodically send an MPOA-NHRP Data Plane Purge Request on that shortcut to the ingress MPC ... "

E.3.6.Section 4.7.1.6, Paragraph 1, 2nd line (clarification that action is mandatory)

"This notification ~~allows~~is used by the egress MPS to issue associated NHRP Purge Requests."

E.3.7. Section 4.7.1.6, Paragraph 3,after 1st sentence (addition on no-reply flag)

The no-reply flag (N-bit) is used to indicate whether the egress MPC wishes to receive an MPOA Egress Cache Purge Reply. It is recommended that the no-reply flag is always set by the egress MPC.

E.3.8. Section 4.7.2.2, Ingress MPSs and NHRP Purges

Clarification on various types of NHRP purges; Recommendation not to request an NHRP ingress cache reply:

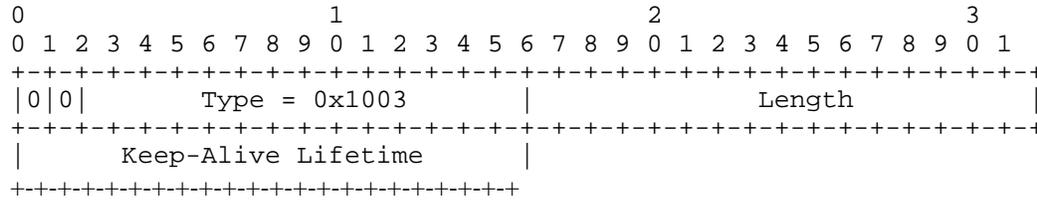
When an ingress MPS receives an NHRP purge request, it must send NHRP ingress cache purge requests to all relevant MPC's for which it is maintaining ingress state for the purged destination address(es).

If the no-reply flag is clear in the received NHRP purge request (meaning an NHRP purge reply is requested), the ingress MPS must respond to the NHRP purge request. The ingress MPS should not use the retry procedure, defined in Section 4.3, to ensure reliable delivery of NHRP ingress cache purge requests to relevant MPCs. If the ingress MPS expects a response from an MPC, it must clear the no-reply flag in its generated NHRP ingress cache purge request, and use the retry procedure. The ingress MPS may send an NHRP purge reply without waiting for NHRP ingress cache purge replies from all MPC's.

E.3.9. MPOA Keep-Alive Lifetime Extension Format of Section 5.3.2.4.4

The diagram implies the length of the Keep-Alive Lifetime field to be 4 bytes when it is actually 2 bytes.

E.3.9.1. Proposed Diagram

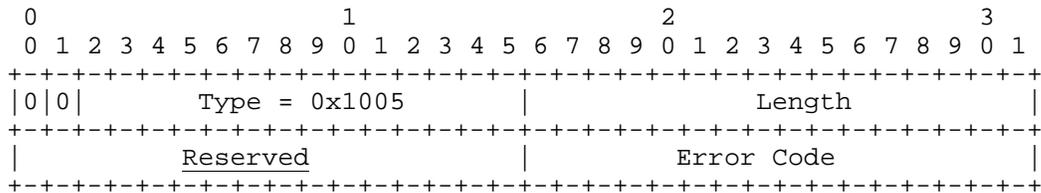


E.3.10. Section 5.3.2.4.6 MPOA Original Error Code Extension

Clarification on the format and use of the extension. Format of Error Code: 16 Bit only.

5.3.2.4.6 MPOA Original Error Code Extension

(...)



There are no MPOA-specific error codes currently defined.

E.3.11. Appendix IV.2, last paragraph (Clarification related to Tag Extension)

The latter option may be used only if the cache imposition includes an MPOA Egress Cache Tag extension TAG TLV indicating that ...

E.4. Corrections

E.4.1. Last Paragraph in Section 3.6

~~Appendix I~~ Appendix II provides example scenarios for the data and control flows.

E.4.2. Figure 7 in Section 4.4

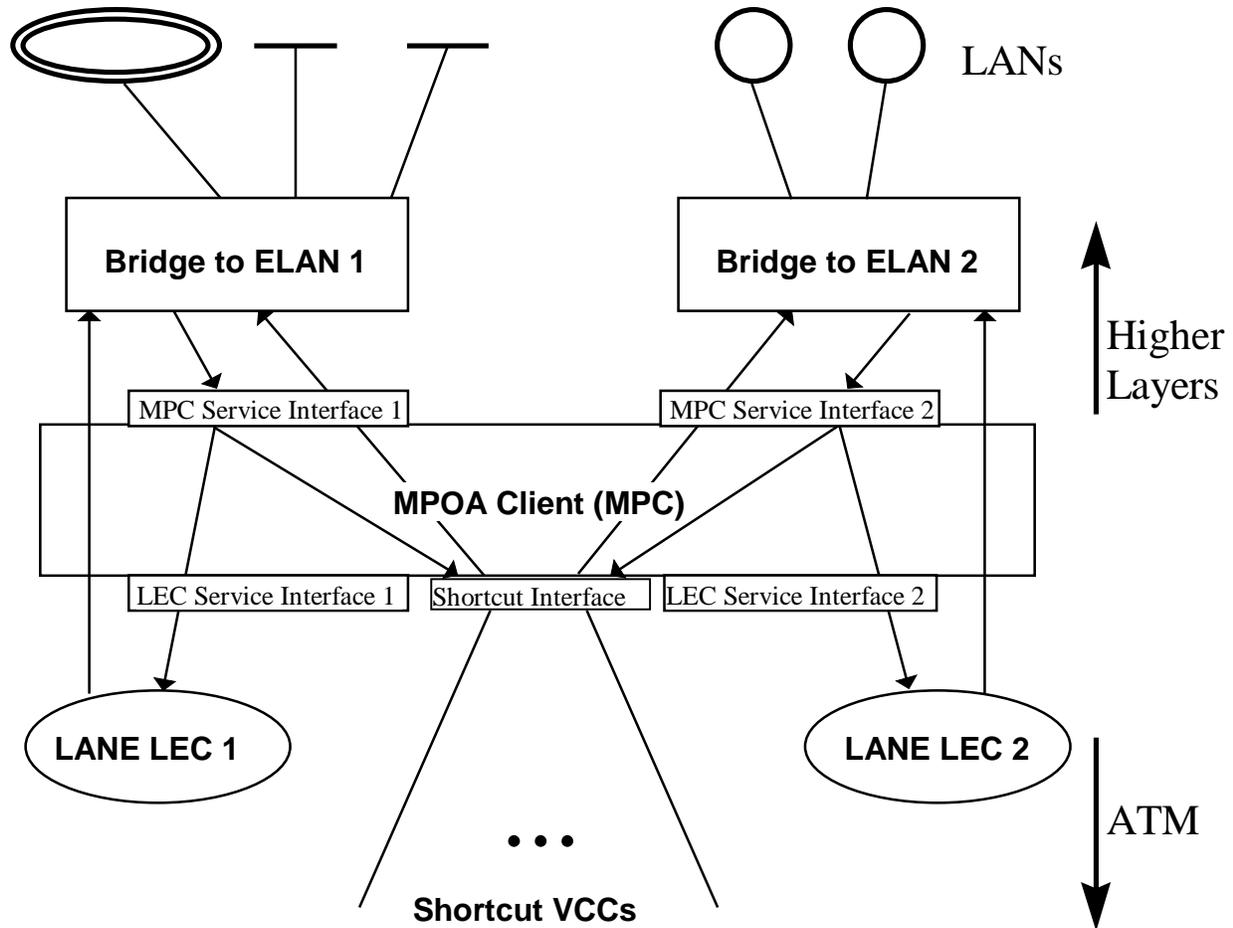


Figure 23 MPOA Edge Device MPC Example

E.4.3. Last Line of Paragraph 1 of Section 4.4.4.1

..., the MPC should use the retry procedure defined in ~~Section 4.2.3~~ Section 4.3.

E.4.4. Last Line of Last Paragraph of Section 4.4.4.2

..., it should initiate a Data Plane Purge as described in ~~Section 4.7.2.2~~ Section 4.7.2.3.

E.4.5.1st Paragraph of Section 4.4.6.1.4

These retries must use the MPOA retry mechanism described in ~~4.2.3~~Section 4.3.

E.4.6.1st Bullet Item in Paragraph 2 of Section 4.4.6.1.4 (replace "should" by "must")

- Request Id - The Request Id that was used in an outstanding request, this ~~should~~ must be reused for subsequent retries.

E.4.7.Paragraph 3 of Section 4.4.6.1.4 (Typo)

While attempting to resolve an address, an MPC may decided at any time that a shortcut ...

E.4.8.Paragraphs 3, 4 and 5 of Section 4.5 - 5 Typos

The MPS must be aware of the router's configuration and forwarding tables to the extent of knowing (or being able to ask the router) whether a given internetwork layer destination address should be forwarded to a LEC, and which one. For each LEC and MAC Address on which the MPS is to be enabled, the MPS supplies the LEC with the MPS's control ATM address. Each LEC so notified includes the MPOA Device Type TLV in every LE_REGISTER_REQUEST, LE_ARP_REQUEST, and LE_ARP_RESPONSE ~~response~~ response that it sends, ~~which~~ to indicates to the recipient that an MPS is connected to the responding LEC and MAC Address, and to indicates the control ATM address of the MPS.

Having advertised its control ATM address via LE_ARP_LANE control messages, an MPS may receive MPOA Resolution Requests from MPCs. In addition, in its capacity as an NHS, the MPS/NHS may receive NHRP queries from NHCs or peer NHSs. The MPS/NHS handles both types of queries.

If the routing and subnet convergence information available to the MPS/NHS indicates that the next hop is ~~an~~ directly connected MPC, then the Resolution Request is passed on to that LEC's MPC as an MPOA Cache Imposition Request. Otherwise, the request should be treated as a standard NHRP Resolution Request and forwarded or answered as indicated in [NHRP].

E.4.9.Section 4.5.2.2

Add to the bulleted list:

- The prefix length is set or adjusted as appropriate.

Note that if a CIE was not present in the Resolution Request, one must be added to send the Holding Time and prefix length.

E.4.10.Paragraph 2 of Section 4.5.2.4 - incomplete specification

When an I-MPS receives an MPOA resolution request for which it is the logical next hop, it has two choices of what to do:

1. NAK the request with an NHRP error code 12: "No Internetworking Layer Address to NBMA Address Binding Exists", and force the packets to be bridged via LANE.

2. Provide a reply using its own data ATM address.

E.4.11.Bulleted List in Section 4.7.1.6

(Ingress MPC Data ATM Address was removed as an option for reducing the scope of the purge; clarifications on mandatory nature of all items)

Information ~~to~~which must be included in an MPOA Egress Cache Purge Request is:

- Request ID
- Egress MPS Protocol Address
- Egress MPC Data ATM Address
- Destination Protocol Address (to purge)
- Destination Prefix Length
- ~~Ingress MPC Data ATM Address (Optional)~~
- MPOA DLL Header Extension (~~Mandatory~~)
- no-reply flag

E.4.12.MPOA Trigger Inconsistency

There is an inconsistency in Sections 4.7.2.1, 5.3.10, and B.5 about whether a CIE is included. We propose that a CIE be required only if the suggestion of a prefix is desired as in an MPOA Resolution Request.

Also, as stated in Section 5.3.2.4 "MPOA control messages may have the same Extensions as an NHRP packet, such as Route Record, NHRP Authentication and Vendor Private Extensions." However, the wording in Sections 5.3.10 and B.5 regarding extensions is ambiguous about whether extensions are prohibited or simply not required. We believe the intent was the latter and suggest a rewording.

This would result in the following changes in the MPOA specification:

E.4.12.1.Last paragraph of Section 4.7.2.1

The protocol address to be used in the triggered MPOA Resolution Request. This address is also required for an ingress MPC to build an ingress cache entry and identify the inbound datagrams that will be sent on the shortcut to be established as a result of a successful receipt of a corresponding NHRP Resolution Reply. The MPS may indicate that an address prefix should be requested by adding a CIE with the prefix set appropriately. The use of prefixes at the MPC is optional, however, and the MPC may issue the resolution request for just the single protocol address.

E.4.12.2.Section 5.3.10

Client Information Element

~~No CIE is used.~~

One CIE may be added in conformance with NHRP.

Field	Usage
Prefix Length	Widest Acceptable prefix length.
MTU	Unused and must be set to zero (0).

Extensions

No extensions are required used.

E.4.12.3. Section B.5 MPOA Trigger

MPOA Triggers operate in a single phase from right to left as follows:

	Ingress MPC	Ingress MPS
Packet Type		MPOA Trigger
Request ID		unused
Source Protocol Address		<u>I-MPS Protocol Address</u> NULL
Destination Protocol Address		<u>Destination Protocol Address (to trigger)</u> NULL
Source NBMA Address		I-MPS Control ATM Address
Client Protocol Address (6)		Destination Protocol Address (to trigger)
Prefix Length (6)		Destination Prefix Length (to trigger)
Client NBMA Address (6)		NULL
Extensions		None

(6) Optional

E.4.13. Paragraph 3 of Section 4.7.2.1 (correct reference)

An ingress MPS must use the MPOA retry procedure defined in ~~Section 4.2.3~~ Section 4.3 to control the sending of MPOA Trigger messages ...

E.4.14. Paragraph 2 of Section 4.7.2.3 - E-MPC not forced to send purges when an MPS fails

Egress MPS Dies:

If an egress MPC fails to receive an MPOA Keep-Alive message from an MPS that has imposed egress cache entries within the MPOA Keep-Alive Lifetime (as specified in the last received MPOA Keep-Alive message) then it may send NHRP Purge Requests which invalidate all the cache entries imposed by the failed MPS, which are currently associated with a shortcut VCC. Alternatively, the egress MPC may invalidate these cache entries locally and send NHRP Purge Requests only when data is received that uses these entries. If there is no open VCC to the source ATM address as specified in an egress cache entry, it is not necessary to establish a VCC for the purpose of sending an NHRP Purge Request.

E.4.15. Paragraph 2 of Section 4.8.9

The use of the signaling IEs is the same as that outlined here for point-to-point calls, subject to the constraints imposed by ~~{UNI 3.1}~~ [UNI 3.0, UNI 3.1 or UNI 4.0].

E.4.16. Paragraph 1 of Section 5.2.3

The MPOA Device Type TLV contains two to five sub-fields, depending on the device, within the value field, as shown below. One sub-field identifies the type of MPOA device (server or client). The other sub-field specifies the ATM address of the MPOA device using UNI 4.0 encoding.

E.4.17. Paragraph 4 of Section 5.3.3 - "Largest" should be "Widest"**Client Information Element**

One CIE may be added in conformance with NHRP.

Field	Usage
Prefix Length	Widest Largest-Acceptable prefix.
MTU	Unused and must be set to zero (0).

E.4.18. Section 5.3.2.4.7 (MPOA Authentication Extension)

New section on MPOA Authentication Extension added.

E.4.19 Section 5.3.7 (MPOA Egress Cache Purge Request Format)

Add / replace in common header:

Field	Usage
Flags	(...) N: No-Reply Flag (recommended to be set)
(...)	(...)

E.4.20 Section 5.3.11, Fixed Header of NHRP Purge (Data Plane)

Editorial / Typo: ar\$op.type is set to 0x05 (NHRP Purge Request)

E.4.21 Paragraph 4 of Section 5.3.11

There is no requirement that only the addresses associated with a given shortcut be purged. Also, the usage of 0x00 defined is inconsistent with the NHRP specification.

Client Information Element

The purge request includes one or more CIEs as follows:

Field	Usage
Code	Set to 0x00.
Prefix Length	Prefix Length in bits for the Client Protocol Address(es) being purged. <u>At least</u> Only the ingress cache entries associated with the shortcut over which the MPOA Data Plane Purge is received are purged. All ingress cache entries for this shortcut must be purged if the Prefix Length is set to 0x00.
Maximum Transmission Unit	Unused.
Holding Time	Unused.
Cli Addr T/L	This field must be set to zero and no storage is allocated for the Client NBMA address at all.
Cli SAddr T/L	This field must be set to zero and no storage is allocated for the Client NBMA Subaddress at all.
Cli Proto Len	The length in octets of the Client Protocol Address.
Client NBMA Address	Egress MPC data ATM address.
Client Protocol Address	The internetwork layer address that is being purged from the ingress MPC's cache.

E.4.22 Sections 5.3.12 and 5.3.13

Missing Sections on NHRP Ingress Cache Purge Request / Reply Formats added.

E.4.23 Section 5.3.14

New section on MPOA Error Indication added.

E.4.24 Section 6 (References) (apply appropriate RFC References for IPOA and NHRP)

[CLIP][IPOA] ~~M. Laubach, RFC 1577, M. Laubach / J. Halpern, RFC 2225, Classical IP and ARP over ATM, January, 1994. April 1998~~

...

[NHRP] ~~See Annex C: J. Luciani, et al., RFC 2332, NBMA Next Hop Resolution Protocol (NRHP), April 1998~~

E.4.25 Note (2) Section B.1

The MPS must always add a CIE for the holding time, so the case described in the second sentence of note (2) can never occur. Delete the second sentence of note (2) as shown below.

(2) An E-MPC can modify the Prefix Length to make it a host entry if a CIE was included in the request. ~~An E-MPC must add a CIE with a host entry if a CIE was not included in the request.~~

E.4.26 Request phase of Egress MPC-Initiated Cache Purge

Corrections / clarifications on the use of E-MPC Protocol Address and of extensions.

E.4.26.1 Section B.2

	Ingress MPC	Ingress MPS	Egress MPS	Egress MPC
Packet Type		NHRP Purge Request	NHRP Purge Request	MPOA Egress Cache Purge Request
Request ID		Request ID 3	Request ID 2	Request ID 1
Source Protocol Address		<u>I</u> E-MPS Protocol Address	E-MPS Protocol Address	<u>E-MPC Protocol Address or NULL</u>
Destination Protocol Address		I-MP <u>S</u> C Protocol Address <u>or NULL</u>	I-MPS Protocol Address	E-MPS Protocol Address
Source NBMA Address		<u>E-MPC I-MPS</u> Data ATM Address	E-MPC Data ATM Address	E-MPC Data ATM Address
Client Protocol Address		Destination Protocol Address (to purge)	Destination Protocol Address (to purge)	Destination Protocol Address (to purge)
Prefix Length		Destination Prefix Length	Destination Prefix Length	Destination Prefix Length
Client NBMA Address		E-MPC Data ATM Address	E-MPC Data ATM Address	E-MPC Data ATM Address
Extensions		Received Extensions	Received Extensions (except for MPOA DLL Header Extension)	MPOA DLL Header Extension (Cache ID) <u>Local Received</u> Extensions

E.4.26.2 Section 4.7.1.6

The MPC-Initiated Egress Cache Purge protocol provides the capability for an egress MPC to notify the egress MPS when it discovers an invalid egress cache entry. This notification allows the egress MPS to issue associated NHRP Purge Requests. The MPC-Initiated Egress Cache Purge Request is most likely used when either the bridge topology has changed and a destination is no longer behind the same edge device, or the destination has aged out of the bridge forwarding table for lack of communication.

Information to be included in an MPOA Egress Cache Purge Request is:

- Request ID
- Egress MPS Protocol Address
- Egress MPC Protocol Address or NULL
- Egress MPC Data ATM Address
- Destination Protocol Address (to purge)
- Destination Prefix Length
- MPOA DLL Header Extension (Mandatory)
- no-reply flag

E.4.27 Reply phase of Egress MPC-Initiated Cache Purge

Errata made for the Egress MPC-initiated Cache Purge Request also imply similar errata in the corresponding reply phase:

	Ingress MPC	Ingress MPS	Egress MPS	Egress MPC
Packet Type	NHRP Purge Reply	NHRP Purge Reply	MPOA Egress Cache Purge Reply	
Request ID	Request ID 3	Request ID 2	Request ID 1	
Source Protocol Address	I E-MPS Protocol Address	E-MPS Protocol Address	<u>E-MPC Protocol Address or NULL</u>	
Destination Protocol Address	I-MPSC Protocol Address <u>or NULL</u>	I-MPS Protocol Address	E-MPS Protocol Address	
Source NBMA Address	E-MPC I-MPS Data ATM Address	E-MPC Data ATM Address	E-MPC Data ATM Address	
Client Protocol Address	Destination Protocol Address (to purge)	Destination Protocol Address (to purge)	Destination Protocol Address (to purge)	
Prefix Length	Destination Prefix Length	Destination Prefix Length	Destination Prefix Length	
Client NBMA Address	E-MPC Data ATM Address	E-MPC Data ATM Address	E-MPC Data ATM Address	
Extensions	Received Extensions	Received Extensions	Received Extensions	

E.4.28 Section B.3 - Egress MPS-Initiated Cache Purge

Errata similar as in Annex B.2 and minor editorials:

	Ingress MPC	Ingress MPS	Egress MPS	Egress MPS	Egress MPC
Packet Type		NHRP Purge Request	NHRP Purge Request	MPOA Cache Imposition Request	
Direction		←	←	→	
Request ID		Request ID 3	Request ID 2	Request ID 1	
Source Protocol Address		I E-MPS Protocol Address	E-MPS Protocol Address	E-MPS Protocol Address	
Destination Protocol Address		I-MPSC Protocol Address Addr or NULL	<u>I-MPS Protocol Address Addr</u>	Destination Protocol Address (to purge)	
Source NBMA Address		E-MPC I-MPS Data ATM Address	E-MPC Data ATM Address	NULL	
Client Protocol Address		Destination Protocol Address (to purge)	Destination Protocol Address (to purge)	NULL	
Prefix Length		Destination Prefix Length	Destination Prefix Length	Destination Prefix Length	
Holding Time				0	

Client NBMA Address		E-MPC Data ATM Address	E-MPC Data ATM Address	NULL	
Extension				MPOA DLL Header Extension (Cache ID) (4)	

The reply phase proceeds as follows:

	Ingress MPC	Ingress MPS	Egress MPS	Egress MPS	Egress MPC
Packet Type	NHRP Purge Reply	NHRP Purge Reply			MPOA Cache Imposition Reply
Direction	→	→			←
Request ID	Request ID 3	Request ID 2			Request ID 1
Source Protocol Address	I-E-MPS Protocol Address	E-MPS Protocol Address			E-MPS Protocol Address
Destination Protocol Address	I-MPCS Protocol Address or NULL	I-MPS Protocol Address			Destination Protocol Address (to purge)
Source NBMA Address	I-MPS E-MPC Data ATM Address	E-MPC Data ATM Address			NULL
Client Protocol Address	Destination Protocol Address (to purge)	Destination Protocol Address (to purge)			NULL
Prefix Length	Destination Prefix Length	Destination Prefix Length			Destination Prefix Length
Client NBMA Address	E-MPC Data ATM Address	E-MPC Data ATM Address			NULL
Extensions					Received Extensions

(4) Optional

E.4.29 Annex C (replace cover sheet and copy of version 11 of the draft NHRP by the RFC reference)

It is proposed to replace the current whole Annex C by just the following sentence and Note:

The NBMA Next Hop Resolution Protocol (NRHP) [NHRP] as specified in IETF RFC 2332 is a mandatory part of this specification.

Note: At the time MPOA 1.0 was specified, a draft version of NHRP had been put into this Annex C as a placeholder. It was made clear that the intention was to replace the copy of that draft by a reference to the RFC version of NHRP as soon as it would be available.

E.4.30 Annex D (PICS Proforma)

New Annex D on PICS Proforma added.

E.4.31 Figure 23 - MPC-c4 through MPC-c7 changed to MPC-c3 through MPC-c6 and add New_MPS event generation

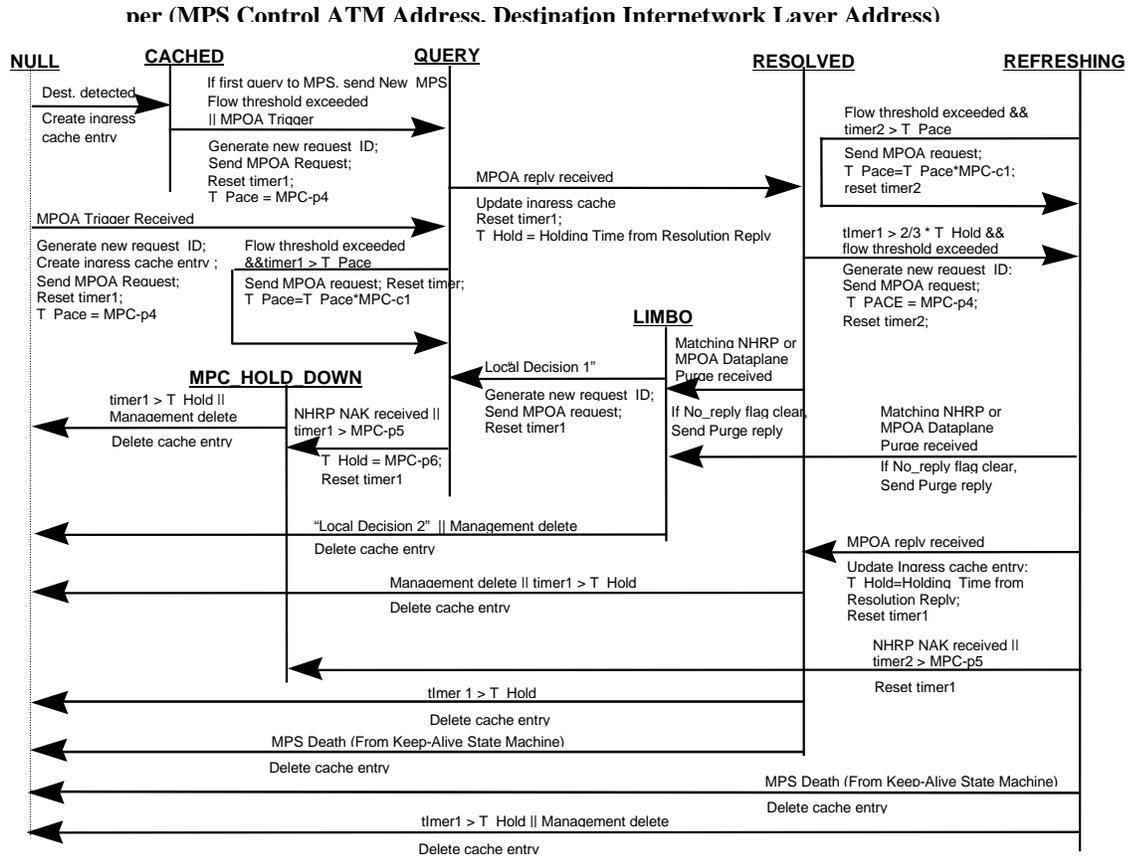


Figure 24 Ingress MPC Control State Machine

E.4.32 Figure 24 - MPS-p4 replaced by MPS-p6 c6 and add New_MPC event generation

per (Ingress MPC Control ATM Address, Request ID)

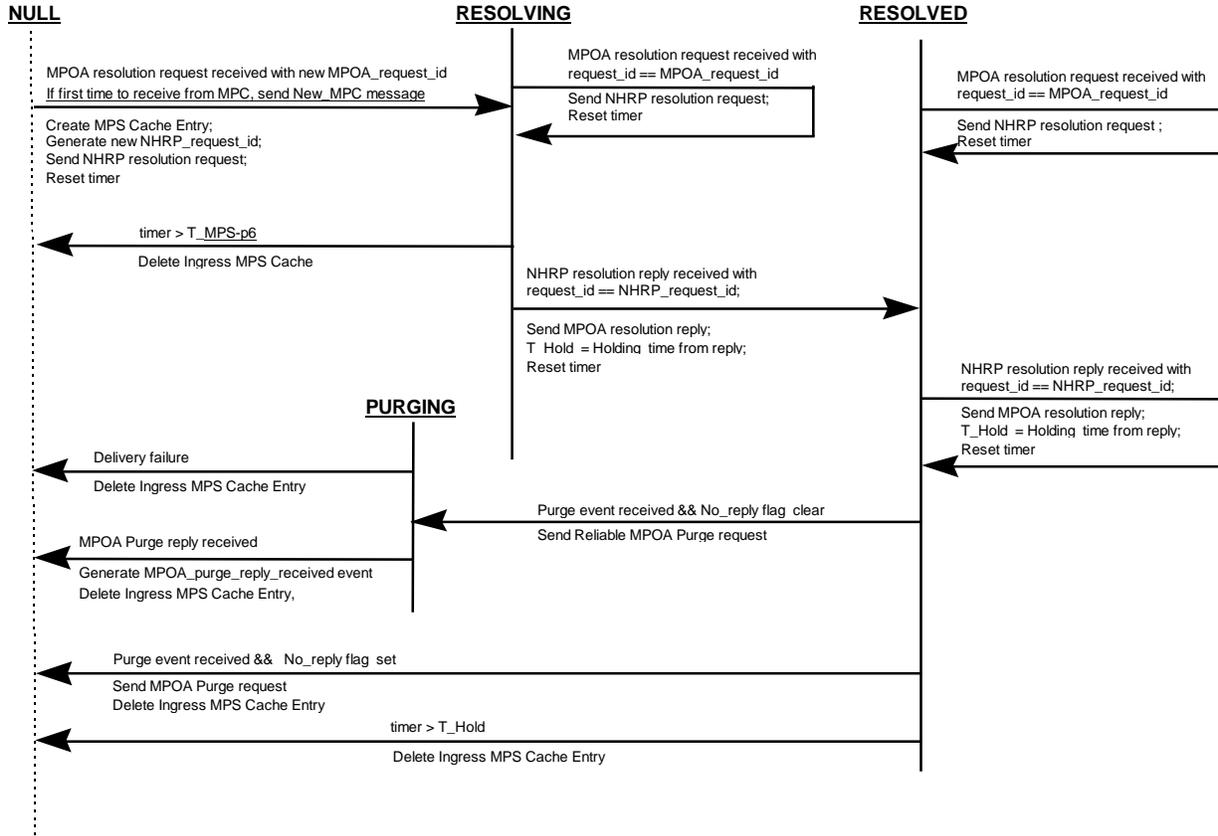
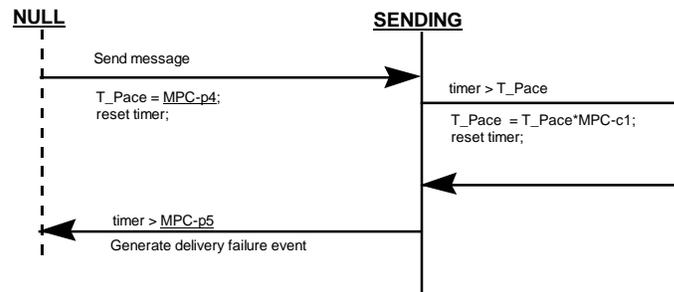


Figure 25 Ingress MPS Control State Machine

E.4.33 Figure 27 - MPC-c4 through MPC-c7 changed to MPC-c3 through MPC-c6

Reliable MPC Message Send (per request)



Reliable MPS Message Send (per request)

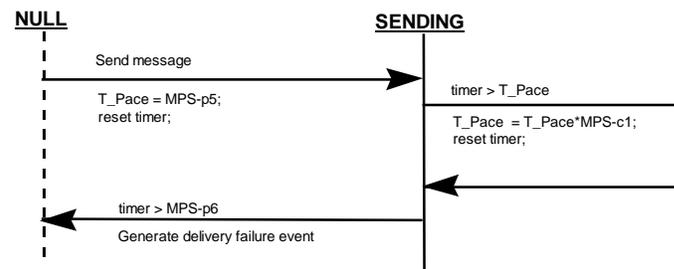


Figure 26 Reliable Delivery State Machines

E.4.34 Apx I, Title I.7 - Applies to both Egress and Ingress ("Egress" removed from beginning)

I.7 Egress MPC and MPS Keep-Alive State Machines

E.4.35 Appendix V - add two more entries

This appendix lists some optional extensions and procedures that an NHC/NHS may wish to implement for more efficient interoperation with MPOA devices. These features are not needed for interoperation between NHRP and MPOA devices, but in some cases efficiency may be improved.

1. support the use of the MPOA egress cache tag extension
2. support the use of the ATM Service Category extension
3. always include a non-zero value for MTU size in a Resolution Reply
4. support CPCS-SDU size negotiation during signalling
5. Use the same address for Source Protocol address in Resolution Request and shortcut VCC.
6. Accept control packets on shortcuts (for example, the data plane purge can be sent to an NHC)

E.4.36 Appendix VII - Tip Sheet, also illustrating corrections

Both Sections of that new Appendix also provide further information on corrections to MPOA 1.0 which have been made. Section 1 of Appendix VII provides further information on corrections to the MPOA MIB (see also section

E.5 below). Section 2 illustrates corrections made on NHRP Ingress Cache Purge, including the addition of the following new last paragraph in section 4.7.2.2:

section 4.7.2.2: new last paragraph:

When processing an NHRP ingress cache purge request, the ingress MPC should not use the received source address information to decide which entries to purge (see also Appendix VII.2).

E.5. Corrections to MPOA 1.0 MIB

E.5.1 Summary of Changes related to MPS Multinetting

(#1-4, changes highlighted by revision bars)

- 1.) The LAST-UPDATED Field and the DESCRIPTION field of the MODULE-IDENTITY statement need to change to reflect the obsolescence of the mpcMpsMacAddressTable and the addition of mpcMpsMultipleMacAddressTable.

mpoaMIB MODULE-IDENTITY

LAST-UPDATED "9811090000Z"

ORGANIZATION "ATM Forum LANE/MPOA Working Group"

CONTACT-INFO

"The ATM Forum

2570 West El Camino Real, Suite 304

Mountain View, CA 94040-1313

USA

Tel: +1-650-949-6700

Fax: +1-650-949-6705

Web: <http://www.atmforum.com>

E-mail: info@atmforum.com"

DESCRIPTION

"This module defines a portion of the management information base (MIB) for managing Multiprotocol Over ATM clients and servers which was revised based on MPOA Errata contained in MPOA v1.1.

The difference between af-mpoa-0092.000 and this version is the mpcMpsMacAddressTable has been obsoleted. The mpcMpsMultipleMacAddressTable has been added. The mpcMpsMultipleMacAddressTable replaces the mpcMpsMacAddressTable."

REVISION "9811090000Z"

DESCRIPTION

"MPOA v 1.1, Nov 9, 1998

Version of the MIB module MPOA-MIB

that is contained in the MPOA v1.1 document.

The difference between af-mpoa-0092.000 and this version is the mpcMpsMacAddressTable has been obsoleted. The mpcMpsMultipleMacAddressTable has been added. The mpcMpsMultipleMacAddressTable replaces the mpcMpsMacAddressTable."

REVISION "9805220000Z"

DESCRIPTION

"Final Ballot Version, May 22, 1998

Version of the MIB module MPOA-MIB

that is in: AF-MPOA-0092.000."

- 2.) The mpcMpsMacAddressTable will have its status changed from "current" to "obsolete". This means the Table

should no longer be supported by any
mpoa mib implementations.

```
|
| -- the following table has been obsoleted, it does
| -- not support multinetted capabilities.
| -- Please see Appendix VII of MPOA 1.1
| -- for a detailed explanation of the
| -- multinetted capabilities.
| -- This table has been replaced by the
| -- mpcMpsMultipleMacAddressTable.
```

```
mpcMpsMacAddressTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MpcMpsMacAddressEntry
    MAX-ACCESS  not-accessible
    STATUS      obsolete
    DESCRIPTION
        "This is a read-only table which contains
        information about all the MPSs' MAC Addresses
        that these MPCs know about."
    ::= { mpcObjects 9 }
```

```
mpcMpsMacAddressEntry OBJECT-TYPE
    SYNTAX      MpcMpsMacAddressEntry
    MAX-ACCESS  not-accessible
    STATUS      obsolete
    DESCRIPTION
        "A row is created by an MPC. The MPC learns about an
        MPS's MAC Address and creates a row."
    INDEX       { mpcMpsIndex,
                 mpcLecIndex
               }
    ::= { mpcMpsMacAddressTable 1 }
```

```
MpcMpsMacAddressEntry ::= SEQUENCE {
    mpcLecIndex      LecIndex,
    mpcMpsMacAddress MacAddress
}
```

```
mpcLecIndex OBJECT-TYPE
    SYNTAX      LecIndex
    MAX-ACCESS  not-accessible
    STATUS      obsolete
    DESCRIPTION
        "The lecIndex which represents the associated LEC."
    ::= { mpcMpsMacAddressEntry 1 }
```

```
mpcMpsMacAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-only
    STATUS      obsolete
    DESCRIPTION
        "The MAC Address of the MPS."
    REFERENCE
        "Multiprotocol Over ATM Version 1.0 (Letter Ballot),
        Section 3.3.3.1"
    ::= { mpcMpsMacAddressEntry 2 }
```

```
|-- end of obsoleted table
```

3.) A new Table, the
mpcMpsMultipleMacAddressTable will be

added. This will replace the mpcMpsMacAddressTable.

```

    mpcMpsObjects OBJECT IDENTIFIER ::= { mpcObjects 16 }

-- The following Table replaces the mpcMpsMacAddressTable (which
-- has been obsoleted.) The following table is more flexible in
-- that it can represent more than one MPS MAC Address used by
-- the MPC during flow detection.

mpcMpsMultipleMacAddressTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MpcMpsMultipleMacAddressEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This is a read-only table which contains
        information about all the MPSs' MAC Addresses
        that these MPCs use during flow detection.
        Note that due to the multinetted case an MPC may learn
        about more than one MAC address from an MPS,
        thus there may be more than one MAC address for the
        same MPC - MPS - LecIndex represented in this
        Table. These MacAddresses are differentiated by
        the mpcMpsMacAddressIndex."
    ::= { mpcMpsObjects 1 }

mpcMpsMultipleMacAddressEntry OBJECT-TYPE
    SYNTAX      MpcMpsMultipleMacAddressEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A row is created by an MPC. The MPC learns about a
        MPS and the one or more MAC Address of the MPS which
        the MPC uses during flow detection. Each row represents
        an MPS MAC Address used by an MPC during flow detection."
    INDEX      {
        mpcMpsIndex,
        mpcFlowDetectLecIndex,
        mpcMpsMacAddressIndex
    }
    ::= { mpcMpsMultipleMacAddressTable 1 }

MpcMpsMultipleMacAddressEntry ::= SEQUENCE {
    mpcFlowDetectLecIndex      LecIndex,
    mpcMpsMacAddressIndex      Integer32,
    mpcMpsFlowDetectMacAddress MacAddress
}

mpcFlowDetectLecIndex OBJECT-TYPE
    SYNTAX      LecIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The lecIndex which represents the associated LEC."
    ::= { mpcMpsMultipleMacAddressEntry 1 }

mpcMpsMacAddressIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This value is used to differentiate MAC Addresses from

```

```

        the same MPS used by the same MPC during flow detection.
        This value should be unique within the scope of this table."
 ::= { mpcMpsMultipleMacAddressEntry 2 }

mpcMpsFlowDetectMacAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "An MPS MAC Address used by an MPC during flow detection."
 ::= { mpcMpsMultipleMacAddressEntry 3 }

```

- 4.) In the Conformance Statements, the STATUS of mpcMpsMacAddressGroup need to be changed to obsolete.

```

mpcMpsMacAddressGroup OBJECT-GROUP
    OBJECTS {
        mpcMpsMacAddress
    }
    STATUS       obsolete
    DESCRIPTION
        "A collection of objects which aid the MPCs to track
        MAC Address information for all the MPSs which are known
        by the MPCs."
 ::= { mpoaMIBGroups 10 }

```

And replaced by:

```

mpcMpsMultipleMacAddressGroup OBJECT-GROUP
    OBJECTS {
        mpcMpsFlowDetectMacAddress
    }
    STATUS       current
    DESCRIPTION
        "A collection of objects which aid the MPCs to track
        MAC Address information for all the MPSs which are
        used during flow detection by the MPCs."
 ::= { mpoaMIBGroups 25 }

```

NOTE: also every occurrence of mpcMpsMacAddressGroup in the Conformance statement now also has an occurrence of the mpsMpsMultipleMacAddressGroup.

E.5.2 Correction of a Syntax Error for Index Item "mpsEgressCacheId"

E: f(mpoa.mib), (3428,24) Index item "mpsEgressCacheId" must be defined with syntax that includes a range

The fix is to change this INDEX variable from Integer32 to

```
SYNTAX INTEGER (1..2147483647)
```

This would NOT result in anything other than changing the index from

SYNTAX of Integer32 to INTEGER(1..2147483647)
This restricts the use of number 0, (zero) as an index.

But does NOT change the semantics of the MIB.

Annex F: MPOA 1.1 MIB

Note: The MPOA 1.1 MIB specified in this Annex is believed to be correct at the time of its specification. In case that inconsistencies or errors are detected, a subsequent separate Errata document or corrected version of this MIB may be specified which would take precedence over this Annex. Implementors of this MIB are therefore encouraged to check whether such an Errata document or update of the MPOA 1.1 MIB exists.

```
MPOA-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
MODULE-IDENTITY, OBJECT-TYPE,
enterprises, Counter32, Counter64, Integer32
FROM SNMPv2-SMI
TEXTUAL-CONVENTION,
MacAddress, TimeInterval, TimeStamp, TruthValue, RowStatus
FROM SNMPv2-TC
MODULE-COMPLIANCE, OBJECT-GROUP
FROM SNMPv2-CONF
lecIndex
FROM LAN-EMULATION-CLIENT-MIB
AtmAddr
FROM ATM-TC-MIB
;
```

```
mpoaMIB MODULE-IDENTITY
```

```
LAST-UPDATED "9811090000Z"
ORGANIZATION "ATM Forum LANE/MPOA Working Group"
CONTACT-INFO
"The ATM Forum
2570 West El Camino Real, Suite 304
Mountain View, CA 94040-1313
USA

Tel: +1-650-949-6700
Fax: +1-650-949-6705

Web: http://www.atmforum.com
E-mail: info@atmforum.com"
```

```
DESCRIPTION
```

```
"This module defines a portion of the management
information base (MIB) for managing Multiprotocol Over
ATM clients and servers which was revised based
on MPOA Errata contained in MPOA v1.1.
```

```
The difference between af-mpoa-0092.000 and this
version is the mpcMpsMacAddressTable has been
obsoleted. The mpcMpsMultipleMacAddressTable
has been added. The mpcMpsMultipleMacAddressTable
replaces the mpcMpsMacAddressTable."
```

```
REVISION "9811090000Z"
```

```
DESCRIPTION
```

```
"MPOA v 1.1, Nov 9, 1998
Version of the MIB module MPOA-MIB
that is contained in the MPOA v1.1 document.
```

```
The difference between af-mpoa-0092.000 and this
version is the mpcMpsMacAddressTable has been
```

obsoleted. The mpcMpsMultipleMacAddressTable has been added. The mpcMpsMultipleMacAddressTable replaces the mpcMpsMacAddressTable."

REVISION "9805220000Z"

DESCRIPTION

"Final Ballot Version, May 22, 1998
Version of the MIB module MPOA-MIB
that is in: AF-MPOA-0092.000."

REVISION "9802250000Z"

DESCRIPTION

"Straw Ballot Revision 1.0, February 25, 1998

Version of the MIB module MPOA-MIB
that is in STR-MPOA-MIB-01.01."

::= { atmfMpoa 1 }

atmForum OBJECT IDENTIFIER ::= { enterprises 353 }

atmForumNetworkManagement OBJECT IDENTIFIER ::= { atmForum 5 }

atmfMpoa OBJECT IDENTIFIER ::= { atmForumNetworkManagement 8 }

--

-- Textual Conventions

--

LecIndex ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The value of this object identifies the LEC for which the entry contains management information. The value of this object for a particular LAN Emulation Client (LEC) has the same value as the lecIndex object, defined in the LAN-EMULATION-CLIENT MIB, for the same LEC."

SYNTAX INTEGER (1..2147483647)

AtmConfigAddr ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The ATM address used by the network entity. The address types are: NSAP SEL Byte (1 octet) E.164 (8 octets), and NSAP (20 octets). Note: If the 1 octet NSAP SEL is given, the other 19 octets of the NSAP are derived from the system either through ILMI or another method. Note: The E.164 address is encoded in BCD format."

SYNTAX OCTET STRING (SIZE(1|8|20))

InternetworkAddrType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Internetwork Layer Address Types. Values are defined in Assigned Numbers, RFC1700. Note: not all of these values make sense in all contexts where this type is used in this MIB, but they are included for completeness."

REFERENCE

"Assigned Numbers, RFC1700, ADDRESS FAMILY NUMBERS"

SYNTAX INTEGER {
other(0),

```

        ipV4(1),
        ipV6(2),
        nsap(3),
        hdlc(4),
        bbn1822(5),
        ieee802(6),
        e163(7),
        e164(8),
        f69(9),
        x121(10),
        ipx(11),
        appleTalk(12),
        decnetIV(13),
        banyanVines(14),
        e164WithNsap(15)
    }

InternetNetworkAddr ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION
        "The value of an internetwork layer address."
    SYNTAX          OCTET STRING (SIZE (0..60))

MpcIndex ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION
        "A unique value, for each MPOA client
        which this SNMP agent manages.  It is recommended that
        values
        are assigned contiguously starting from 1.  The value for
        each MPOA Client must remain constant, even if the MPOA
        Client or SNMP agent is re-initialized."
    SYNTAX          Integer32 (1..2147483647)

MpsIndex ::= TEXTUAL-CONVENTION
    STATUS          current
    DESCRIPTION
        "A unique value, for each MPOA Server
        which this SNMP agent manages.  It is recommended that
        values
        are assigned contiguously starting from 1.  The value for
        each MPOA Server must remain constant, even if the MPOA
        Server or SNMP agent is re-initialized."
    SYNTAX          Integer32 (1..2147483647)

mpoaMIBObjects OBJECT IDENTIFIER ::= { mpoaMIB 1 }

-- This MIB module consists of the following groups:
--
-- (1) MPOA Common Groups
--     (a) Device Type group
--     (b) Device Type Mps Mac group
--
-- (2) MPOA Client Groups
--     (a) Configuration group
--     (b) Actual group
--     (c) Data Atm Address group
--     (d) Statistics group
--     (e) Protocol support group
--     (f) LEC -> MPC Mapping group
--     (g) MPC's MPS Information group
--     (h) MAC Address group

```

```

--      (i) Ingress Cache Total Packet group
--      (j) Ingress Cache Total Octet group
--      (k) Ingress Cache group
--      (l) Egress Cache Total Packet group
--      (m) Egress Cache Total Octet group
--      (n) Egress Cache group
--
--      (3) MPOA Server groups
--      (a) Configuration group
--      (b) Actual group
--      (c) Statistics group
--      (d) Protocol support group
--      (e) LEC -> MPS Mapping group
--      (f) Ingress Cache group
--      (g) Egress Cache group
--

```

```

-----
mpoaCommonObjects OBJECT IDENTIFIER ::= { mpoaMIBObjects 1 }

```

```

deviceTypeTable OBJECT-TYPE

```

```

SYNTAX      SEQUENCE OF DeviceTypeEntry

```

```

MAX-ACCESS      not-accessible

```

```

STATUS      current

```

```

DESCRIPTION

```

```

"The device type table
represents the mapping of the Lane Data ATM address
to the MAC device capability.
The unique key is the Lane data ATM address and Lec Index
of the LEC associated with the MAC addresses.
This table contains information which was
gathered from its environment about neighboring
machines.
This Device type table represents the information of
other/remote MPOA devices, discovered/gathered by each MPOA
device. This MIB is not the device type of MPS/MPC
itself; it is DISCOVERED information."

```

```

REFERENCE

```

```

"Sections: 5.2.3 Device Type TLV, and 4.2 Device Discovery,
MPOA Version 1.0 (Letter Ballot) AF-MPOA-0087.000"

```

```

::= { mpoaCommonObjects 1 }

```

```

deviceTypeEntry OBJECT-TYPE

```

```

SYNTAX      DeviceTypeEntry

```

```

MAX-ACCESS      not-accessible

```

```

STATUS      current

```

```

DESCRIPTION

```

```

"An entry in this table and corresponding entries in the
deviceTypeMpsMacAddressTable represent a mapping of a MAC device
capability (i.e. the set of MAC addresses from a device) to the
LEC ATM Address."

```

```

REFERENCE

```

```

"Section 5.2.3 Device Type TLV
MPOA Version 1.0 (Letter Ballot) AF-MPOA-0087.000"

```

```

INDEX      { deviceTypeIndex }

```

```

::= { deviceTypeTable 1 }

```

```

DeviceTypeEntry ::= SEQUENCE {

```

```

deviceTypeIndex          INTEGER,
deviceTypeLecIndex      LecIndex,
deviceTypeRemoteLecAtmAddress AtmAddr,
deviceTypeType          INTEGER,
deviceTypeMpsAtmAddress AtmAddr,
deviceTypeMpcAtmAddress AtmAddr
}

deviceTypeIndex OBJECT-TYPE
SYNTAX      INTEGER (1..2147483647)
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "Index into this table and also used as one of the indices for
    the deviceTypeMpsMacAddressTable.  This index has local
    significance within the mpoaDeviceGroup.
    Entries in the 'deviceTypeMpsMacAddressTable' which correspond
    to this index, and have the 'deviceTypeType' value of 'mps' or
    'mpsAndMps' are considered to be MPS MAC addresses."
REFERENCE
    "Section 5.2.3 Device Type TLV
    MPOA Version 1.0 (Letter Ballot) AF-MPOA-0087.000"
 ::= { deviceTypeEntry 1 }

deviceTypeLecIndex OBJECT-TYPE
SYNTAX      LecIndex
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "LecIndex of LEC that supports this
    data ATM address"
REFERENCE
    "Section 5.2.3 Device Type TLV
    MPOA Version 1.0 (Letter Ballot) AF-MPOA-0087.000"
 ::= { deviceTypeEntry 2 }

deviceTypeRemoteLecAtmAddress OBJECT-TYPE
SYNTAX      AtmAddr
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The ATM address learned by LE ARP."
REFERENCE
    "Section 5.2.3 Device Type TLV
    MPOA Version 1.0 (Letter Ballot) AF-MPOA-0087.000"
 ::= { deviceTypeEntry 3 }

deviceTypeType OBJECT-TYPE
SYNTAX      INTEGER {
                nonMpoa(1),
                mps(2),
                mpc(3),
                mpsAndMpc(4)
            }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "same as the TLV"
REFERENCE
    "Section 5.2.3 Device Type TLV
    MPOA Version 1.0 (Letter Ballot) AF-MPOA-0087.000"
 ::= { deviceTypeEntry 4 }

```

```

deviceTypeMpsAtmAddress OBJECT-TYPE
    SYNTAX      AtmAddr
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Associated MPS address, zeros for non-MPOA and mpc"
    REFERENCE
        "Section 5.2.3 Device Type TLV
        MPOA Version 1.0 (Letter Ballot) AF-MPOA-0087.000"
    ::= { deviceTypeEntry 5 }

deviceTypeMpcAtmAddress OBJECT-TYPE
    SYNTAX      AtmAddr
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Associated MPC address, zeros for non-MPOA & mps"
    REFERENCE
        "Section 5.2.3 Device Type TLV
        MPOA Version 1.0 (Letter Ballot) AF-MPOA-0087.000"
    ::= { deviceTypeEntry 6 }

-- The deviceTypeMpsMacAddress Table contains MAC addresses
-- from the device type TLV.  If the deviceTypeType was 'mpsAndMpc'
-- there must be at least one MPS MAC Address (i.e. at least one entry in
-- this table.)  If the deviceTypeType is 'mps', there may be zero or more
-- MPS MAC addresses in this table.  If the deviceTypeType is 'nonMpoa'
-- or 'mpc' then there will be no corresponding entries in this table.
-- See Section 5.2.3 of the MPOA Letter Ballot 1.0

deviceTypeMpsMacAddressTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DeviceTypeMpsMacAddressEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains MPS MAC Address information gathered from
        the MPOA DeviceTypeTLV."
    REFERENCE
        "Section 5.2.3 Device Type TLV
        MPOA Version 1.0 (Letter Ballot) AF-MPOA-0087.000"
    ::= { mpoaCommonObjects 2 }

deviceTypeMpsMacAddressEntry OBJECT-TYPE
    SYNTAX      DeviceTypeMpsMacAddressEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Each entry represents an MPS MAC Address.  Each entry
        corresponds to a deviceTypeIndex value for which the
        deviceTypeType attribute is 'mps' or 'mpsAndMpc'."
    REFERENCE
        "Section 5.2.3 Device Type TLV
        MPOA Version 1.0 (Letter Ballot) AF-MPOA-0087.000"
    INDEX      { deviceTypeIndex,
                deviceTypeMpsMacAddress
              }
    ::= { deviceTypeMpsMacAddressTable 1 }

DeviceTypeMpsMacAddressEntry ::=SEQUENCE {
    deviceTypeMpsMacAddress      MacAddress
}

```

```

deviceTypeMpsMacAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "MPS MAC address contained in the Device Type TLV which
        is identified by the deviceTypeIndex in the deviceTypeTable."
    REFERENCE
        "Section 5.2.3 Device Type TLV
        MPOA Version 1.0 (Letter Ballot) AF-MPOA-0087.000"
    ::= { deviceTypeMpsMacAddressEntry 1 }

--
-- MPOA Client Objects
--

mpcObjects OBJECT IDENTIFIER ::= { mpoaMIBObjects 2 }

--
-- MPOA Client configuration group
--

mpcNextIndex OBJECT-TYPE
    SYNTAX      Integer32 (0..2147483647)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "This object contains an appropriate value to
        be used for mpcIndex when creating entries in
        the mpcConfigTable.  The value 0 indicates that
        no new rows can be created.  Otherwise, it is
        recommended that values are assigned contiguously,
        starting from 1.

        MPC creation by a Manager:  To obtain the mpcIndex value
        for a new entry, the manager issues a management protocol
        retrieval operation to obtain the current value of this
        object.  If the value retrieved is 0 (zero), the manager
        cannot create a row.

        After each retrieval of a non-zero value, the manager
        should issue a management protocol SET operation using
        the value just retrieved.  If the SET is successful,
        the agent should update the value to the next unassigned
        index, or zero if appropriate.

        NOTE:  the manager may also issue a set on this object
        with a value of its own choosing.  If the set is successful,
        the manager may use this value for the mpcIndex.  In this
        case, the agent would update the value to the next
        unassigned index, or zero if appropriate.
        The definition of 'next unassigned index' is any
        mpcNextIndex value that has not yet been set by a manager,
        or reserved by the agent (see next paragraph), since this
        agent was last re-initialized.

        MPC creation by an Agent:  When a row in the
        mpcConfigTable is created by an agent, the agent should
        reserve the value of the index by updating the value of
        this object to the next unassigned index
        or zero if appropriate.  Thus, a manager will not

```

be able to set an index reserved by an agent.

In the situation of an agent re-initialization, all currently used mpcIndexes must be preserved.

In other words, the Agent should store in non-volatile memory all of the currently used mpcIndexes (along with all necessary configuration information from the mpcConfigTable). When the agent is re-initialized, the mpcNextIndex value is any valid Integer32 value which is not being used as an mpcIndex, except 0 which maintains its original definition of indicating that a row cannot be created."

```
::= { mpcObjects 1 }
```

```
mpcConfigTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MpcConfigEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The MPOA Client Configuration Table.
        This table contains configuration information
        for all MPOA Clients which this agent manages."
    ::= { mpcObjects 2 }
```

```
mpcConfigEntry OBJECT-TYPE
    SYNTAX      MpcConfigEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "MPOA Client Configuration Entry.
        Each entry contains configuration information
        for one MPOA Client. The configuration information, including
        the mpcIndex, must be restored after a re-initialization of an
        MPC or a re-initialization of the SNMP agent."
    INDEX       { mpcIndex }
    ::= { mpcConfigTable 1 }
```

```
MpcConfigEntry ::=SEQUENCE {
    --
    -- Primary config info: Index, mode and control address information
    --
    mpcIndex          MpcIndex,
    mpcRowStatus      RowStatus,
    mpcConfigMode     INTEGER,
    mpcCtrlAtmAddr    AtmConfigAddr,
    --
    -- MPC parameters which may be obtained from
    -- the LECS.
    --
    mpcSCSetupFrameCount Integer32,    -- MPC-p1
    mpcSCSetupFrameTime  Integer32,    -- MPC-p2
    -- The Flow-detection Protocols (denoted with MPC-p3)
    -- are represented in the mpcProtocolsTable.
    mpcInitialRetryTime  Integer32,    -- MPC-p4
    mpcRetryTimeMaximum  Integer32,    -- MPC-p5
    mpcHoldDownTime     Integer32,    -- MPC-p6
}
}
```

```
mpcIndex OBJECT-TYPE
    SYNTAX      MpcIndex
    MAX-ACCESS  not-accessible
```

```

STATUS      current
DESCRIPTION
    "A value which uniquely identifies this conceptual
    row in the mpcConfigTable. The 'mpcNextIndex' object
    needs to be used to determine the value of this object.
    A row cannot be added, unless the mpcCtrlAtmAddress is unique.
    In the event of either an MPC re-initialization or an agent
    re-initialization, the value of this mpcIndex must remain the
    same. In other words, the row needs to be saved and restored
    in the event of an MPC or SNMP Agent re-initialization."
 ::= { mpcConfigEntry 1 }

mpcRowStatus OBJECT-TYPE
SYNTAX      RowStatus
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "This object allows creation and deletion of
    MPOA Clients.

    Within each conceptual mpcConfigTable row, writable objects
    may be modified, regardless of the value of mpcRowStatus.
    It is not necessary to set a row's status to 'notInService'
    first.

    A row cannot be created, unless the mpcAtmCtrlAddress
    in this table is unique.
    When an MPOA Client is created via this object,
    it will initially have
    'mpcActualState' = 'initialState'"
 ::= { mpcConfigEntry 2 }

mpcConfigMode OBJECT-TYPE
SYNTAX      INTEGER {
                automatic(1),
                manual(2)
            }
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "Indicates whether this MPC should auto-configure
    the next time it is (re-)initialized.
    During the (re-)initialization of this MPC, if
    the mode is automatic(1), the LECS is contacted and
    requests are made for all MPC-p* parameters.
    Otherwise, if the mode is manual(2), the values of the
    configuration parameters are obtained from the
    mpcConfigTable and the mpcProtocolTable. In other words, when
    the MPC is first initialized, it can use the default or
    configured values from the mpcConfigTable and
    mpcProtocolTable. If the mode is manual (2), no further
    action is required. If the mode is automatic (1), then
    the LECS should be contacted and all available MPC-p1
    to MPC-p6 parameters would be retrieved.
    These parameters would then overwrite the existing MPC-p1 to
    MPC-p6 parameters. The actual values are reflected in the
    mpcActualTable."
DEFVAL { automatic }
 ::= { mpcConfigEntry 3 }

mpcCtrlAtmAddr OBJECT-TYPE
SYNTAX      AtmConfigAddr

```

```

MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "The MPC's Control ATM Address.  There exists one
    Control ATM Address per MPC, therefore, the value
    of this entry is unique within the table.  The control ATM
    Address is the address which is used by the MPC in its requests
    to the MPS.  The value of this object should not change, once
    created."
 ::= { mpcConfigEntry 4 }

mpcSCSetupFrameCount OBJECT-TYPE
SYNTAX          Integer32 (1..65535)
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "This represents the MPC-p1 Short-cut setup frame count
    parameter.

    The MPC-p1 value is frames measured over mpcSCFrameTime seconds.
    Flow detection is protocol independent. i.e. all network layers
    mpcProtocolEntries for this MPC share the flow rate
    specification.
    A value of 1 causes all flows to initiate resolution/shortcut
    process."
DEFVAL { 10 }
 ::= { mpcConfigEntry 5 }

mpcSCSetupFrameTime OBJECT-TYPE
SYNTAX          Integer32 (1..60)
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "MPC-p2 Short-cut setup frame time, in seconds."
DEFVAL { 1 }
 ::= { mpcConfigEntry 6 }

mpcInitialRetryTime OBJECT-TYPE
SYNTAX          Integer32 (1..300)
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "MPC-p4 is the initial value for the retry time out
    period used for timing out MPOA Resolution Requests
    in seconds.  Retry time consists of this initial
    time-out (MPC-p4) and a retry multiplier (MPC-c1).
    If a response is not received, then another request
    is sent with a timeout of 'retry time' * MPC-c1
    seconds, or until mpcRetryTimeMaximum."
DEFVAL { 5 }
 ::= { mpcConfigEntry 7 }

mpcRetryTimeMaximum OBJECT-TYPE
SYNTAX          Integer32 (10..300)
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "MPC-p5 cumulative max value for Retry Time (MPC-p4).
    Retries are attempted at intervals determined by the
    algorithm described in the definition of mpcInitialRetryTime."
REFERENCE
    "Multiprotocol Over ATM.  AF-MPOA-0087.000."

```

```

        Section 4.3 MPOA Retry Mechanism"
DEFVAL { 40 }
::= { mpcConfigEntry 8 }

mpcHoldDownTime OBJECT-TYPE
SYNTAX      Integer32 (30..1200)
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "MPC-p6 Hold Down Time
    Minimum time to wait before reinitiating
    a failed resolution attempt. Default is
    mpcRetryTimeMaximum * 4."
REFERENCE
    "Multiprotocol Over ATM. AF-MPOA-0087.000.
    Section 4.1.2.1 MPC Parameters"
DEFVAL { 160 }
::= { mpcConfigEntry 9 }

--
-- MPOA Client Actual group
--

mpcActualTable OBJECT-TYPE
SYNTAX      SEQUENCE OF MpcActualEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "MPOA Client Actual Table.
    A read-only table which contains state
    information and reflects the actual values which
    these MPOA Clients are using. The actual
    values may differ from the configured values.
    For example, the mpcConfigMode takes affect
    only during (re-)initialization of the MPC.
    The MPC-p1 to MPC-p6 parameters may differ
    from the configured values because, if the
    MPC was (re-)initialized and the mpcConfigMode
    was set to automatic (1) then some, perhaps all,
    of the MPC-p1 to MPC-p6 parameters were retrieved
    from the LECS and the values from the LECS may
    differ from the configured/default values of the
    mpcConfigTable. NOTE: after re-initialization
    a set to an object in the mpcConfigTable which
    changes the value of the object will be reflect
    in this Table, except for a set to the
    mpcConfigMode which takes effect only during
    re-initialization."
::= { mpcObjects 3 }

mpcActualEntry OBJECT-TYPE
SYNTAX      MpcActualEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "An entry in the MPC Actual Table.
    An entry represents a specific MPOA Client's
    state information and the actual values
    which are being used by the MPOA Client.
    For example, the corresponding mpcConfigEntry contains
    default and/or configured parameters, if mpcConfigMode was
    set to manual, then these are the objects values' which are

```

reflected for the mpcActualEntry. However, if the mpcConfigMode is automatic, then the mpcActualEntry will be all the corresponding mpcConfigEntry's object, excluding any objects which were retrieved from the LECS. In other words, the objects retrieved from the LECS during the (re-)initialization of the MPC overwrite any of the default and/or configured values.

NOTE: any subsequent 'set' to the configured values, e.g. an SNMP

set operation, which is successful could result in a change to an mpcConfigTable value, and will be reflected in this table as well."

```
AUGMENTS { mpcConfigEntry }
 ::= { mpcActualTable 1 }
```

```
MpcActualEntry ::=SEQUENCE {
  mpcActualState          INTEGER,
  mpcDiscontinuityTime   TimeStamp,
  --
  -- Actual values for the MPCs.
  --
  mpcActualConfigMode     INTEGER,
  mpcActualSCSetupFrameCount Integer32,  -- MPC-p1
  mpcActualSCSetupFrameTime Integer32,  -- MPC-p2
  -- The flow-detection protocols for MPC-p3 are represented
  -- in the mpcProtocolTable. There is no actual counterpart
  -- for them.
  mpcActualInitialRetryTime Integer32,  -- MPC-p4
  mpcActualRetryTimeMaximum Integer32,  -- MPC-p5
  mpcActualHoldDownTime   Integer32    -- MPC-p6
}
```

```
mpcActualState OBJECT-TYPE
SYNTAX      INTEGER {
              unknown(1),
              initialState(2),
              up(3),
              down(4)
            }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "This object indicates the actual state
  of the MPOA Client."
 ::= { mpcActualEntry 1 }
```

```
mpcDiscontinuityTime OBJECT-TYPE
SYNTAX      TimeStamp
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
  "The value of sysUpTime on the most recent occasion at
  which any one or more of this MPC's counters
  experienced a discontinuity. The relevant counters
  are the specific instances associated with this
  MPC. If discontinuities have not occurred since the
  last re-initialization of the local management subsystem,
  then this object contains a zero value."
 ::= { mpcActualEntry 2 }
```

```
mpcActualConfigMode OBJECT-TYPE
```

```

SYNTAX      INTEGER {
                automatic(1),
                manual(2)
            }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Indicates whether this MPC auto-configured
    when it was last (re-)initialized."
 ::= { mpcActualEntry 3 }

mpcActualSCSetupFrameCount OBJECT-TYPE
SYNTAX      Integer32 (1..65535)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "MPC-p1 Short-cut setup frame count.
    In frames measured over mpcShortcutFrameTime seconds.
    Flow detection is protocol independent. i.e. all
    network layers mpcProtocolEntry share the flow rate
    specification.
    A value of 1 implies that resolutions for short-cuts
    are attempted for all flows."
 ::= { mpcActualEntry 4 }

mpcActualSCSetupFrameTime OBJECT-TYPE
SYNTAX      Integer32 (1..60)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Actual MPC-p2 Short-cut setup frame time, in seconds."
 ::= { mpcActualEntry 5 }

mpcActualInitialRetryTime OBJECT-TYPE
SYNTAX      Integer32 (1..300)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Actual MPC-p4 is initial value for the
    retry time out."
 ::= { mpcActualEntry 6 }

mpcActualRetryTimeMaximum OBJECT-TYPE
SYNTAX      Integer32 (30..300)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "MPC-p5 cumulative maximum value for Retry Time (MPC-p4).
    Retries are attempted at intervals determined by the
    algorithm described in the definition of
    mpcActualInitialRetryTime."
REFERENCE
    "Multiprotocol Over ATM. AF-MPOA-0087.000.
    Section 4.3 MPOA Retry Mechanism"
 ::= { mpcActualEntry 7 }

mpcActualHoldDownTime OBJECT-TYPE
SYNTAX      Integer32 (30..1200)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION

```

```

        "MPC-p6 Hold Down Time
        Minimum time to wait before reinitiating a failed
        resolution attempt. The default value is
        mpcRetryTimeMaximum * 4."
 ::= { mpcActualEntry 8 }

--
-- DataAtmAddresses -> MPC
--

mpcDataAtmAddressTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MpcDataAtmAddressEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table which shows all the data ATM addresses
        associated with all MPOA Clients."
 ::= { mpcObjects 4 }

mpcDataAtmAddressEntry OBJECT-TYPE
    SYNTAX      MpcDataAtmAddressEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Each row defines one data ATM address associated with
        an MPC. NOTE: if an MPC has more than one data ATM
        address then there will be another entry which contains
        the same mpcIndex subIdentifier, with a different
        mpcDataAtmAddress."
    INDEX       { mpcIndex,
                  mpcDataAtmAddress
                }
 ::= { mpcDataAtmAddressTable 1 }

MpcDataAtmAddressEntry ::= SEQUENCE {
    mpcDataAtmAddress      AtmAddr,
    mpcDataAtmAddressRowStatus RowStatus
}

mpcDataAtmAddress OBJECT-TYPE
    SYNTAX      AtmAddr
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A data ATM Address which is associated with
        an MPOA Client specified by the mpcIndex."
 ::= { mpcDataAtmAddressEntry 1 }

mpcDataAtmAddressRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object allows creation and deletion of
        an MPOA Client's Data ATM Addresses. The
        row can be created/deleted by either an NMS
        or by the SNMP agent."
 ::= { mpcDataAtmAddressEntry 2 }

--
-- MPOA Client statistics group

```

--

```

mpcStatisticsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MpcStatisticsEntry
    MAX-ACCESS  not-accessible

    STATUS      current
    DESCRIPTION
        "A read-only table which contains statistical
        information for all MPOA Clients that
        this agent manages."
    ::= { mpcObjects 5 }

mpcStatisticsEntry OBJECT-TYPE
    SYNTAX      MpcStatisticsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Each row in this table contains statistics
        for one MPOA Client."
    AUGMENTS   { mpcConfigEntry }
    ::= { mpcStatisticsTable 1 }

MpcStatisticsEntry ::=SEQUENCE {
    mpcStatTxMpoaResolveRequests      Counter32,
    mpcStatRxMpoaResolveReplyAcks     Counter32,
    mpcStatRxMpoaResolveReplyInsufECResources Counter32,
    mpcStatRxMpoaResolveReplyInsufSCResources Counter32,
    mpcStatRxMpoaResolveReplyInsufEitherResources Counter32,
    mpcStatRxMpoaResolveReplyUnsupportedInetProt Counter32,
    mpcStatRxMpoaResolveReplyUnsupportedMacEncaps Counter32,
    mpcStatRxMpoaResolveReplyUnspecifiedOther Counter32,
    mpcStatRxMpoaImpRequests          Counter32,
    mpcStatTxMpoaImpReplyAcks         Counter32,
    mpcStatTxMpoaImpReplyInsufECResources Counter32,
    mpcStatTxMpoaImpReplyInsufSCResources Counter32,
    mpcStatTxMpoaImpReplyInsufEitherResources Counter32,
    mpcStatTxMpoaImpReplyUnsupportedInetProt Counter32,
    mpcStatTxMpoaImpReplyUnsupportedMacEncaps Counter32,
    mpcStatTxMpoaImpReplyUnspecifiedOther Counter32,
    mpcStatTxMpoaEgressCachePurgeRequests Counter32,
    mpcStatRxMpoaEgressCachePurgeReplies Counter32,
    mpcStatRxMpoaKeepAlives          Counter32,
    mpcStatRxMpoaTriggers             Counter32,
    mpcStatRxMpoaDataPlanePurges     Counter32,
    mpcStatTxMpoaDataPlanePurges     Counter32,
    mpcStatRxNhrpPurgeRequests       Counter32,
    mpcStatTxNhrpPurgeReplies        Counter32,

    -- NOTE: since the MPC supersedes the NHC's role,
    -- the following counters should be counted here,
    -- as opposed to the NHC.

    mpcStatRxErrUnrecognizedExtensions Counter32,
    mpcStatRxErrLoopDetecteds         Counter32,
    mpcStatRxErrProtoAddrUnreachables Counter32,
    mpcStatRxErrProtoErrors           Counter32,
    mpcStatRxErrSduSizeExceededs     Counter32,
    mpcStatRxErrInvalidExtensions     Counter32,
    mpcStatRxErrInvalidReplies       Counter32,

```

```

mpcStatRxErrAuthenticationFailures      Counter32,
mpcStatRxErrHopCountExceededs          Counter32
}

mpcStatTxMpoaResolveRequests OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of MPOA Resolve Requests transmitted
        by this MPC.

        Discontinuities in the value of this counter can occur
        at re-initialization of the management system, and/or
        re-initialization of the MPC, and at other times,
        as indicated by the value of mpcDiscontinuityTime."
    ::= { mpcStatisticsEntry 1 }

mpcStatRxMpoaResolveReplyAcks OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of positively acknowledged MPC
        Resolved Replies received by this MPC with an MPOA
        CIE Code of 0x00 (Success).
        Discontinuities in the value of this counter can occur at
        re-initialization of the management system, and/or re-
        initialization
        of the MPC, and at other times, as indicated by the value of
        mpcDiscontinuityTime."
    REFERENCE  "Multiprotocol Over ATM.  AF-MPOA-0087.000.  Page 62."
    ::= { mpcStatisticsEntry 2 }

mpcStatRxMpoaResolveReplyInsufECResources OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of MPOA Resolution Replies received with
        an MPOA CIE Code of 0x81, 'Insufficient resources to
        accept egress cache entry'.
        Discontinuities in the value of this counter can occur
        at re-initialization of the management system, and/or
        re-initialization of the MPC, and at other times,
        as indicated by the value of mpcDiscontinuityTime."
    REFERENCE  "Multiprotocol Over ATM.  AF-MPOA-0087.000.  Page 62."
    ::= { mpcStatisticsEntry 3 }

mpcStatRxMpoaResolveReplyInsufSCResources OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of MPOA Resolution Replies received with
        an MPOA CIE Code of 0x82, 'Insufficient resources to
        accept the shortcut'.
        Discontinuities in the value of this counter can occur
        at re-initialization of the management system, and/or
        re-initialization of the MPC, and at other times,

```

as indicated by the value of mpcDiscontinuityTime."

REFERENCE
 "Multiprotocol Over ATM. AF-MPOA-0087.000. Page 62."
 ::= { mpcStatisticsEntry 4 }

mpcStatRxMpoaResolveReplyInsufEitherResources OBJECT-TYPE
 SYNTAX Counter32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "The number of MPOA Resolution Replies received with an MPOA CIE Code of 0x83, 'Insufficient resources to accept either shortcut or egress cache entry'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPC, and at other times, as indicated by the value of mpcDiscontinuityTime."
 REFERENCE
 "Multiprotocol Over ATM. AF-MPOA-0087.000. Page 62."
 ::= { mpcStatisticsEntry 5 }

mpcStatRxMpoaResolveReplyUnsupportedInetProt OBJECT-TYPE
 SYNTAX Counter32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "The number of MPOA Resolution Replies received with an MPOA CIE Code of 0x84, 'Unsupported Inter-network Layer protocol'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPC, and at other times, as indicated by the value of mpcDiscontinuityTime."
 REFERENCE
 "Multiprotocol Over ATM. AF-MPOA-0087.000. Page 62."
 ::= { mpcStatisticsEntry 6 }

mpcStatRxMpoaResolveReplyUnsupportedMacEncaps OBJECT-TYPE
 SYNTAX Counter32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "The number of MPOA Resolution Replies received with an MPOA CIE Code of 0x85, 'Unsupported MAC layer encapsulation'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPC, and at other times, as indicated by the value of mpcDiscontinuityTime."
 REFERENCE
 "Multiprotocol Over ATM. AF-MPOA-0087.000. Page 62."
 ::= { mpcStatisticsEntry 7 }

mpcStatRxMpoaResolveReplyUnspecifiedOther OBJECT-TYPE
 SYNTAX Counter32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "The number of MPOA Resolution Replies received with an MPOA CIE Code of 0x88, 'Unspecified/Other'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPC, and at other times,

```

        as indicated by the value of mpcDiscontinuityTime."
REFERENCE
    "Multiprotocol Over ATM.  AF-MPOA-0087.000.  Page 62."
 ::= { mpcStatisticsEntry 8 }

mpcStatRxMpoaImpRequests OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of MPOA Cache Imposition Requests received
        by this MPC.
        Discontinuities in the value of this counter can occur at re-
        initialization
        of the management system, and/or re-initialization of the MPC,
        and at
        other times, as indicated by the value of mpcDiscontinuityTime."
 ::= { mpcStatisticsEntry 9 }

mpcStatTxMpoaImpReplyAcks OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of successful MPOA Cache Imposition replies
        transmitted by this MPC with an MPOA CIE Code of 0x00
        `Success`.
        Discontinuities in the value of this counter can occur at re-
        initialization
        of the management system, and/or re-initialization of the MPC,
        and at
        other times, as indicated by the value of mpcDiscontinuityTime."
REFERENCE
    "Multiprotocol Over ATM.  AF-MPOA-0087.000.  Page 62."
 ::= { mpcStatisticsEntry 10 }

mpcStatTxMpoaImpReplyInsufECResources OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of MPOA Cache Imposition Replies transmitted with
        an MPOA CIE Code of 0x81, `Insufficient resources to
        accept egress cache entry`.
        Discontinuities in the value of this counter can occur
        at re-initialization of the management system, and/or
        re-initialization of the MPC, and at other times,
        as indicated by the value of mpcDiscontinuityTime."
REFERENCE
    "Multiprotocol Over ATM.  AF-MPOA-0087.000.  Page 62."
 ::= { mpcStatisticsEntry 11 }

mpcStatTxMpoaImpReplyInsufSCResources OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of MPOA Imposition Replies transmitted with
        an MPOA CIE Code of 0x82, `Insufficient resources to
        accept shortcut`.
        Discontinuities in the value of this counter can occur
        at re-initialization of the management system, and/or

```

```

        re-initialization of the MPC, and at other times,
        as indicated by the value of mpcDiscontinuityTime."
REFERENCE
    "Multiprotocol Over ATM.  AF-MPOA-0087.000.  Page 62."
 ::= { mpcStatisticsEntry 12 }

mpcStatTxMpoaImpReplyInsufEitherResources OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of MPOA Imposition Replies transmitted with
    an MPOA CIE Code of 0x83, 'Insufficient resources to
    accept either shortcut or egress cache entry'.
    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and/or
    re-initialization of the MPC, and at other times,
    as indicated by the value of mpcDiscontinuityTime."
REFERENCE
    "Multiprotocol Over ATM.  AF-MPOA-0087.000.  Page 62."
 ::= { mpcStatisticsEntry 13 }

mpcStatTxMpoaImpReplyUnsupportedInetProt OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of MPOA Imposition Replies transmitted with
    an MPOA CIE Code of 0x84, 'Unsupported Internetwork Layer
    protocol'.
    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and/or
    re-initialization of the MPC, and at other times,
    as indicated by the value of mpcDiscontinuityTime."
REFERENCE
    "Multiprotocol Over ATM.  AF-MPOA-0087.000.  Page 62."
 ::= { mpcStatisticsEntry 14 }

mpcStatTxMpoaImpReplyUnsupportedMacEncaps OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of MPOA Imposition Replies transmitted with
    an MPOA CIE Code of 0x85, 'Unsupported MAC Layer
    encapsulation'.
    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and/or
    re-initialization of the MPC, and at other times,
    as indicated by the value of mpcDiscontinuityTime."
REFERENCE
    "Multiprotocol Over ATM.  AF-MPOA-0087.000.  Page 62."
 ::= { mpcStatisticsEntry 15 }

mpcStatTxMpoaImpReplyUnspecifiedOther OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of MPOA Imposition Replies transmitted with
    an MPOA CIE Code of 0x88, 'Unspecified/Other'.
    Discontinuities in the value of this counter can occur

```

at re-initialization of the management system, and/or re-initialization of the MPC, and at other times, as indicated by the value of mpcDiscontinuityTime."

REFERENCE

"Multiprotocol Over ATM. AF-MPOA-0087.000. Page 62."

::= { mpcStatisticsEntry 16 }

mpcStatTxMpoaEgressCachePurgeRequests OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of MPOA Egress Cache Purge Requests transmitted by this MPC.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPC, and at other times, as indicated by the value of mpcDiscontinuityTime."

::= { mpcStatisticsEntry 17 }

mpcStatRxMpoaEgressCachePurgeReplies OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of MPOA Egress Cache Purge Replies received by this MPC.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPC, and at other times, as indicated by the value of mpcDiscontinuityTime."

::= { mpcStatisticsEntry 18 }

mpcStatRxMpoaKeepAlives OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of MPOA Keep Alive messages received by this MPC.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPC, and at other times, as indicated by the value of mpcDiscontinuityTime."

::= { mpcStatisticsEntry 19 }

mpcStatRxMpoaTriggers OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of MPOA Trigger messages received by this MPC.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPC, and at other times, as indicated by the value of mpcDiscontinuityTime."

::= { mpcStatisticsEntry 20 }

mpcStatRxMpoaDataPlanePurges OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

```

        "The number of MPOA Data Plane Purge messages received
        by this MPC.
        Discontinuities in the value of this counter can occur
        at re-initialization of the management system, and/or
        re-initialization of the MPC, and at other times,
        as indicated by the value of mpcDiscontinuityTime."
 ::= { mpcStatisticsEntry 21 }

mpcStatTxMpoaDataPlanePurges OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of MPOA Data Plane Purge messages transmitted
        by this MPC.
        Discontinuities in the value of this counter can occur
        at re-initialization of the management system, and/or
        re-initialization of the MPC, and at other times,
        as indicated by the value of mpcDiscontinuityTime."
 ::= { mpcStatisticsEntry 22 }

mpcStatRxNhrpPurgeRequests OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of Purge Requests received by this MPC.
        Discontinuities in the value of this counter can occur
        at re-initialization of the management system, and/or
        re-initialization of the MPC, and at other times,
        as indicated by the value of mpcDiscontinuityTime."
 ::= { mpcStatisticsEntry 23 }

mpcStatTxNhrpPurgeReplies OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of Purge Replies transmitted by this MPC.
        Discontinuities in the value of this counter can occur
        at re-initialization of the management system, and/or
        re-initialization of the MPC, and at other times,
        as indicated by the value of mpcDiscontinuityTime."
 ::= { mpcStatisticsEntry 24 }

mpcStatRxErrUnrecognizedExtensions OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of Error Indication packets received by
        this MPC with the error code 'Unrecognized Extension'.
        Discontinuities in the value of this counter can occur
        at re-initialization of the management system, and/or
        re-initialization of the MPC, and at other times,
        as indicated by the value of mpcDiscontinuityTime."
 ::= { mpcStatisticsEntry 25 }

mpcStatRxErrLoopDetecteds OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current

```

DESCRIPTION

"The number of Error Indication packets received by this MPC with the error code 'Loop Detected'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPC, and at other times, as indicated by the value of mpcDiscontinuityTime."

```
::= { mpcStatisticsEntry 26 }
```

mpcStatRxErrProtoAddrUnreachables OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
```

DESCRIPTION

"The number of Error Indication packets received by this MPC with the error code 'Protocol Address Unreachable'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPC, and at other times, as indicated by the value of mpcDiscontinuityTime."

```
::= { mpcStatisticsEntry 27 }
```

mpcStatRxErrProtoErrors OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
```

DESCRIPTION

"The number of Error Indication packets received by this MPC with the error code 'Protocol Errors'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPC, and at other times, as indicated by the value of mpcDiscontinuityTime."

```
::= { mpcStatisticsEntry 28 }
```

mpcStatRxErrSduSizeExceededs OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
```

DESCRIPTION

"The number of Error Indication packets received by this MPC with the error code 'SDU Size Exceeded'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPC, and at other times, as indicated by the value of mpcDiscontinuityTime."

```
::= { mpcStatisticsEntry 29 }
```

mpcStatRxErrInvalidExtensions OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
```

DESCRIPTION

"The number of Error Indication packets received by this MPC with the error code 'Invalid Extensions'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPC, and at other times, as indicated by the value of mpcDiscontinuityTime."

```
::= { mpcStatisticsEntry 30 }
```

mpcStatRxErrInvalidReplies OBJECT-TYPE

```

SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of Error Indication packets received by
    this MPC with the error code 'Invalid Reply'.
    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and/or
    re-initialization of the MPC, and at other times,
    as indicated by the value of mpcDiscontinuityTime."
 ::= { mpcStatisticsEntry 31 }

mpcStatRxErrAuthenticationFailures OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of Error Indication packets received by
    this MPC with the error code 'Authentication Failure'.
    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and/or
    re-initialization of the MPC, and at other times,
    as indicated by the value of mpcDiscontinuityTime."
 ::= { mpcStatisticsEntry 32 }

mpcStatRxErrHopCountExceeded OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of Error Indication packets received by
    this MPC with the error code 'Hop Count Exceeded'.
    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and/or
    re-initialization of the MPC, and at other times,
    as indicated by the value of mpcDiscontinuityTime."
 ::= { mpcStatisticsEntry 33 }

--
-- MPOA Client Protocol support group
--

mpcProtocolTable OBJECT-TYPE
SYNTAX      SEQUENCE OF MpcProtocolEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "List of protocols, per MPC, for which flow
    detection is enabled"
REFERENCE
    "Multiprotocol Over ATM Version 1.0 (Letter Ballot),
    Section 4.1.2.1 MPC Parameters"
 ::= { mpcObjects 6 }

mpcProtocolEntry OBJECT-TYPE
SYNTAX      MpcProtocolEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "Each row indicates one protocol for which an MPC will
    do flow detection.  If the LECS was contacted for
    configuration information, and the Control Octet of

```

the MPC-p3 has the value of 0x01, 'Enable', then protocol values retrieved from the LECS are reflected in this table and the mpcLECSValue object will be (1) true. Also, the user or agent can create rows which appropriately correspond to the MPC denoted by mpcIndex, and the mpcLECSValue object will be set to (2) false. NOTE: if the LECS does not return information for the MPC-p3 parameter, or if in manual mode, the user or agent should create at least one entry for the corresponding MPC.

Both, LECS and user and/or agent created rows may exist in this Table."

```

INDEX          {      mpcIndex,
                  mpcFlowDetectProtocol
                }
 ::= { mpcProtocolTable 1 }

MpcProtocolEntry ::= SEQUENCE {
    mpcFlowDetectProtocol  InternetworkAddrType,
    mpcLECSValue           TruthValue,
    mpcProtocolRowStatus   RowStatus
}

mpcFlowDetectProtocol OBJECT-TYPE
    SYNTAX      InternetworkAddrType
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The protocol on which flow detection is performed.
        If this value was obtained from the LECS then this
        value is one of the collection of values returned
        in the MPC-p3 parameter."
    ::= { mpcProtocolEntry 1 }

mpcLECSValue OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object reflects if the current entry is due to
        a retrieval from the LECS or not.  If this entry is
        due to the LECS, then true(1) is the value for this
        object, otherwise, false (2)."
```

```

 ::= { mpcProtocolEntry 2 }

mpcProtocolRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object is used by an agent or manager to
        create, delete or modify a row in this table."
    ::= { mpcProtocolEntry 3 }

--
-- LEC -> MPC Mapping group
--
```

```

mpcMappingTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MpcMappingEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A table mapping the 'lecIndex' values of
        LANE Clients to the 'mpcIndex' values of
        corresponding MPOA Clients."
    ::= { mpcObjects 7 }

mpcMappingEntry OBJECT-TYPE
    SYNTAX      MpcMappingEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Each row defines one lecIndex --> mpcIndex mapping.
        The mpcIndex that a lecIndex maps to is not
        necessarily unique (an MPC can serve many LECs, however,
        a LEC cannot be served by more than one MPC)."
    REFERENCE
        "Multiprotocol Over ATM Version 1.0 (Letter Ballot),
        Section 4.4."
    INDEX       { lecIndex }
    ::= { mpcMappingTable 1 }

MpcMappingEntry ::= SEQUENCE {
    mpcMappingRowStatus      RowStatus,
    mpcMappingIndex          MpcIndex
}

mpcMappingRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object is used by an agent or manager to create, delete
        or modify a row in this table."
    ::= { mpcMappingEntry 1 }

mpcMappingIndex OBJECT-TYPE
    SYNTAX      MpcIndex
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The mpcIndex of the MPOA Client that is performing
        flow detection for the LANE Client represented by
        the lecIndex."
    ::= { mpcMappingEntry 2 }

--
-- MPOA Client MPS information group
--

mpcMpsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MpcMpsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This is a read-only table which contains
        information about the MPSs that these MPCs
        know about."
    ::= { mpcObjects 8 }

```

```

mpcMpsEntry OBJECT-TYPE
    SYNTAX      MpcMpsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A row created by an MPC.  The MPC learns about
        an MPS and creates a row."
    INDEX       { mpcMpsIndex }
    ::= { mpcMpsTable 1 }

MpcMpsEntry ::= SEQUENCE {
    mpcMpsIndex      MpsIndex,
    mpcMpsAtmAddr   AtmAddr
}

mpcMpsIndex OBJECT-TYPE
    SYNTAX      MpsIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The MPS's index which is used to identify
        a row in this table."
    ::= { mpcMpsEntry 1 }

mpcMpsAtmAddr OBJECT-TYPE
    SYNTAX      AtmAddr
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Control ATM Address of the MPS"
    ::= { mpcMpsEntry 2 }

--
-- MPOA Client's MAC Address group
--

-- the following table has been obsoleted it does
-- not support multinetted capabilities.
-- Please see the section of the
-- MPOA v1.1 for a complete explanation of the
-- multinetted capabilities.
-- This table has been replaced by the
-- mpcMpsMultipleMacAddressTable.

mpcMpsMacAddressTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MpcMpsMacAddressEntry
    MAX-ACCESS  not-accessible
    STATUS      obsolete
    DESCRIPTION
        "This is a read-only table which contains
        information about all the MPSs' MAC Addresses
        that these MPCs know about."
    ::= { mpcObjects 9 }

mpcMpsMacAddressEntry OBJECT-TYPE
    SYNTAX      MpcMpsMacAddressEntry
    MAX-ACCESS  not-accessible
    STATUS      obsolete
    DESCRIPTION
        "A row is created by an MPC.  The MPC learns about an
        MPS's MAC Address and creates a row."

```

```

INDEX          {      mpcMpsIndex,
                  mpcLecIndex
                }
 ::= { mpcMpsMacAddressTable 1 }

MpcMpsMacAddressEntry ::= SEQUENCE {
    mpcLecIndex      LecIndex,
    mpcMpsMacAddress MacAddress
}

mpcLecIndex OBJECT-TYPE
    SYNTAX      LecIndex
    MAX-ACCESS  not-accessible
    STATUS      obsolete
    DESCRIPTION
        "The lecIndex which represents the associated LEC."
    ::= { mpcMpsMacAddressEntry 1 }

mpcMpsMacAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-only
    STATUS      obsolete
    DESCRIPTION
        "The MAC Address of the MPS."
    REFERENCE
        "Multiprotocol Over ATM Version 1.0 (Letter Ballot),
        Section 3.3.3.1"
    ::= { mpcMpsMacAddressEntry 2 }

-- end of obsoleted table

--
-- MPOA Client Ingress Cache group
--

mpcIngressCacheTxTotalPackets OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total number of packets transmitted over MPC Short Cuts."
    ::= { mpcObjects 10 }

mpcIngressCacheTxTotalOctets OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total number of octets transmitted over MPC Short Cuts."
    ::= { mpcObjects 11 }

mpcIngressCacheTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MpcIngressCacheEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains information for MPC Caches for
        the ingress MPC."
    ::= { mpcObjects 12 }

```

```

mpcIngressCacheEntry OBJECT-TYPE
    SYNTAX      MpcIngressCacheEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry contains control information
        for a row in a MPC's Ingress Cache."
    INDEX       {      mpcIngressCacheDestInetAddressType,
                    mpcIngressCacheDestAddr,
                    mpcIndex,
                    mpcMpsIndex
                }
    ::= { mpcIngressCacheTable 1 }

MpcIngressCacheEntry ::= SEQUENCE {
    mpcIngressCacheDestInetAddressType  InetAddressType,
    mpcIngressCacheDestAddr             InetAddress,
    mpcIngressCachePrefixLen            Integer32,
    mpcIngressCacheDestAtmAddr          AtmAddr,

    mpcIngressCacheSrcAtmAddr           AtmAddr,
    mpcIngressCacheEntryState           INTEGER,

    mpcIngressCacheEgressCacheTagValid TruthValue,
    mpcIngressCacheEgressCacheTag       Integer32,

    --
    -- Information for diagnosing problems
    --
    mpcIngressCacheLastNhrpCieCode      INTEGER,
    mpcIngressCacheSigErrCode           Integer32,
    mpcIngressCacheRetries               Counter32,
    mpcIngressCacheTimeUntilNextResolutionRequest
                                        TimeInterval,
    mpcIngressCacheHoldingTime          TimeInterval,
    mpcIngressCacheServiceCategory      INTEGER
}

mpcIngressCacheDestInetAddressType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The type of the destination internetwork layer address."
    ::= { mpcIngressCacheEntry 1 }

mpcIngressCacheDestAddr OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The destination internetwork layer address for which
        this entry is defined."
    ::= { mpcIngressCacheEntry 2 }

mpcIngressCachePrefixLen OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Defines an equivalence class of addresses that match

```

```

        Prefix Length bit positions of the destination
        internetwork layer address."
 ::= { mpcIngressCacheEntry 3 }

mpcIngressCacheDestAtmAddr OBJECT-TYPE
    SYNTAX      AtmAddr
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The Destination ATM Address received in the MPOA
        Resolution Reply."
 ::= { mpcIngressCacheEntry 4 }

mpcIngressCacheSrcAtmAddr OBJECT-TYPE
    SYNTAX      AtmAddr
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The Source ATM Address for the MPOA Resolution Request."
 ::= { mpcIngressCacheEntry 5 }

mpcIngressCacheEntryState OBJECT-TYPE
    SYNTAX      INTEGER {
                    doesNotExist (1),
                    inactive (2),
                    active(3),
                    negative(4)
                }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The present state of this MPC ingress cache.
        The states are:
        doesNotExist (1) -- the state is not yet available
        inactive (2)    -- state exists, entry is not yet
                        active. For an example, if
                        using the Finite State Machine
                        Appendix I.2, then the states
                        Cached and query are considered
                        'inactive'.
        active (3)     -- state exists, entry is active.
                        For an example, if using the
                        Finite State Machine in Appendix I.2,
                        then the states resolved and refresh
                        are considered 'active'.
        negative (4)   -- state exists, entry is negative,
                        which could mean a NAK response was
                        received, or entry is doing a
                        retry, etc. For example, if using
                        the Finite State Machine in Appendix
                        I.2, then the state 'hold down' is
                        considered 'negative'."

    REFERENCE
        "Multiprotocol Over ATM, Letter Ballot, Appendix I.2."
 ::= { mpcIngressCacheEntry 6 }

mpcIngressCacheEgressCacheTagValid OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "If the value of this object is true(1), then a valid

```

Egress Cache Tag is present and the value of the Egress Cache Tag is in mpcIngressCacheEgressCacheTag. Otherwise, if this value is false(2), then there was no Egress Cache Tag, and the value of mpcIngressCacheEgressCacheTag is undefined."

```
::= { mpcIngressCacheEntry 7 }
```

mpcIngressCacheEgressCacheTag OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"If a valid Egress Cache Tag is present, then this object contains the value of that tag. To determine if this object contains a valid value, mpcIngressCacheEgressTagValid should be used."

REFERENCE

"Multiprotocol Over ATM Version 1.0 (Letter Ballot), Section 4.4.4.1."

```
::= { mpcIngressCacheEntry 8 }
```

mpcIngressCacheLastNhrpCieCode OBJECT-TYPE

SYNTAX INTEGER (0..255)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The last NHRP CIE code received for this entry. This value is valid only during the Hold Down period of the cache entry. This value is undefined otherwise."

REFERENCE

"Normative section 4.4.6.1.1 of Multiprotocol Over ATM Version 1.0 (Letter Ballot)"

```
::= { mpcIngressCacheEntry 9 }
```

mpcIngressCacheSigErrCode OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Error code or Success of the last signalling request for this cache entry."

```
::= { mpcIngressCacheEntry 10 }
```

mpcIngressCacheRetries OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The current number of times this MPC has issued a resolution request since it received a valid reply."

```
::= { mpcIngressCacheEntry 11 }
```

mpcIngressCacheTimeUntilNextResolutionRequest OBJECT-TYPE

SYNTAX TimeInterval

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The amount of time the MPC must wait before issuing the next resolution request."

```
::= { mpcIngressCacheEntry 12 }
```

mpcIngressCacheHoldingTime OBJECT-TYPE

```

SYNTAX      TimeInterval
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The time that this MPC's Ingress Cache Entry will remain
    valid.  If the mpcIngressCacheEntryState is not active
    this value will be zero."
 ::= { mpcIngressCacheEntry 13 }

mpcIngressCacheServiceCategory OBJECT-TYPE
SYNTAX      INTEGER (0..65535)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The service categories supported for this shortcut."
REFERENCE
    "Lane V2 LUNI TLVs. AF-LANE-0084 page 122"
 ::= { mpcIngressCacheEntry 14 }

--
-- MPOA Client Egress Cache group
--

mpcEgressCacheRxTotalPackets OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This counts the total number of packets
    received by MPC Short Cuts."
 ::= { mpcObjects 13 }

mpcEgressCacheRxTotalOctets OBJECT-TYPE
SYNTAX      Counter64
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This counts the total number of octets
    received by MPC Short Cuts."
 ::= { mpcObjects 14 }

mpcEgressCacheTable OBJECT-TYPE
SYNTAX      SEQUENCE OF MpcEgressCacheEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "This table contains Egress Cache information for
    all the MPCs which this agent manages."
 ::= { mpcObjects 15 }

mpcEgressCacheEntry OBJECT-TYPE
SYNTAX      MpcEgressCacheEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "An entry in the MPOA Client's Egress Cache table."
INDEX      {
            mpcEgressCacheId,
            mpcIndex,
            mpcMpsIndex
          }
 ::= { mpcEgressCacheTable 1 }

```

```

MpcEgressCacheEntry ::= SEQUENCE {
    mpcEgressCacheId                Integer32,
    mpcEgressCacheInetAddressType   InternetNetworkAddrType,
    mpcEgressCacheIDestAddr         InternetNetworkAddr,
    mpcEgressCachePrefixLen        Integer32,
    mpcEgressCacheEntryState       INTEGER,
    mpcEgressCacheEgressCacheTagValid TruthValue,
    mpcEgressCacheEgressCacheTag   Integer32,
    mpcEgressCacheHoldTime         TimeInterval,
    mpcEgressCacheDataLinkHeader   OCTET STRING,
    mpcEgressCacheIngressMpcDataAtmAddr AtmAddr,
    mpcEgressCacheLecIndex         LecIndex,
    mpcEgressCacheServiceCategory  INTEGER
}

mpcEgressCacheId OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Cache ID Provided by the MPS in the Cache
        Imposition Request."
    ::= { mpcEgressCacheEntry 1 }

mpcEgressCacheInetAddressType OBJECT-TYPE
    SYNTAX      InternetNetworkAddrType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Type of Internetnetwork Address in this cache entry."
    ::= { mpcEgressCacheEntry 2 }

mpcEgressCacheIDestAddr OBJECT-TYPE
    SYNTAX      InternetNetworkAddr
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The destination internetnetwork layer address
        for which this entry is defined."
    ::= { mpcEgressCacheEntry 3 }

mpcEgressCachePrefixLen OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Defines an equivalence class of addresses that match
        Prefix Length bit positions of the destination
        internetnetwork layer address."
    ::= { mpcEgressCacheEntry 4 }

mpcEgressCacheEntryState OBJECT-TYPE
    SYNTAX      INTEGER {
        doesNotExist(1),
        inactive(2),
        active(3),
        negative(4)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The present state of this MPC egress cache entry."

```

The states are:

- doesNotExist (1) -- the state is not yet available,
- inactive (2) -- state exists, entry is not yet active,
- active (3) -- state exists, entry is active.

For example, if using the Finite State Machine in Appendix 1.5, the states active and flooding are 'active' state.

- negative (4) -- state exists, entry is negative.

For example, if using the Finite State Machine in Appendix 1.5, the state purging is 'negative'."

REFERENCE

"MPOA Letter Ballot, Appendix I.5."

```
::= { mpcEgressCacheEntry 5 }
```

mpcEgressCacheEgressCacheTagValid OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"If the value of this object is true(1), then a valid Egress Cache Tag is present and the value of the Egress Cache Tag is in mpcEgressCacheEgressCacheTag. Otherwise, if this value is false(2), then there was no Egress Cache Tag, and the value of mpcEgressCacheEgressCacheTag is undefined."

```
::= { mpcEgressCacheEntry 6 }
```

mpcEgressCacheEgressCacheTag OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"If a valid Egress Cache Tag is present, then this object contains the value of that tag. To determine if this object contains a valid value, mpcEgressCacheEgressCacheTagValid should be used."

```
::= { mpcEgressCacheEntry 7 }
```

mpcEgressCacheHoldTime OBJECT-TYPE

SYNTAX TimeInterval

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The remaining time for which this entry is valid."

```
::= { mpcEgressCacheEntry 8 }
```

mpcEgressCacheDataLinkHeader OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (0..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The DataLink header that the egress client rebuilds the original DataLink packet with."

```
::= { mpcEgressCacheEntry 9 }
```

mpcEgressCacheIngressMpcDataAtmAddr OBJECT-TYPE

SYNTAX AtmAddr

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The Data ATM Address of the ingress MPC that issued the MPOA Resolution request"

REFERENCE

"Multiprotocol Over ATM. AF-MPOA-0087.000. Page 81."

::= { mpcEgressCacheEntry 10 }

mpcEgressCacheLecIndex OBJECT-TYPE

SYNTAX LecIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This is the lecIndex of the LANE Client that this flow is associated with. This can be used to get the ELAN name as well as other LANE parameters."

::= { mpcEgressCacheEntry 11 }

mpcEgressCacheServiceCategory OBJECT-TYPE

SYNTAX INTEGER (0..65535)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This is a bitmap describing the service categories supported for this shortcut. This value represents an inclusive OR of the bits:

bit 1 - if rt-VBR is supported,
bit 2 - if nrt-VBR is supported,
bit 4 - if ABR is supported,
bit 8 - if CBR is supported

A value of 0 (zero) indicates that UBR is supported."

REFERENCE

"Lane V2 LUNI TLVs. AF-LANE-0084, page 122."

::= { mpcEgressCacheEntry 12 }

-- The following Table replaces the mpcMpsMacAddressTable (which
-- has been obsoleted.) The following table is more flexible in
-- that it can represent more than one MPS MAC Address used by
-- the MPC during flow detection.

mpcMpsObjects OBJECT IDENTIFIER ::= { mpcObjects 16 }

mpcMpsMultipleMacAddressTable OBJECT-TYPE

SYNTAX SEQUENCE OF MpcMpsMultipleMacAddressEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This is a read-only table which contains information about all the MPSS' MAC Addresses that these MPCs use during flow detection. Note that due to the multinetted case an MPC may learn about more than one MAC address from an MPS, thus there may be more than one MAC address for the same MPC - MPS - LecIndex represented in this Table. These MacAddresses are differentiated by the mpcMpsMacAddressIndex."

::= { mpcMpsObjects 1 }

```

mpcMpsMultipleMacAddressEntry OBJECT-TYPE
    SYNTAX      MpcMpsMultipleMacAddressEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A row is created by an MPC.  The MPC learns about a
        MPS and the one or more MAC Address of the MPS which
        the MPC uses during flow detection.  Each row represents
        an MPS MAC Address used by an MPC during flow detection."
    INDEX      {      mpcMpsIndex,
                    mpcFlowDetectLecIndex,
                    mpcMpsMacAddressIndex
                }
    ::= { mpcMpsMultipleMacAddressTable 1 }

MpcMpsMultipleMacAddressEntry ::= SEQUENCE {
    mpcFlowDetectLecIndex      LecIndex,
    mpcMpsMacAddressIndex      Integer32,
    mpcMpsFlowDetectMacAddress MacAddress
}

mpcFlowDetectLecIndex OBJECT-TYPE
    SYNTAX      LecIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The lecIndex which represents the associated LEC."
    ::= { mpcMpsMultipleMacAddressEntry 1 }

mpcMpsMacAddressIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This value is used to differentiate MAC Addresses from
        the same MPS used by the same MPC during flow detection.
        This value should be unique within the scope of this table."
    ::= { mpcMpsMultipleMacAddressEntry 2 }

mpcMpsFlowDetectMacAddress OBJECT-TYPE
    SYNTAX      MacAddress
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "An MPS MAC Address used by an MPC during flow detection."
    ::= { mpcMpsMultipleMacAddressEntry 3 }

--
-- MPOA Server Objects
--

mpsObjects OBJECT IDENTIFIER ::= { mpoaMIBObjects 3 }

mpsNextIndex OBJECT-TYPE
    SYNTAX      Integer32 (0..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object contains an appropriate value to
        be used for mpsIndex when creating entries in
        the mpsConfigTable.  The value 0 indicates that

```

no new rows can be created. Otherwise, it is recommended that values are assigned contiguously, starting from 1.

MPS creation by a Manager: To obtain the mpsIndex value for a new entry, the manager issues a management protocol retrieval operation to obtain the current value of this object. If the value retrieved is 0 (zero), the manager cannot create a row.

After each retrieval of a non-zero value, the manager should issue a management protocol SET operation using the value just retrieved. If the SET is successful, the agent should update the value to the next unassigned index, or zero if appropriate.

NOTE: the manager may also issue a set on this object with a value of its own choosing. If the set is successful, the manager may use this value for the mpsIndex. In this case, the agent would update the value to the next unassigned index, or zero if appropriate. The definition of 'next unassigned index' is any mpsNextIndex value that has not yet been set by a manager, or reserved by the agent (see next paragraph), since this agent was last re-initialized.

MPS creation by an Agent: When a row in the mpsConfigTable is created by an agent, the agent should reserve the value of the index by updating the value of this object to the next unassigned index or zero if appropriate. Thus, a manager will not be able to set an index reserved by an agent. In the situation of an agent re-initialization all currently used mpsIndexes must be preserved. In other words, the Agent should store in non-volatile memory all the currently used mpsIndexes (along with all necessary configuration information from the mpsConfigTable). When the agent is re-initialized, the mpsNextIndex value is any valid Integer32 which is not being used as an mpsIndex, except 0 which maintains its original definition of indicating that a row cannot be created."

```
::= { mpsObjects 1 }
```

```
--
```

```
-- MPOA Server configuration group
```

```
--
```

```
mpsConfigTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF MpsConfigEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

```
"The MPOA Server Configuration Table.
```

```
This table represents the configuration information for  
all MPOA Servers which this agent manages."
```

```
::= { mpsObjects 2 }
```

```
mpsConfigEntry OBJECT-TYPE
```

```
SYNTAX MpsConfigEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

```
"MPOA Server Configuration Entry.
```

```

        Each entry contains configuration information
        for one MPOA Server."
INDEX      { mpsIndex }
::= { mpsConfigTable 1 }

MpsConfigEntry ::= SEQUENCE {

    -- Primary config info: Index, mode and address information
    --
    mpsIndex          MpsIndex,
    mpsRowStatus      RowStatus,
    mpsConfigMode     INTEGER,
    mpsCtrlAtmAddr    AtmConfigAddr,

    --
    -- MPS parameters that can be obtained from
    -- the LECS.
    --
    mpsKeepAliveTime      Integer32,      -- MPS-p1
    mpsKeepAliveLifeTime  Integer32,      -- MPS-p2
    -- The Flow-detection Protocols (denoted with MPS-p3)
    -- are represented in the mpcProtocolsTable.
    mpsInitialRetryTime   Integer32,      -- MPS-p4
    mpsRetryTimeMaximum   Integer32,      -- MPS-p5
    mpsGiveupTime         Integer32,      -- MPS-p6
    mpsDefaultHoldingTime Integer32       -- MPS-p7
}

mpsIndex OBJECT-TYPE
    SYNTAX      MpsIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A value which uniquely identifies this conceptual
        row in the mpsConfigTable. The 'mpsNextIndex' object
        needs to be used to determine the value of this object.
        A row cannot be added, unless the mpsCtrlAtmAddress
        is unique.
        In the event of an MPS re-initialization, the value
        of this mpsIndex must remain the same. However, in
        the event of an agent re-initialization, this value
        does not need to be preserved."
    ::= { mpsConfigEntry 1 }

mpsRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object allows creation and deletion of MPOA Servers.
        Within each conceptual mpsConfigTable row, objects which are
        writable may be modified regardless of the value of
        mpsRowStatus. It is not necessary to set a row's status
        to 'notInService' first.

        A row cannot be created, unless the mpsAtmCtrlAddress
        in this table is unique. When an MPOA Server is created
        via this object, it will initially have
        'mpsActualState' = 'initialState'."
    ::= { mpsConfigEntry 2 }

mpsConfigMode OBJECT-TYPE

```

```

SYNTAX      INTEGER {
                automatic(1),
                manual(2)
            }
MAX-ACCESS   read-create
STATUS      current
DESCRIPTION
    "Indicates whether this MPS should auto-configure
    the next time it is (re-)initialized.
    In automatic(1) mode the LECS is contacted and
    requests are made for the MPS-p* parameters.
    In manual(2) mode, the values of the configuration
    parameters are obtained from the mpsConfigTable
    and the mpsProtocolTable."
DEFVAL { automatic }
 ::= { mpsConfigEntry 3 }

mpsCtrlAtmAddr OBJECT-TYPE
SYNTAX      AtmConfigAddr
MAX-ACCESS   read-create
STATUS      current
DESCRIPTION
    "The MPS's Control ATM Address.  There exists one
    Control ATM Address per MPS, therefore, the value
    of this entry is unique within the table."
 ::= { mpsConfigEntry 4 }

mpsKeepAliveTime OBJECT-TYPE
SYNTAX      Integer32 (1..300)
MAX-ACCESS   read-create
STATUS      current
DESCRIPTION
    "MPS-p1 Keep-alive time is max interval between
    the MPS sending MPOA Keep-Alives in seconds."
DEFVAL { 10 }
 ::= { mpsConfigEntry 5 }

mpsKeepAliveLifeTime OBJECT-TYPE
SYNTAX      Integer32 (3..1000)
MAX-ACCESS   read-create
STATUS      current
DESCRIPTION
    "MPS-p2 Keep-Alive Lifetime
    The length of time an MPC may consider a Keep-Alive
    valid in seconds.  This value must be at least
    three times the mpsKeepAliveTime (MPS-p1)."
DEFVAL { 35 }
 ::= { mpsConfigEntry 6 }

mpsInitialRetryTime OBJECT-TYPE
SYNTAX      Integer32 (1..300)
MAX-ACCESS   read-create
STATUS      current
DESCRIPTION
    "MPS-p4 is initial value in seconds for the
    MPOA retry mechanism."
DEFVAL { 5 }
 ::= { mpsConfigEntry 7 }

mpsRetryTimeMaximum OBJECT-TYPE
SYNTAX      Integer32 (10..300)

```

```

MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "MPS-p5 cumulative max value in seconds
    for Retry Time (MPS-p4)."
```

```

REFERENCE
    "Multiprotocol Over ATM.  AF-MPOA-0087.000.
    Section 4.3 MPOA Retry Mechanism"
DEFVAL { 40 }
 ::= { mpsConfigEntry 8 }
```

```

mpsGiveupTime OBJECT-TYPE
SYNTAX          Integer32 (5..300)
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "MPS-p6 Give Up Time.
    Minimum time in seconds to wait before giving up on a
    pending resolution request."
DEFVAL { 40 }
 ::= { mpsConfigEntry 9 }
```

```

mpsDefaultHoldingTime OBJECT-TYPE
SYNTAX          Integer32 (1..120)
MAX-ACCESS      read-create
STATUS          current
DESCRIPTION
    "MPS-p7 Default Holding Time in minutes.
    The default Holding Time used in NHRP
    Resolution Replies.  An egress MPS may use
    local information to determine a more
    appropriate Holding Time."
DEFVAL { 20 }
 ::= { mpsConfigEntry 10 }
```

```

--
-- MPOA Server Actual group
--
```

```

mpsActualTable OBJECT-TYPE
SYNTAX          SEQUENCE OF MpsActualEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "A read-only table containing identification, status,
    and operational information about the MPOA
    Servers this agent manages."
 ::= { mpsObjects 3 }
```

```

mpsActualEntry OBJECT-TYPE
SYNTAX          MpsActualEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "An entry in the MPS Actual Table.
    An entry represents a specific MPOA Server's
    status and operation information."
AUGMENTS { mpsConfigEntry }
 ::= { mpsActualTable 1 }
```

```

MpsActualEntry ::= SEQUENCE {
```

```

mpsActualState          INTEGER,
mpsDiscontinuityTime    TimeStamp,
mpsActualConfigMode     INTEGER,

--
-- Actual values of parameters
--
mpsActualKeepAlive      Integer32, -- MPS-p1
mpsActualKeepAliveLifeTime Integer32, -- MPS-p2
-- The Internetwork-layer Protocols for MPS-p3 are
-- represented in the mpsProtocolTable.
--
mpsActualInitialRetryTime Integer32, -- MPS-p4
mpsActualRetryTimeMaximum Integer32, -- MPS-p5
mpsActualGiveupTime      Integer32, -- MPS-p6
mpsActualDefaultHoldingTime Integer32 -- MPS-p7

}

mpsActualState OBJECT-TYPE
    SYNTAX      INTEGER {
                unknown(1),
                initialState(2),
                up(3),
                down(4)
            }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This object indicates the current operational
        status of the MPOA Server."
    ::= { mpsActualEntry 1 }

mpsDiscontinuityTime OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The value of sysUpTime on the most recent occasion at
        which any one or more of this MPS's counters
        experienced a discontinuity. The relevant counters
        are the specific instances associated with this
        MPS. If discontinuities have not occurred since the
        last re-initialization of the local management subsystem,
        then this object contains a zero value."
    ::= { mpsActualEntry 2 }

mpsActualConfigMode OBJECT-TYPE
    SYNTAX      INTEGER {
                automatic(1),
                manual(2)
            }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Indicates whether this MPS auto-configured
        when it was last (re-)initialized."
    ::= { mpsActualEntry 3 }

mpsActualKeepAlive OBJECT-TYPE
    SYNTAX      Integer32

```

```

MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The maximum amount of time in seconds this MPS waits
    between sending MPOA Keep-Alives."
 ::= { mpsActualEntry 5 }

mpsActualKeepAliveLifeTime OBJECT-TYPE
SYNTAX          Integer32
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The length of time in seconds this MPS
    considers a Keep-Alive valid."
 ::= { mpsActualEntry 6 }

mpsActualInitialRetryTime OBJECT-TYPE
SYNTAX          Integer32 (1..300)
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The actual initial value in seconds for the MPOA
    retry mechanism."
DEFVAL { 5 }
 ::= { mpsActualEntry 7 }

mpsActualRetryTimeMaximum OBJECT-TYPE
SYNTAX          Integer32 (30..300)
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The actual cumulative max value in seconds
    for Retry Time."
REFERENCE
    "Multiprotocol Over ATM. AF-MPOA-0087.000.
    Section 4.3 MPOA Retry Mechanism"
DEFVAL { 40 }
 ::= { mpsActualEntry 8 }

mpsActualGiveupTime OBJECT-TYPE
SYNTAX          Integer32 (5..300)
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "Minimum time in seconds that this MPS waits
    before giving up on a pending resolution request."
DEFVAL { 40 }
 ::= { mpsActualEntry 9 }

mpsActualDefaultHoldingTime OBJECT-TYPE
SYNTAX          Integer32 (1..120)
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The actual Holding Time in minutes used in
    NHRP Resolution Replies."
 ::= { mpsActualEntry 10 }

--
-- MPOA Server statistics group
--

```

```

mpsStatisticsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MpsStatisticsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table represents the statistical information
        for the MPSs, which this agent manages."
    ::= { mpsObjects 4 }

mpsStatisticsEntry OBJECT-TYPE
    SYNTAX      MpsStatisticsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Each row in this table contains statistics
        for one MPOA server."
    AUGMENTS   { mpsConfigEntry }
    ::= { mpsStatisticsTable 1 }

MpsStatisticsEntry ::= SEQUENCE {
    mpsStatRxMpoaResolveRequests          Counter32,
    mpsStatTxMpoaResolveReplyAcks        Counter32,
    mpsStatTxMpoaResolveReplyInsufECResources  Counter32,
    mpsStatTxMpoaResolveReplyInsufSCResources  Counter32,
    mpsStatTxMpoaResolveReplyInsufEitherResources Counter32,
    mpsStatTxMpoaResolveReplyUnsupportedInetProt Counter32,
    mpsStatTxMpoaResolveReplyUnsupportedMacEncaps Counter32,
    mpsStatTxMpoaResolveReplyUnspecifiedOther Counter32,
    mpsStatTxMpoaResolveReplyOther        Counter32,
    mpsStatGiveupTimeExpireds              Counter32,

    mpsStatTxMpoaImpRequests                Counter32,
    mpsStatRxMpoaImpReplyAcks                Counter32,
    mpsStatRxMpoaImpReplyInsufECResources    Counter32,
    mpsStatRxMpoaImpReplyInsufSCResources    Counter32,
    mpsStatRxMpoaImpReplyInsufEitherResources Counter32,
    mpsStatRxMpoaImpReplyUnsupportedInetProt Counter32,
    mpsStatRxMpoaImpReplyUnsupportedMacEncaps Counter32,
    mpsStatRxMpoaImpReplyUnspecifiedOther    Counter32,
    mpsStatRxMpoaImpReplyOther                Counter32,

    mpsStatRxMpoaEgressCachePurgeRequests    Counter32,
    mpsStatTxMpoaEgressCachePurgeReplies     Counter32,
    mpsStatTxMpoaTriggers                      Counter32,
    mpsStatTxNhrpResolveRequests              Counter32,
    mpsStatRxNhrpResolveReplies               Counter32,
    mpsStatRxNhrpResolveRequests              Counter32,
    mpsStatTxNhrpResolveReplies               Counter32
}

mpsStatRxMpoaResolveRequests OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of MPOA Resolve Requests received
        by this MPS which are translated to NHRP
        resolve requests.
        Discontinuities in the value of this counter can occur
        at re-initialization of the management system, and/or
        re-initialization of the MPS, and at other times,
        as indicated by the value of mpsDiscontinuityTime."

```

```
::= { mpsStatisticsEntry 1 }
```

```
mpsStatTxMpoaResolveReplyAcks OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

"The number of MPOA Resolve Replies transmitted by this MPS which contain the MPOA CIE Code of 0x00, 'Success'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPS, and at other times, as indicated by the value of mpsDiscontinuityTime."

```
REFERENCE
```

"Multiprotocol Over ATM. AF-MPOA-0087.000. Page 62."

```
::= { mpsStatisticsEntry 2 }
```

```
mpsStatTxMpoaResolveReplyInsufECResources OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

"The number of MPOA Resolve Replies transmitted by this MPS which contain the MPOA CIE Code of 0x81, 'Insufficient resources to accept egress cache entry'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPS, and at other times, as indicated by the value of mpsDiscontinuityTime."

```
REFERENCE
```

"Multiprotocol Over ATM. AF-MPOA-0087.000. Page 62."

```
::= { mpsStatisticsEntry 3 }
```

```
mpsStatTxMpoaResolveReplyInsufSCResources OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

"The number of MPOA Resolve Replies transmitted by this MPS which contain the MPOA CIE Code of 0x82, 'Insufficient resources to accept shortcut'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPS, and at other times, as indicated by the value of mpsDiscontinuityTime."

```
REFERENCE
```

"Multiprotocol Over ATM. AF-MPOA-0087.000. Page 62."

```
::= { mpsStatisticsEntry 4 }
```

```
mpsStatTxMpoaResolveReplyInsufEitherResources OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

"The number of MPOA Resolve Replies transmitted by this MPS which contain the MPOA CIE CODE of 0x83, 'Insufficient resources to accept either shortcut or egress cache entry'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPS, and at other times,

```

        as indicated by the value of mpsDiscontinuityTime."
REFERENCE
    "Multiprotocol Over ATM.  AF-MPOA-0087.000.  Page 62."
 ::= { mpsStatisticsEntry 5 }

mpsStatTxMpoaResolveReplyUnsupportedInetProt OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of MPOA Resolve Replies transmitted by this MPS
    which contain the MPOA CIE CODE of 0x84, 'Unsupported
    Internetwork
    Layer protocol'.
    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and/or
    re-initialization of the MPS, and at other times,
    as indicated by the value of mpsDiscontinuityTime."
REFERENCE
    "Multiprotocol Over ATM.  AF-MPOA-0087.000.  Page 62."
 ::= { mpsStatisticsEntry 6 }

mpsStatTxMpoaResolveReplyUnsupportedMacEncaps OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of MPOA Resolve Replies transmitted by this MPS
    which contain the MPOA CIE CODE of 0x85, 'Unsupported MAC
    layer encapsulation'.
    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and/or
    re-initialization of the MPS, and at other times,
    as indicated by the value of mpsDiscontinuityTime."
REFERENCE
    "Multiprotocol Over ATM.  AF-MPOA-0087.000.  Page 62."
 ::= { mpsStatisticsEntry 7 }

mpsStatTxMpoaResolveReplyUnspecifiedOther OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of MPOA Resolve Replies transmitted by this MPS
    which contain the MPOA CIE CODE of 0x88, 'Unspecified/Other'.
    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and/or
    re-initialization of the MPS, and at other times,
    as indicated by the value of mpsDiscontinuityTime."
REFERENCE
    "Multiprotocol Over ATM.  AF-MPOA-0087.000.  Page 62."
 ::= { mpsStatisticsEntry 8 }

mpsStatTxMpoaResolveReplyOther OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of MPOA Resolve Replies transmitted by this
    MPS which are not counted above.  NOTE:  this would
    include NHRP errors.
    Discontinuities in the value of this counter can occur

```

at re-initialization of the management system, and/or re-initialization of the MPS, and at other times, as indicated by the value of mpsDiscontinuityTime."

REFERENCE

"Multiprotocol Over ATM. AF-MPOA-0087.000. Page 62."

```
::= { mpsStatisticsEntry 9 }
```

mpsStatGiveupTimeExpires OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of times the MPS Give up Time (MPS-p6) has expired while waiting for a reply from a re-originated MPOA resolution request, i.e. a reply for a translated NHRP resolution request. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPS, and at other times, as indicated by the value of mpsDiscontinuityTime."

```
::= { mpsStatisticsEntry 10 }
```

mpsStatTxMpoaImpRequests OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of MPOA Cache Imposition Requests transmitted by this MPS. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPS, and at other times, as indicated by the value of mpsDiscontinuityTime."

```
::= { mpsStatisticsEntry 11 }
```

mpsStatRxMpoaImpReplyAcks OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of successful MPOA Cache Imposition Replies received by this MPS which contain an MPOA CIE Code of 0x00, 'Success'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPS, and at other times, as indicated by the value of mpsDiscontinuityTime."

```
::= { mpsStatisticsEntry 12 }
```

mpsStatRxMpoaImpReplyInsufECResources OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of MPOA Cache Imposition Replies received by this MPS which contain the MPOA CIE Code of 0x81, 'Insufficient resources to accept egress cache entry'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPS, and at other times, as indicated by the value of mpsDiscontinuityTime."

REFERENCE

"Multiprotocol Over ATM. AF-MPOA-0087.000. Page 62."
 ::= { mpsStatisticsEntry 13 }

mpsStatRxMpoaImpReplyInsufSCResources OBJECT-TYPE

SYNTAX Counter32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"The number of MPOA Cache Imposition Replies received by this MPS which contain the MPOA CIE Code of 0x82, 'Insufficient resources to accept shortcut'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPS, and at other times, as indicated by the value of mpsDiscontinuityTime."

REFERENCE

"Multiprotocol Over ATM. AF-MPOA-0087.000. Page 62."
 ::= { mpsStatisticsEntry 14 }

mpsStatRxMpoaImpReplyInsufEitherResources OBJECT-TYPE

SYNTAX Counter32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"The number of MPOA Cache Imposition Replies received by this MPS which contain the MPOA CIE Code of 0x83, 'Insufficient resources to accept either shortcut or egress cache entry'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPS, and at other times, as indicated by the value of mpsDiscontinuityTime."

REFERENCE

"Multiprotocol Over ATM. AF-MPOA-0087.000. Page 62."
 ::= { mpsStatisticsEntry 15 }

mpsStatRxMpoaImpReplyUnsupportedInetProt OBJECT-TYPE

SYNTAX Counter32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"The number of MPOA Cache Imposition Replies received by this MPS which contain the MPOA CIE Code of 0x84, 'Unsupported Internetwork Layer protocol'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPS, and at other times, as indicated by the value of mpsDiscontinuityTime."

REFERENCE

"Multiprotocol Over ATM. AF-MPOA-0087.000. Page 62."
 ::= { mpsStatisticsEntry 16 }

mpsStatRxMpoaImpReplyUnsupportedMacEncaps OBJECT-TYPE

SYNTAX Counter32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"The number of MPOA Cache Imposition Replies received by this MPS which contain the MPOA CIE Code of 0x85, 'Unsupported MAC layer encapsulation'. Discontinuities in the value of this counter can occur

at re-initialization of the management system, and/or re-initialization of the MPS, and at other times, as indicated by the value of mpsDiscontinuityTime."

REFERENCE
 "Multiprotocol Over ATM. AF-MPOA-0087.000. Page 62."
 ::= { mpsStatisticsEntry 17 }

mpsStatRxMpoaImpReplyUnspecifiedOther OBJECT-TYPE
 SYNTAX Counter32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "The number of MPOA Cache Imposition Replies received by this MPS which contain the MPOA CIE Code of 0x88, 'Unspecified/Other'. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPS, and at other times, as indicated by the value of mpsDiscontinuityTime."
 REFERENCE
 "Multiprotocol Over ATM. AF-MPOA-0087.000. Page 62."
 ::= { mpsStatisticsEntry 18 }

mpsStatRxMpoaImpReplyOther OBJECT-TYPE
 SYNTAX Counter32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "The number of MPOA Cache Imposition Replies received by this MPS which are not counted previously. NOTE: this would include NHRP errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPS, and at other times, as indicated by the value of mpsDiscontinuityTime."
 REFERENCE
 "Multiprotocol Over ATM. AF-MPOA-0087.000. Page 62."
 ::= { mpsStatisticsEntry 19 }

mpsStatRxMpoaEgressCachePurgeRequests OBJECT-TYPE
 SYNTAX Counter32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "The number of MPOA Egress Cache Purges Requests received by this MPS. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPS, and at other times, as indicated by the value of mpsDiscontinuityTime."
 ::= { mpsStatisticsEntry 20 }

mpsStatTxMpoaEgressCachePurgeReplies OBJECT-TYPE
 SYNTAX Counter32
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "The number of MPOA Egress Cache Purge Replies transmitted by this MPS. Discontinuities in the value of this counter can occur at re-initialization of the management system, and/or re-initialization of the MPS, and at other times,

```

        as indicated by the value of mpsDiscontinuityTime."
 ::= { mpsStatisticsEntry 21 }

mpsStatTxMpoaTriggers OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The number of MPOA Trigger messages transmitted by this MPS.
    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and/or
    re-initialization of the MPS, and at other times,
    as indicated by the value of mpsDiscontinuityTime."
REFERENCE
    "Multiprotocol Over ATM. AF-MPOA-0087.000.
    Section 4.7.2.1 MPOA Trigger"
 ::= { mpsStatisticsEntry 22 }

mpsStatTxNhrpResolveRequests OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Total count of MPOA Resolution Requests received
    by the Ingress MPS which were translated to NHRP
    Resolution Requests and transmitted to the NHS.
    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and/or
    re-initialization of the MPS, and at other times,
    as indicated by the value of mpsDiscontinuityTime."
 ::= { mpsStatisticsEntry 23 }

mpsStatRxNhrpResolveReplies OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Total count of NHRP Resolution Replies received
    by the Ingress.
    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and/or
    re-initialization of the MPS, and at other times,
    as indicated by the value of mpsDiscontinuityTime."
 ::= { mpsStatisticsEntry 24 }

mpsStatRxNhrpResolveRequests OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Total count of NHRP Resolution Requests received by
    the Egress MPS from the NHS.
    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and/or
    re-initialization of the MPS, and at other times,
    as indicated by the value of mpsDiscontinuityTime."
 ::= { mpsStatisticsEntry 25 }

mpsStatTxNhrpResolveReplies OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current

```

```

DESCRIPTION
    "Total count of NHRP Resolution Replies transmitted
    by the Egress MPS to the NHS.
    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and/or
    re-initialization of the MPS, and at other times,
    as indicated by the value of mpsDiscontinuityTime."
 ::= { mpsStatisticsEntry 26 }

--
-- MPOA Server Protocol support group
--

mpsProtocolTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MpsProtocolEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "List of protocols, per MPS, for which
        MPOA resolution is enabled."
    REFERENCE
        "Multiprotocol Over ATM Version 1.0 (Letter Ballot),
        Section 4.1.1.1 MPS Parameters"
    ::= { mpsObjects 5 }

mpsProtocolEntry OBJECT-TYPE
    SYNTAX      MpsProtocolEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "Each row indicates one protocol for which an MPS
        will perform MPOA resolution.
        If the LECS was contacted for configuration
        information, and the MPS-p3's control octet is set to Enable,
        0x01, then protocol values retrieved from the LECS are reflected
        in this table and the mpsLECSValue object will be (1) true.
        Also, the user or agent can create rows which
        appropriately correspond to the MPS denoted by mpsIndex,
        and the mpsLECSValue object will be set to (2) false.
        NOTE: if the LECS does not return information for
        the MPS-p3 parameter, or if in manual mode, the user or agent
        should create at least one entry for the corresponding MPS.
        Both, LECS and user and/or agent created rows may exist in
        this Table."
    INDEX       { mpsIndex,
                 mpsInternetNetworkLayerProtocol
               }
    ::= { mpsProtocolTable 1 }

MpsProtocolEntry ::= SEQUENCE {
    mpsInternetNetworkLayerProtocol  InternetNetworkAddrType, -- MPS-p3
    mpsLECSValue                      TruthValue,
    mpsProtocolRowStatus                RowStatus
}

mpsInternetNetworkLayerProtocol OBJECT-TYPE

    SYNTAX      InternetNetworkAddrType
    MAX-ACCESS   not-accessible
    STATUS      current

```

```

DESCRIPTION
    "MPS-p3 A protocol on which to perform MPOA resolution."
 ::= { mpsProtocolEntry 1 }

mpsLECSValue OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This object reflects if the current entry is due to
        a retrieval from the LECS or not.  If this entry is
        due to the LECS, then true(1) is the value for this object,
        otherwise, false (2)."

```

```

        "Allows creation, enabling/disabling of this row."
 ::= { mpsMappingEntry 1 }

mpsMappingIndex OBJECT-TYPE
    SYNTAX      MpsIndex
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The mpsMappingIndex of the MPOA Server that is associated with
        this LEC.  The mpsMappingIndex corresponds to the mpsIndex."
 ::= { mpsMappingEntry 2 }

--
-- MPOA Server MPC Information Group
--

mpsMpcTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MpsMpcEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This read-only table contains information about the
        MPCs that these MPSs know about."
 ::= { mpsObjects 9 }

mpsMpcEntry OBJECT-TYPE
    SYNTAX      MpsMpcEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A row created by an MPS.  The MPS learns about the MPC and
        creates a row."
    INDEX       {
                mpsIndex,
                mpsMpcIndex
                }
 ::= { mpsMpcTable 1 }

MpsMpcEntry ::= SEQUENCE {
    mpsMpcIndex      MpcIndex,
    mpsMpcCtrlAtmAddr  AtmAddr
}

mpsMpcIndex OBJECT-TYPE
    SYNTAX      MpcIndex
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The local index for the mpc represented by this entry"
 ::= { mpsMpcEntry 1 }

mpsMpcCtrlAtmAddr OBJECT-TYPE
    SYNTAX      AtmAddr
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "MPC control ATM address "
 ::= { mpsMpcEntry 2 }

--
-- MPOA Server Ingress Cache (Address Resolution) group
--

```

```

mpsIngressCacheTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MpsIngressCacheEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table tracks all the Ingress Cache information
        of the MPSs which this agents manages."
    ::= { mpsObjects 7 }

mpsIngressCacheEntry OBJECT-TYPE
    SYNTAX      MpsIngressCacheEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A entry contains parameters and state variables
        for a row in a MPS's Ingress Cache."
    INDEX       {
        mpsIngressCacheDestInternetworkAddrType,
        mpsIngressCacheDestAddr,
        mpsIndex,
        mpsMpcIndex
    }
    ::= { mpsIngressCacheTable 1 }

MpsIngressCacheEntry ::= SEQUENCE {
    mpsIngressCacheDestInternetworkAddrType  InternetworkAddrType,
    mpsIngressCacheDestAddr                  InternetworkAddr,
    mpsIngressCachePrefixLen                 Integer32,
    mpsIngressCacheEntryState                INTEGER,
    mpsIngressCacheSrcInternetworkAddrType   InternetworkAddrType,
    mpsIngressCacheSrcAddr                   InternetworkAddr,
    mpsIngressCacheSourceMpcCtrlAtmAddr      AtmAddr,
    mpsIngressCacheResolvedAtmAddr           AtmAddr,
    mpsIngressCacheHoldTime                  TimeInterval,
    mpsIngressCacheMpoaRequestId             Integer32,
    mpsIngressCacheNhrpRequestId             Integer32,
    mpsIngressCacheServiceCategory           INTEGER
}

mpsIngressCacheDestInternetworkAddrType OBJECT-TYPE
    SYNTAX      InternetworkAddrType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The type of internetwork layer address of the
        Destination Address."
    ::= { mpsIngressCacheEntry 1 }

mpsIngressCacheDestAddr OBJECT-TYPE
    SYNTAX      InternetworkAddr
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Ingress MPS Destination Internetwork Layer Address."
    ::= { mpsIngressCacheEntry 2 }

mpsIngressCachePrefixLen OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Prefix Length of the mpsIngressCacheDestAddr."
    ::= { mpsIngressCacheEntry 3 }

```

```

mpsIngressCacheEntryState OBJECT-TYPE
    SYNTAX      INTEGER { doesNotExist (1),
                          inactive(2),
                          active(3),
                          negative(4)
                        }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The state of this MPS Ingress cache.
        The states are:
            doesNotExist (1) -- the state is not yet available
            inactive (2) -- state exists, entry is not yet active
                           For example,if using the Finite
                           State Machine in Appendix I.3,
                           the state resolving is 'inactive'.
            active (3) -- state exists, entry is active.
                           For example,if using the Finite
                           State Machine in Appendix I.3,
                           the state resolved
                           is 'active' state.
            negative (4) -- state exists, entry is negative.
                           For example,if using the Finite
                           State Machine in Appendix I.3,
                           the state purging is 'negative'."
    REFERENCE
        "Multiprotocol Over ATM, Letter Ballot, Appendix I.3."
    ::= { mpsIngressCacheEntry 4 }

mpsIngressCacheSrcInternetNetworkAddrType OBJECT-TYPE
    SYNTAX      InternetNetworkAddrType
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The type of internetwork layer address of the
        Source Address."
    ::= { mpsIngressCacheEntry 5 }

mpsIngressCacheSrcAddr OBJECT-TYPE
    SYNTAX      InternetNetworkAddr
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The Ingress MPS Source Internetwork Layer Address."
    ::= { mpsIngressCacheEntry 6 }

mpsIngressCacheSourceMpcCtrlAtmAddr OBJECT-TYPE
    SYNTAX      AtmAddr
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The ATM Address from the source of this MPOA request.
        In other words, the Ingress MPC's Control Atm Address."
    ::= { mpsIngressCacheEntry 7 }

mpsIngressCacheResolvedAtmAddr OBJECT-TYPE
    SYNTAX      AtmAddr
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The ATM Address which has been resolved by an Egress MPC."

```

```

 ::= { mpsIngressCacheEntry 8 }

mpsIngressCacheHoldTime OBJECT-TYPE
    SYNTAX      TimeInterval
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Time interval that this value is valid."
 ::= { mpsIngressCacheEntry 9 }

mpsIngressCacheMpoaRequestId OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The request ID contained in the MPOA resolution request
         from the local Ingress MPC."
 ::= { mpsIngressCacheEntry 10 }

mpsIngressCacheNhrpRequestId OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The request ID which this MPS generates to identify the
         NHRP resolution request."
 ::= { mpsIngressCacheEntry 11 }

mpsIngressCacheServiceCategory OBJECT-TYPE
    SYNTAX      INTEGER (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The service categories supported for this shortcut."
    REFERENCE
        "Lane V2 LUNI TLVs"
 ::= { mpsIngressCacheEntry 12 }

--
-- MPOA Server Egress Cache (Impositions) group
--

mpsEgressCacheTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MpsEgressCacheEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains information regarding the
         Egress MPOA Server Cache Table."
 ::= { mpsObjects 8 }

mpsEgressCacheEntry OBJECT-TYPE
    SYNTAX      MpsEgressCacheEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry represents an entry in the MPS's Egress
         cache Table which keeps track of the state of the
         impositions."
    INDEX      { mpsEgressCacheId,
                 mpsIndex,
                 mpsMpcIndex

```

```

    ::= { mpsEgressCacheTable 1 }

MpsEgressCacheEntry ::= SEQUENCE {
    mpsEgressCacheId                INTEGER,
    mpsEgressCacheDestInternetworkAddrType InternetworkAddrType,
    mpsEgressCacheDestAddr          InternetworkAddr,
    mpsEgressCachePrefixLen         INTEGER,
    mpsEgressCacheHoldTime          TimeInterval,
    mpsEgressCacheEntryState        INTEGER,
    mpsEgressCacheDataLinkHeader    OCTET STRING,
    mpsEgressCacheElanId            Integer32,
    mpsEgressCacheSourceClientAtmAddr AtmAddr,
    mpsEgressCacheNhrpRequestId     Integer32,
    mpsEgressCacheMpoaRequestId     Integer32,
    mpsEgressCacheServiceCategory   INTEGER,
    mpsEgressCacheNextHopInternetworkAddrType InternetworkAddrType,
    mpsEgressCacheNextHopAddr       InternetworkAddr
}

mpsEgressCacheId OBJECT-TYPE
    SYNTAX      INTEGER (1..2147483647)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The id which identifies this cache entry."
    ::= { mpsEgressCacheEntry 1 }

mpsEgressCacheDestInternetworkAddrType OBJECT-TYPE
    SYNTAX      InternetworkAddrType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The destination protocol address type."
    ::= { mpsEgressCacheEntry 2 }

mpsEgressCacheDestAddr OBJECT-TYPE
    SYNTAX      InternetworkAddr
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The destination protocol address."
    ::= { mpsEgressCacheEntry 3 }

mpsEgressCachePrefixLen OBJECT-TYPE
    SYNTAX      INTEGER (0..255)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The destination prefix length."
    ::= { mpsEgressCacheEntry 4 }

mpsEgressCacheHoldTime OBJECT-TYPE
    SYNTAX      TimeInterval
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Time interval that this value is valid."
    ::= { mpsEgressCacheEntry 5 }

```

```

mpsEgressCacheEntryState OBJECT-TYPE
    SYNTAX      INTEGER      {      doesNotExist(1),
                                inactive(2),
                                active(3),
                                negative(4)
                                }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The present states of this MPS egress cache entry.
        The states are:
            doesNotExist (1) -- the state is not yet available
            inactive (2) -- state exists, entry is not yet active
                For example,if using the Finite
                State Machine in Appendix 1.4,
                the state imposing is 'inactive'.
            active (3) -- state exists, entry is active.
                For example,if using the Finite
                State Machine in Appendix 1.4,
                the state imposed
                is 'active' state.
            negative (4) -- state exists, entry is negative.
                For example,if using the Finite
                State Machine in Appendix 1.4,
                the states purging and clearing
                are 'negative'."
    REFERENCE
        "Multiprotocol Over ATM.  AF-MPOA-0087.000.
        Appendix I.4 Egress MPS Control State Machine."
    ::= { mpsEgressCacheEntry 6 }

mpsEgressCacheDataLinkHeader OBJECT-TYPE
    SYNTAX      OCTET STRING  (SIZE (0..255))
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Data-Link Layer Header."
    ::= { mpsEgressCacheEntry 7 }

mpsEgressCacheElanId OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The elan id that this Cache Imposition is sent on."
    ::= { mpsEgressCacheEntry 8 }

mpsEgressCacheSourceClientAtmAddr OBJECT-TYPE
    SYNTAX      AtmAddr
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The Ingress NHC's Atm Address used in the
        original cache imposition."
    REFERENCE
        "Multiprotocol Over ATM.  AF-MPOA-0087.000.
        Page 45."
    ::= { mpsEgressCacheEntry 9 }

mpsEgressCacheNhrpRequestId OBJECT-TYPE
    SYNTAX      Integer32

```

```

MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "The request id from the original NHRP Resolution Request,
    may be only useful in the Resolving State."
 ::= { mpsEgressCacheEntry 10 }

mpsEgressCacheMpoaRequestId OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The new request id which is generated for this imposition's
        request, may be only useful in the Resolving State."
    ::= { mpsEgressCacheEntry 11 }

mpsEgressCacheServiceCategory OBJECT-TYPE
    SYNTAX      INTEGER (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The service categories supported for this shortcut."
    REFERENCE
        "Lane V2 LUNI TLVs"
    ::= { mpsEgressCacheEntry 12 }

mpsEgressCacheNextHopInternetNetworkAddrType OBJECT-TYPE
    SYNTAX      InternetNetworkAddrType
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The NextHop protocol address type."
    ::= { mpsEgressCacheEntry 13 }

mpsEgressCacheNextHopAddr OBJECT-TYPE
    SYNTAX      InternetNetworkAddr
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The NextHop protocol address."
    ::= { mpsEgressCacheEntry 14 }

--
-- Conformance Information
--

mpoaMIBConformance OBJECT IDENTIFIER ::= { mpoaMIB 2 }

mpoaMIBGroups
    OBJECT IDENTIFIER ::= { mpoaMIBConformance 1 }

mpoaMIBCompliances
    OBJECT IDENTIFIER ::= { mpoaMIBConformance 2 }

--
-- Compliance Statements
--

mpoaMpcMibBasicCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION

```

"The basic implementation requirements for SNMP entities which support MPOA Clients."

```

MODULE -- this module
  MANDATORY-GROUPS {
    mpcConfigGroup,
    mpcActualGroup,
    mpcDataAtmAddressGroup,
    mpcStatisticsGroup,
    mpcProtocolGroup,
    mpcMpsGroup,
    mpcMpsMacAddressGroup,
    mpcIngressCacheGroup,
    mpcEgressCacheGroup,
    mpcMpsMultipleMacAddressGroup
  }

OBJECT      mpcRowStatus
  MIN-ACCESS      read-only
  DESCRIPTION
    "Write access is not required."

OBJECT      mpcDataAtmAddressRowStatus
  MIN-ACCESS      read-only
  DESCRIPTION
    "Write access is not required."

OBJECT      mpcProtocolRowStatus
  MIN-ACCESS      read-only
  DESCRIPTION
    "Write access is not required."

--
-- MPC Mapping Group Compliance
--

GROUP mpcMappingGroup
DESCRIPTION
  "This group is mandatory only when there is NOT a one-to-one
  relationship between the MPOA Client and the LANE Client.
  Optionally, a one-to-one relationship between an MPOA Client
  and a LANE Client can be enforced. To enforce this
  one-to-one relationship the lecIndex for the LANE Client
  and the mpcIndex for the MPOA Client must have the same
  value.
  If this one-to-one mapping is enforced, then the implementation
  of the mpcMappingTable is unnecessary. (Since the lecIndex and
  the mpcIndex contain the same value, there is no need to provide
  a mapping of mpcIndex value to lecIndex value.)
  The relationship between MPC and LEC is maintained by
  ensuring that the mpcIndex is the same as the lecIndex
  that is associated with it."

OBJECT      mpcMappingRowStatus
  MIN-ACCESS      read-only
  DESCRIPTION
    "Write access is not required."

OBJECT      mpcMappingIndex
  MIN-ACCESS      read-only
  DESCRIPTION
    "Write access is not required."

```

```

 ::= { mpoaMIBCompliances 1 }

mpoaMpcMibAdvancedCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "The advanced implementation requirements for SNMP entities
    which support MPOA Clients."
  MODULE -- this module
  MANDATORY-GROUPS
    {
      mpoaDeviceTypeGroup,
      mpoaDeviceTypeMpsMacGroup,
      mpcConfigGroup,
      mpcActualGroup,
      mpcDataAtmAddressGroup,
      mpcStatisticsGroup,
      mpcProtocolGroup,
      mpcMpsGroup,
      mpcMpsMacAddressGroup,
      mpcIngressCacheTotalPacketGroup,
      mpcIngressCacheGroup,
      mpcEgressCacheTotalPacketGroup,
      mpcEgressCacheGroup,
      mpcMpsMultipleMacAddressGroup
    }

  OBJECT      mpcRowStatus
  MIN-ACCESS  read-only
  DESCRIPTION
    "Write access is not required."

  OBJECT      mpcDataAtmAddressRowStatus
  MIN-ACCESS  read-only
  DESCRIPTION
    "Write access is not required."

  OBJECT      mpcProtocolRowStatus
  MIN-ACCESS  read-only
  DESCRIPTION
    "Write access is not required."

--
-- MPC Mapping Group Compliance
--

GROUP      mpcMappingGroup
DESCRIPTION
  "This group is mandatory only when there is NOT a one-to-one
  relationship between the MPOA Client and the LANE Client.
  Optionally, a one-to-one relationship between an MPOA Client
  and a LANE Client can be enforced. To enforce this
  one-to-one relationship the lecIndex for the LANE Client
  and the mpcIndex for the MPOA Client must have the same
  value.
  If this one-to-one mapping is enforced, then the implementation
  of the mpcMappingTable is unnecessary. (Since the lecIndex and
  the mpcIndex contain the same value, there is no need to provide
  a mapping of mpcIndex value to lecIndex value.)
  The relationship between MPC and LEC is maintained by
  ensuring that the mpcIndex is the same as the lecIndex
  that is associated with it."

```

```

OBJECT    mpcMappingRowStatus
MIN-ACCESS    read-only
DESCRIPTION    "Write access is not required."

OBJECT    mpcMappingIndex
MIN-ACCESS    read-only
DESCRIPTION    "Write access is not required."

::= { mpoaMIBCompliances 2 }

mpoaMpcMibAdvancedPlusOctetsCompliance MODULE-COMPLIANCE
STATUS    current
DESCRIPTION
    "The AdvancedPlusOctets implementation requirements for SNMP
    entities
    which support MPOA Clients.  This includes supporting
    the 64 bit octet counters."
MODULE -- this module
MANDATORY-GROUPS
    {
        mpoaDeviceTypeGroup,
        mpoaDeviceTypeMpsMacGroup,
        mpcConfigGroup,
        mpcActualGroup,
        mpcDataAtmAddressGroup,
        mpcStatisticsGroup,
        mpcProtocolGroup,
        mpcMpsGroup,
        mpcMpsMacAddressGroup,
        mpcIngressCacheTotalPacketGroup,
        mpcIngressCacheTotalOctetGroup,
        mpcIngressCacheGroup,
        mpcEgressCacheTotalPacketGroup,
        mpcEgressCacheTotalOctetGroup,
        mpcEgressCacheGroup,
        mpcMpsMultipleMacAddressGroup
    }

OBJECT    mpcRowStatus
MIN-ACCESS    read-only
DESCRIPTION    "Write access is not required."

OBJECT    mpcDataAtmAddressRowStatus
MIN-ACCESS    read-only
DESCRIPTION    "Write access is not required."

OBJECT    mpcProtocolRowStatus
MIN-ACCESS    read-only
DESCRIPTION    "Write access is not required."

--
-- MPC Mapping Group Compliance
--

GROUP    mpcMappingGroup
DESCRIPTION
    "This group is mandatory only when there is NOT a one-to-one
    relationship between the MPOA Client and the LANE Client.
    Optionally, a one-to-one relationship between an MPOA Client

```

and a LANE Client can be enforced. To enforce this one-to-one relationship the lecIndex for the LANE Client and the mpcIndex for the MPOA Client must have the same value.

If this one-to-one mapping is enforced, then the implementation of the mpcMappingTable is unnecessary. (Since the lecIndex and the mpcIndex contain the same value, there is no need to provide a mapping of mpcIndex value to lecIndex value.) The relationship between MPC and LEC is maintained by ensuring that the mpcIndex is the same as the lecIndex that is associated with it."

```
OBJECT    mpcMappingRowStatus
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is not required."
```

```
OBJECT    mpcMappingIndex
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is not required."
```

```
::= { mpoaMIBCompliances 3 }
```

```
mpoaMpsMibBasicCompliance MODULE-COMPLIANCE
STATUS    current
DESCRIPTION
    "The implementation requirements for SNMP entities
    which support MPOA Servers."
MODULE -- this module
MANDATORY-GROUPS    {
    mpsConfigGroup,
    mpsActualGroup,
    mpsStatisticsGroup,
    mpsProtocolGroup,
    mpsIngressCacheGroup,
    mpsEgressCacheGroup
}
```

```
OBJECT    mpsRowStatus
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is not required."
```

```
OBJECT    mpsProtocolRowStatus
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is not required."
```

```
--
-- MPS Mapping Group Compliance
--
```

```
GROUP mpsMappingGroup
DESCRIPTION
    "This group is mandatory only when there is NOT a one-to-one
    relationship between the MPOA Server and the LANE Client.
    Optionally, a one-to-one relationship between an MPOA Server
```

and a LANE Client can be enforced. To enforce this one-to-one relationship the lecIndex for the LANE Client and the mpcIndex for the MPOA Server must have the same value.

If this one-to-one mapping is enforced, then the implementation of the mpsMappingTable is unnecessary. (Since the lecIndex and the mpsIndex contain the same value, there is no need to provide a mapping of mpsIndex value to lecIndex value.) The relationship between MPS and LEC is maintained by ensuring that the mpsIndex is the same as the lecIndex that is associated with it."

```
OBJECT      mpsMappingRowStatus
MIN-ACCESS  read-only
DESCRIPTION
    "Write access is not required."
```

```
::= { mpoaMIBCompliances 4 }
```

```
mpoaMpsMibAdvancedCompliance MODULE-COMPLIANCE
STATUS      current
DESCRIPTION
    "The advanced implementation requirements for SNMP entities
    which support MPOA Servers."
MODULE -- this module
MANDATORY-GROUPS
    {
        mpoaDeviceTypeGroup,
        mpoaDeviceTypeMpsMacGroup,
        mpsConfigGroup,
        mpsActualGroup,
        mpsStatisticsGroup,
        mpsProtocolGroup,
        mpsIngressCacheGroup,
        mpsEgressCacheGroup
    }
```

```
OBJECT      mpsRowStatus
MIN-ACCESS  read-only
DESCRIPTION
    "Write access is not required."
```

```
OBJECT      mpsProtocolRowStatus
MIN-ACCESS  read-only
DESCRIPTION
    "Write access is not required."
```

```
--
-- MPS Mapping Group Compliance
```

```
--
```

```
GROUP mpsMappingGroup
DESCRIPTION
    "This group is mandatory only when there is NOT a one-to-one
    relationship between the MPOA Server and the LANE Client.
    Optionally, a one-to-one relationship between an MPOA Server
    and a LANE Client can be enforced. To enforce this
    one-to-one relationship the lecIndex for the LANE Client
```

and the mpcIndex for the MPOA Server must have the same value.
 If this one-to-one mapping is enforced, then the implementation of the mpsMappingTable is unnecessary. (Since the lecIndex and the mpsIndex contain the same value, there is no need to provide a mapping of mpsIndex value to lecIndex value.)
 The relationship between MPS and LEC is maintained by ensuring that the mpsIndex is the same as the lecIndex that is associated with it."

```
OBJECT    mpsMappingRowStatus
MIN-ACCESS    read-only
DESCRIPTION
    "Write access is not required."
```

```
::= { mpoaMIBCompliances 5 }
```

```
--
-- Units of Conformance
--
```

```
mpoaDeviceTypeGroup OBJECT-GROUP
  OBJECTS    {
    deviceTypeLecIndex,
    deviceTypeRemoteLecAtmAddress,
    deviceTypeType,
    deviceTypeMpsAtmAddress,
    deviceTypeMpcAtmAddress
  }
  STATUS    current
  DESCRIPTION
    "A collection of objects which exists when
    the MPOA device learns the MPOA device type and
    MPOA control addresses of neighboring MPOA devices
    using the LANEv2 Device Type TLV."
  ::= { mpoaMIBGroups 1 }
```

```
mpoaDeviceTypeMpsMacGroup OBJECT-GROUP
  OBJECTS    { deviceTypeMpsMacAddress }
  STATUS    current
  DESCRIPTION
    "A collection of objects which is present when the
    MPOA device learns the MPOA device type and MPOA
    control addresses of neighboring MPOA devices using
    the LANEv2 Device Type TLV."
  ::= { mpoaMIBGroups 2 }
```

```
mpcConfigGroup OBJECT-GROUP
  OBJECTS    {
    mpcNextIndex,
    mpcRowStatus,
    mpcConfigMode,
    mpcCtrlAtmAddr,
    mpcSCSetupFrameCount,
    mpcSCSetupFrameTime,
    mpcInitialRetryTime,
    mpcRetryTimeMaximum,
    mpcHoldDownTime
  }
```

```

STATUS    current
DESCRIPTION
    "A collection of objects used for creating and
    configuring MPOA Clients."
 ::= { mpoaMIBGroups 3 }

mpcActualGroup OBJECT-GROUP
OBJECTS    {
    mpcActualState,
    mpcDiscontinuityTime,
    mpcActualConfigMode,
    mpcActualSCSetupFrameCount,
    mpcActualSCSetupFrameTime,
    mpcActualInitialRetryTime,
    mpcActualRetryTimeMaximum,
    mpcActualHoldDownTime
}
STATUS    current
DESCRIPTION
    "A collection of objects describing the status
    and operational parameters of the managed MPC."
 ::= { mpoaMIBGroups 4 }

mpcDataAtmAddressGroup OBJECT-GROUP
OBJECTS    {
    mpcDataAtmAddressRowStatus
}
STATUS    current
DESCRIPTION
    "A collection of objects which describe the
    set of data ATM addresses for the MPCs."
 ::= { mpoaMIBGroups 5 }

mpcStatisticsGroup OBJECT-GROUP
OBJECTS    {
    mpcStatTxMpoaResolveRequests,
    mpcStatRxMpoaResolveReplyAcks,
    mpcStatRxMpoaResolveReplyInsufECResources,
    mpcStatRxMpoaResolveReplyInsufSCResources,
    mpcStatRxMpoaResolveReplyInsufEitherResources,
    mpcStatRxMpoaResolveReplyUnsupportedInetProt,
    mpcStatRxMpoaResolveReplyUnsupportedMacEncaps,
    mpcStatRxMpoaResolveReplyUnspecifiedOther,
    mpcStatRxMpoaImpRequests,
    mpcStatTxMpoaImpReplyAcks,
    mpcStatTxMpoaImpReplyInsufECResources,
    mpcStatTxMpoaImpReplyInsufSCResources,
    mpcStatTxMpoaImpReplyInsufEitherResources,
    mpcStatTxMpoaImpReplyUnsupportedInetProt,
    mpcStatTxMpoaImpReplyUnsupportedMacEncaps,
    mpcStatTxMpoaImpReplyUnspecifiedOther,
    mpcStatTxMpoaEgressCachePurgeRequests,
    mpcStatRxMpoaEgressCachePurgeReplies,
    mpcStatRxMpoaKeepAlives,
    mpcStatRxMpoaTriggers,
    mpcStatRxMpoaDataPlanePurges,
    mpcStatTxMpoaDataPlanePurges,
    mpcStatRxNhrpPurgeRequests,
    mpcStatTxNhrpPurgeReplies,
    mpcStatRxErrUnrecognizedExtensions,
    mpcStatRxErrLoopDetecteds,
    mpcStatRxErrProtoAddrUnreachables,

```

```

        mpcStatRxErrProtoErrors,
        mpcStatRxErrSduSizeExceededs,
        mpcStatRxErrInvalidExtensions,
        mpcStatRxErrInvalidReplies,
        mpcStatRxErrAuthenticationFailures,
        mpcStatRxErrHopCountExceededs
    }
    STATUS    current
    DESCRIPTION
        "A collection of objects that provide statistics
        on the MPOA protocol parameters."
    ::= { mpoaMIBGroups 6 }

mpcProtocolGroup OBJECT-GROUP
    OBJECTS    {
        mpcLECSValue,
        mpcProtocolRowStatus
    }
    STATUS    current
    DESCRIPTION
        "A collection of objects to specify which
        parameters this MPC is enabled for."
    ::= { mpoaMIBGroups 7 }

mpcMappingGroup OBJECT-GROUP
    OBJECTS    {
        mpcMappingRowStatus,
        mpcMappingIndex
    }
    STATUS    current
    DESCRIPTION
        "A collection of objects to map from LEC to MPC"
    ::= { mpoaMIBGroups 8 }

mpcMpsGroup OBJECT-GROUP
    OBJECTS    { mpcMpsAtmAddr }
    STATUS    current
    DESCRIPTION
        "A collection of objects which aid the MPCs to track
        information for all the MPSs which are known by the MPCs."
    ::= { mpoaMIBGroups 9 }

mpcMpsMacAddressGroup OBJECT-GROUP
    OBJECTS    {
        mpcMpsMacAddress
    }
    STATUS    obsolete
    DESCRIPTION
        "A collection of objects which aid the MPCs to track
        MAC Address information for all the MPSs which are known
        by the MPCs."
    ::= { mpoaMIBGroups 10 }

mpcIngressCacheTotalPacketGroup OBJECT-GROUP
    OBJECTS    {
        mpcIngressCacheTxTotalPackets
    }
    STATUS    current
    DESCRIPTION
        "A collection of objects which count the total
        number of packets transmitted over MPC"

```

```

        short cuts."
 ::= { mpoaMIBGroups 11 }

mpcIngressCacheTotalOctetGroup OBJECT-GROUP
OBJECTS {
    mpcIngressCacheTxTotalOctets
}
STATUS current
DESCRIPTION
    "A collection of objects which count the total
    number of octets transmitted over MPC
    short cuts."
 ::= { mpoaMIBGroups 12 }

mpcIngressCacheGroup OBJECT-GROUP
OBJECTS {
    mpcIngressCacheDestInetAddressType,
    mpcIngressCacheDestAddr,
    mpcIngressCachePrefixLen,
    mpcIngressCacheDestAtmAddr,
    mpcIngressCacheSrcAtmAddr,
    mpcIngressCacheEntryState,
    mpcIngressCacheEgressCacheTagValid,
    mpcIngressCacheEgressCacheTag,
    mpcIngressCacheLastNhrpCieCode,
    mpcIngressCacheSigErrCode,
    mpcIngressCacheRetries,
    mpcIngressCacheTimeUntilNextResolutionRequest,
    mpcIngressCacheHoldingTime,
    mpcIngressCacheServiceCategory
}
STATUS current
DESCRIPTION
    "A collection of objects used to monitor the
    MPOA ingress cache."
 ::= { mpoaMIBGroups 13 }

mpcEgressCacheTotalPacketGroup OBJECT-GROUP
OBJECTS {
    mpcEgressCacheRxTotalPackets
}
STATUS current
DESCRIPTION
    "A collection of objects which count the total
    number of packets received by MPC short cuts."
 ::= { mpoaMIBGroups 14 }

mpcEgressCacheTotalOctetGroup OBJECT-GROUP
OBJECTS {
    mpcEgressCacheRxTotalOctets
}
STATUS current
DESCRIPTION
    "A collection of objects which count the total
    number of octets received by MPC short cuts."
 ::= { mpoaMIBGroups 15 }

mpcEgressCacheGroup OBJECT-GROUP
OBJECTS {
    mpcEgressCacheId,
    mpcEgressCacheInetAddressType,
    mpcEgressCacheIDestAddr,

```

```

    mpcEgressCachePrefixLen,
    mpcEgressCacheEntryState,
    mpcEgressCacheEgressCacheTagValid,
    mpcEgressCacheEgressCacheTag,
    mpcEgressCacheHoldTime,
    mpcEgressCacheDataLinkHeader,
    mpcEgressCacheIngressMpcDataAtmAddr,
    mpcEgressCacheLecIndex,
    mpcEgressCacheServiceCategory
}
STATUS    current
DESCRIPTION
    "A collection of objects used to monitor the
    MPOA egress cache."
::= { mpoaMIBGroups 16 }

mpConfigGroup OBJECT-GROUP
OBJECTS    {
    mpsNextIndex,
    mpsRowStatus,
    mpsConfigMode,
    mpsCtrlAtmAddr,
    mpsKeepAliveTime,
    mpsKeepAliveLifeTime,
    mpsInitialRetryTime,
    mpsRetryTimeMaximum,
    mpsGiveupTime,
    mpsDefaultHoldingTime
}
STATUS    current
DESCRIPTION
    "A collection of objects used for creating and
    configuring MPOA Servers."
::= { mpoaMIBGroups 17 }

mpActualGroup OBJECT-GROUP
OBJECTS    {
    mpsActualState,
    mpsDiscontinuityTime,
    mpsActualConfigMode,
    mpsActualKeepAlive,
    mpsActualKeepAliveLifeTime,
    mpsActualInitialRetryTime,
    mpsActualRetryTimeMaximum,
    mpsActualGiveupTime,
    mpsActualDefaultHoldingTime
}
STATUS    current
DESCRIPTION
    "A collection of objects describing the status
    and operational parameters of the managed MPS."
::= { mpoaMIBGroups 18 }

mpStatisticsGroup OBJECT-GROUP
OBJECTS    {
    mpsStatRxMpoaResolveRequests,
    mpsStatTxMpoaResolveReplyAcks,
    mpsStatTxMpoaResolveReplyInsufECResources,
    mpsStatTxMpoaResolveReplyInsufSCResources,
    mpsStatTxMpoaResolveReplyInsufEitherResources,

```

```

        mpsStatTxMpoaResolveReplyUnsupportedInetProt,
        mpsStatTxMpoaResolveReplyUnsupportedMacEncaps,
        mpsStatTxMpoaResolveReplyUnspecifiedOther,
        mpsStatTxMpoaResolveReplyOther,
        mpsStatGiveupTimeExpireds,
        mpsStatTxMpoaImpRequests,
        mpsStatRxMpoaImpReplyAcks,
        mpsStatRxMpoaImpReplyInsufECResources,
        mpsStatRxMpoaImpReplyInsufSCResources,
        mpsStatRxMpoaImpReplyInsufEitherResources,
        mpsStatRxMpoaImpReplyUnsupportedInetProt,
        mpsStatRxMpoaImpReplyUnsupportedMacEncaps,
        mpsStatRxMpoaImpReplyUnspecifiedOther,
        mpsStatRxMpoaImpReplyOther,
        mpsStatRxMpoaEgressCachePurgeRequests,
        mpsStatTxMpoaEgressCachePurgeReplies,
        mpsStatTxMpoaTriggers,
        mpsStatTxNhrpResolveRequests,
        mpsStatRxNhrpResolveReplies,
        mpsStatRxNhrpResolveRequests,
        mpsStatTxNhrpResolveReplies
    }
    STATUS    current
    DESCRIPTION
        "A collection of objects that provide statistics
        on the MPOA Server protocol parameters."
    ::= { mpoaMIBGroups 19 }

mpsProtocolGroup OBJECT-GROUP
    OBJECTS    {
        mpsLECSValue,
        mpsProtocolRowStatus
    }
    STATUS    current
    DESCRIPTION
        "A collection of objects to specify which
        parameters this MPS is enabled for."
    ::= { mpoaMIBGroups 20 }

mpsMappingGroup OBJECT-GROUP
    OBJECTS    {
        mpsMappingRowStatus,
        mpsMappingIndex
    }
    STATUS    current
    DESCRIPTION
        "A collection of objects to map from MPSs to LECs."
    ::= { mpoaMIBGroups 21 }

mpsMpcGroup OBJECT-GROUP
    OBJECTS    { mpsMpcCtrlAtmAddr }
    STATUS    current
    DESCRIPTION
        "A collection of objects which aid the MPSs to track
        information for all the MPCs which are known by the
        MPSs."
    ::= { mpoaMIBGroups 22 }

mpsIngressCacheGroup OBJECT-GROUP
    OBJECTS    {
        mpsIngressCacheDestInternetNetworkAddrType,

```

```

    mpsIngressCacheDestAddr,
    mpsIngressCachePrefixLen,
    mpsIngressCacheEntryState,
    mpsIngressCacheSrcInternetNetworkAddrType,
    mpsIngressCacheSrcAddr,
    mpsIngressCacheSourceMpcCtrlAtmAddr,
    mpsIngressCacheResolvedAtmAddr,
    mpsIngressCacheHoldTime,
    mpsIngressCacheMpoaRequestId,
    mpsIngressCacheNhrpRequestId,
    mpsIngressCacheServiceCategory
}
STATUS    current
DESCRIPTION
    "A collection of objects to monitor the MPS
    ingress cache."
::= { mpoaMIBGroups 23 }

mpsEgressCacheGroup OBJECT-GROUP
OBJECTS    {
    mpsEgressCacheId,
    mpsEgressCacheDestInternetNetworkAddrType,
    mpsEgressCacheDestAddr,
    mpsEgressCachePrefixLen,
    mpsEgressCacheHoldTime,
    mpsEgressCacheEntryState,
    mpsEgressCacheDataLinkHeader,
    mpsEgressCacheElanId,
    mpsEgressCacheSourceClientAtmAddr,
    mpsEgressCacheNhrpRequestId,
    mpsEgressCacheMpoaRequestId,
    mpsEgressCacheServiceCategory,
    mpsEgressCacheNextHopInternetNetworkAddrType,
    mpsEgressCacheNextHopAddr
}
STATUS    current
DESCRIPTION
    "A collection of objects to monitor MPS's egress
    cache parameters."
::= { mpoaMIBGroups 24 }

mpcMpsMultipleMacAddressGroup OBJECT-GROUP
OBJECTS    {
    mpcMpsFlowDetectMacAddress
}
STATUS    current
DESCRIPTION
    "A collection of objects which aid the MPCs to track
    MAC Address information for all the MPSs which are
    used during flow detection by the MPCs."
::= { mpoaMIBGroups 25 }

```

END

Appendix I. State Machine View of MPOA Component Behavior [Informative]

A state machine view of the MPOA component behavior is given in this Section. The state machines shown are intended to give an example of a compliant implementation, but do not represent the complete specification.

I.1 Conventions

Vertical lines represent states, the names of which are labeled above each vertical line. Transitions are horizontal lines. Events are labeled above each transition. Actions are labeled below each transition. The Idle state is shown as a dashed line.

I.2 Ingress MPC Control State Machine

Each instance of the ingress MPC State Machine is defined in the context of an ingress cache entry, namely the <MPS Control ATM Address, Destination Internetwork Layer Address> tuple.

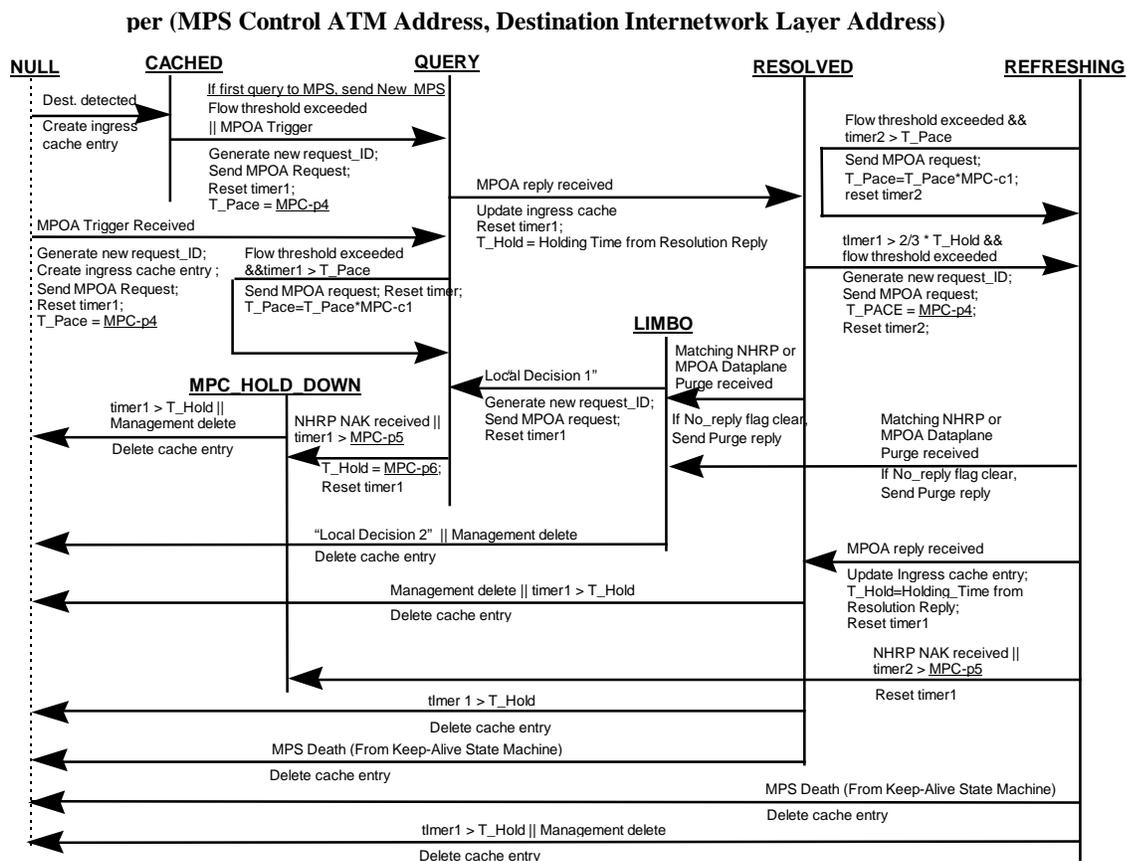


Figure 27 Ingress MPC Control State Machine

Notes on Figure 27:

1. timer1 and timer2 are second timers. Two are required to handle hold times and retry times in the REFRESHING state. Retries are paced until MPC-p5 is reached, in which case the cache entry is deleted when the cache hold time is reached. Alternatively, retries could be paced until the hold time is reached.
2. MPOA trigger stimulates an MPOA request from both NULL and CACHED states.

3. Local Decision 1 and 2 refer to text in Section 4.4.4 on Cache Management

1.3 Ingress MPS Control State Machine

Each instance of the ingress MPS state machine is defined by the <Ingress MPC control ATM Address, Request ID> tuple obtained from the MPOA Resolution Request. The purge description applies to a single cache entry for a single ingress MPC and assumes a higher layer fan-out process.

per (Ingress MPC Control ATM Address, Request ID)

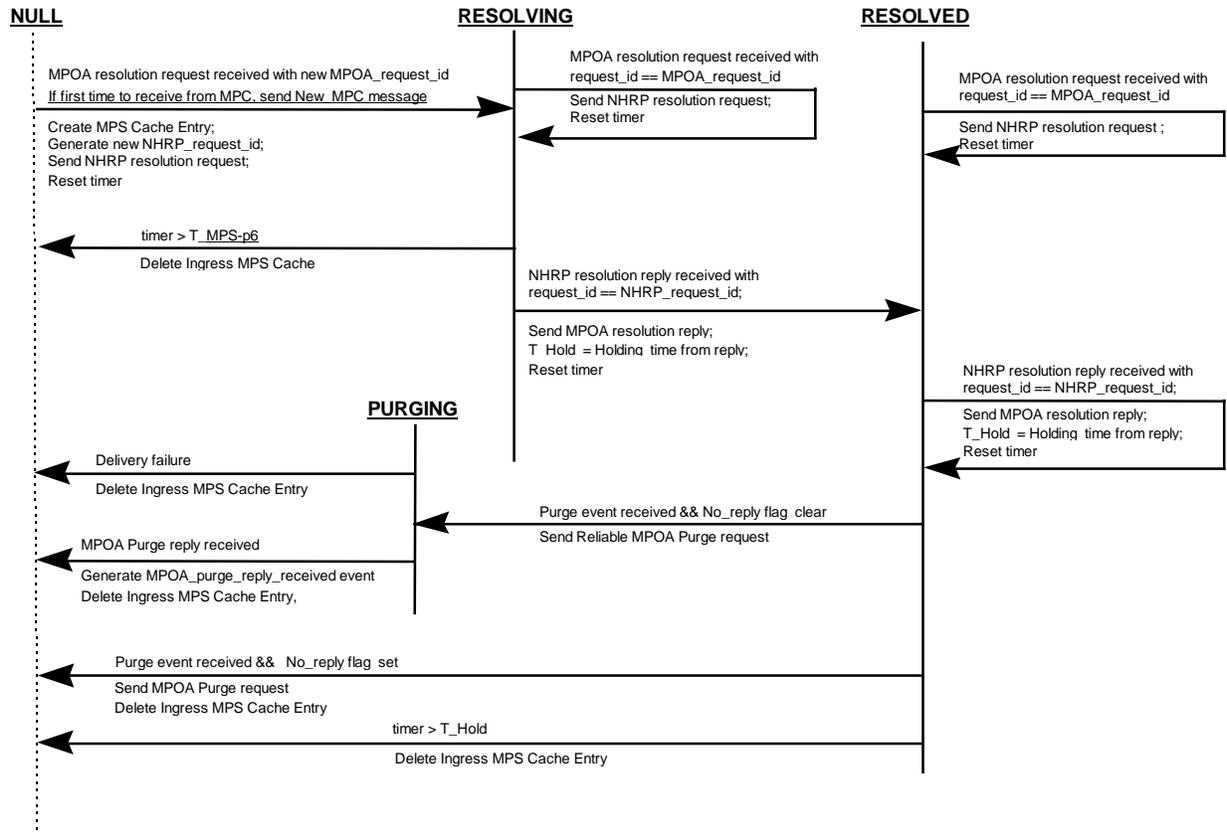


Figure 28 Ingress MPS Control State Machine

Notes on

Figure 28:

1. timer is a second timer
2. Purge events are generated by a process outside the scope of this state machine. This process takes a received NHRP purge request message potentially containing summarized information, and generates events corresponding to instances of this state machine. This process stores state necessary for eventual generation of a NHRP purge reply message.
3. The MPOA_purge_reply_received event is sent from this state machine to the above process. This process uses this event data to determine if and when to generate an NHRP purge reply.
4. The Delivery failure event is generated when reliable delivery fails.

I.4 Egress MPS Control State Machine

per (Source Data ATM Address, Destination Internetwork Layer Address)

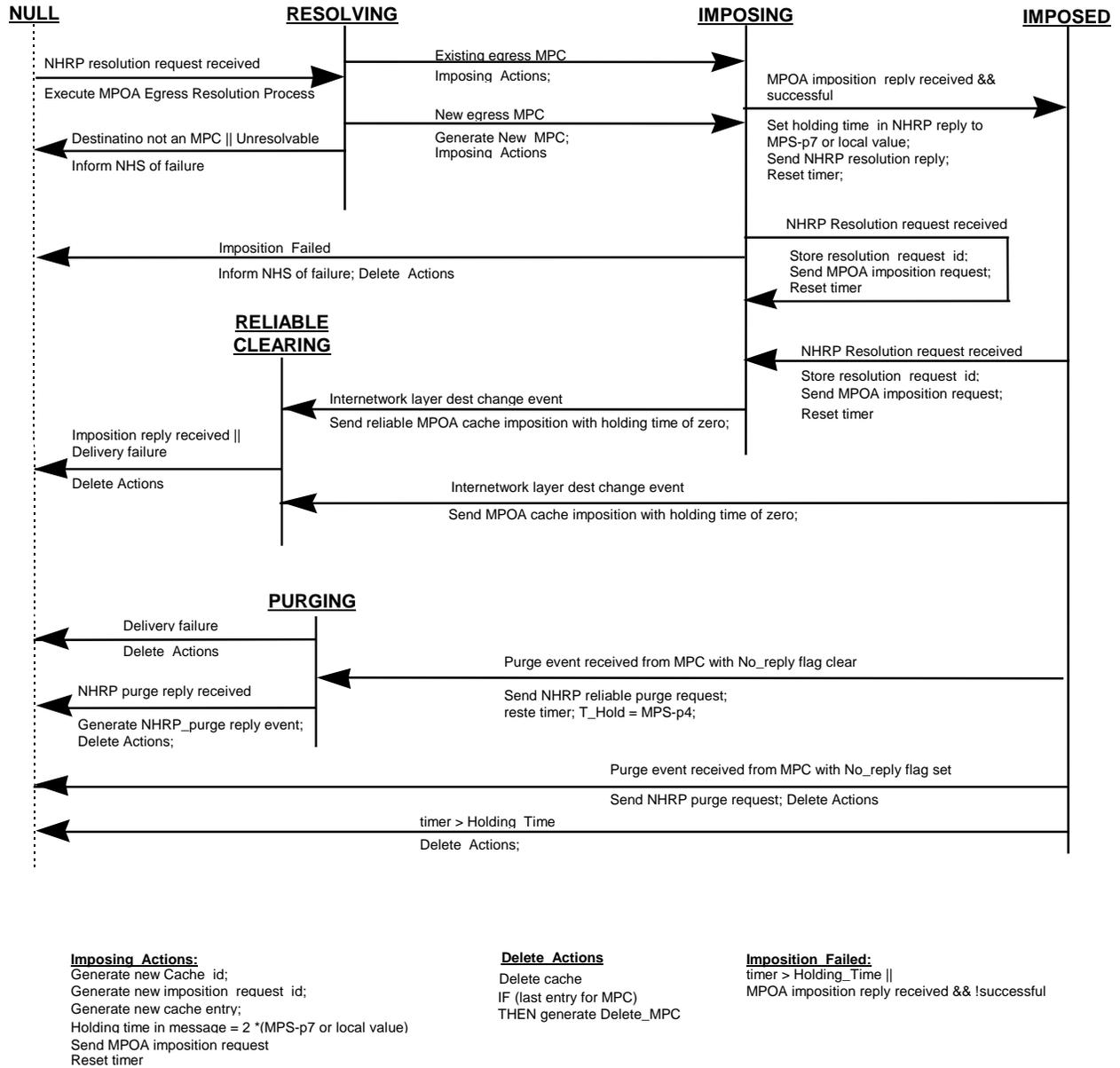


Figure 29 Egress MPS Control State Machine

Notes on Figure 29:

1. timer is a second timer.
2. The purge_event is generated by a process outside the context of this state machine, since the MPOA purge request from the MPS may contain summary information. If the N bit is clear (indicating that a reply is needed), a purge_reply_event is sent to this process when a corresponding NHRP reply is received.
3. The NHS is informed of an imposition failure, so that it may terminate the shortcut if it desires.
4. Internetwork layer dest change event is generated by a process out of the context of this state machine.
5. Holding_Time is derived from either MPS-p7 or 'local information'. Refer to Section 4.1.1.1.

I.5 Egress MPC Control State Machine

(per Cache_ID, MPS Control ATM Address)

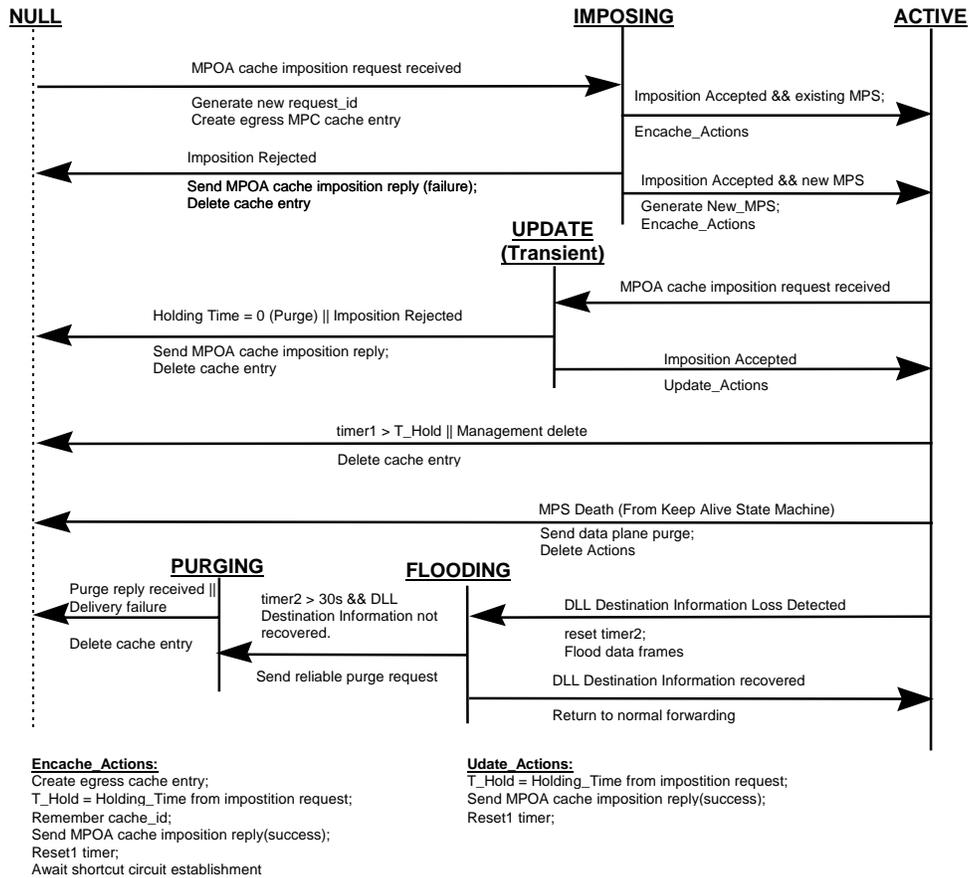


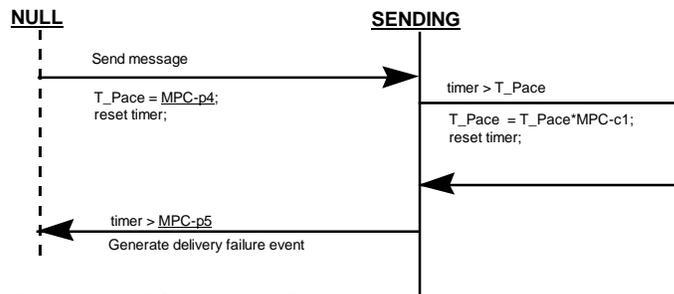
Figure 30 Egress MPC Control State Machine

Notes on Figure 30:

1. Purge event is generated by a process outside the context of this state machine
2. The change in DLL destination state is generated outside the context of this state machine. This event is generated when there is a change in one or more of the following LEC variables: C6, C8, C27, C30

I.6 Reliable Delivery State Machines

Reliable MPC Message Send (per request)



Reliable MPS Message Send (per request)

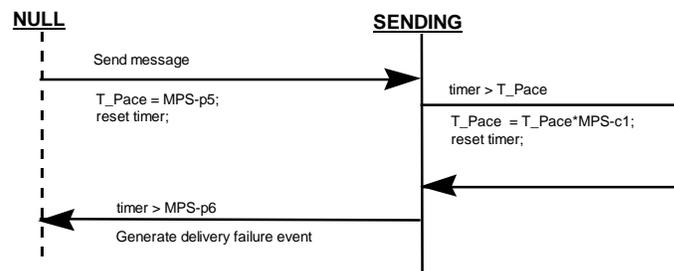


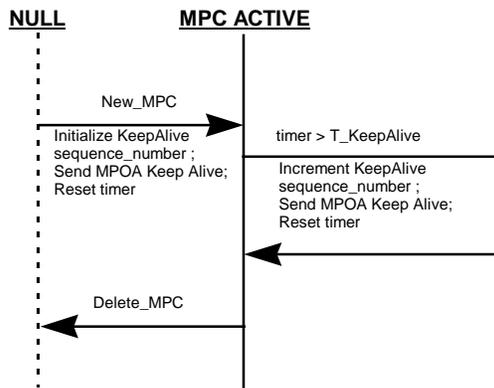
Figure 31 Reliable Delivery State Machines

Notes on Figure 31:

1. timer is a second timer
2. Purge event is generated by a process outside the context of this state machine
3. The change in destination state is generated outside the context of this state machine. This event is generated when there is a change in one or more the following LEC variables: C6, C8, C27, C30

I.7 MPC and MPS Keep-Alive State Machines

MPS Keep Alive (per MPC)



MPC Keep Alive (per MPS)

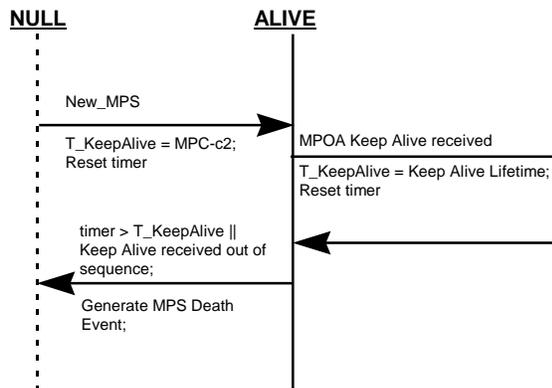


Figure 32 Egress MPC and MPS Keep-Alive State Machines

Notes on Figure 32:

1. timer is a second timer.
2. Purge text described above is from Section 4.5.

Appendix II. Examples of MPOA Control and Data Flows [Informative]

II.1 Scenarios

A simple MPOA network configuration is shown in Figure 33. The MPOA network consists of two ELANs: ELAN-1 and ELAN-2. Each ELAN contains one or more edge devices and MPOA hosts. Each edge device supports one or more LAN hosts.

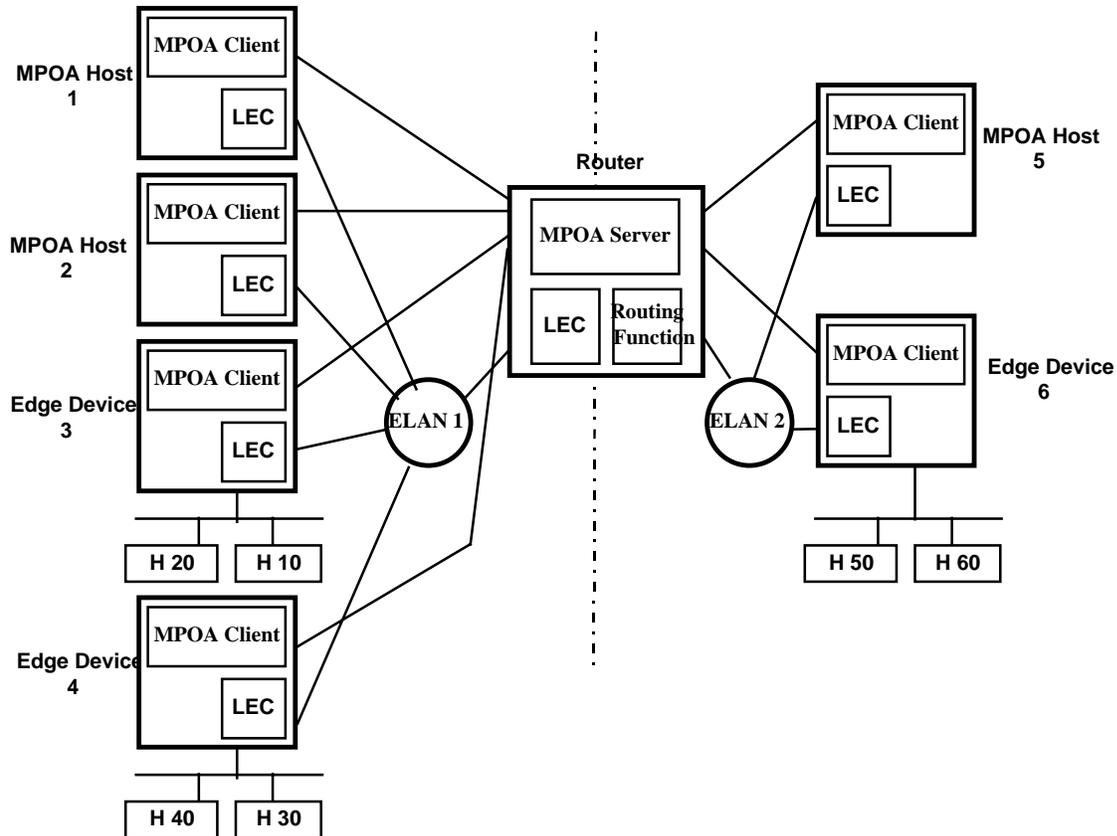


Figure 33 Example Network Configuration

To describe each flow, a source-destination pair (an MPOA host and/or a LAN host) is chosen from Figure 33. The source and destination are chosen within the same ELAN or in different ELANs and the flows are grouped as the intra-ELAN and inter-ELAN flows.

II.1.1 Intra-ELAN Scenarios

Intra-ELAN flows originate from an MPOA host or a LAN host behind an edge device, and flow to an MPOA host or a LAN Host in the same ELAN. These flows use LANE for address resolution and data transfer. The matrix shown in Table 7 illustrates all source-destination pairs with the matrix entry representing the scenario-index. Note that the source and destination are different hosts. The trivial scenario of a LAN-LAN flow on the same edge device is not covered.

Table 7 Intra-ELAN Scenarios

	To MPOA Host	To LAN host
From MPOA Host	(A)	(B)
From LAN host	(C)	(D)

II.1.2 Inter-ELAN Scenarios

The flows listed in Table 8 are between the source-destination pairs for which the source and destination are in different ELANs. These flows may use a default path for short-lived flows or a shortcut for long-lived flows. The default path uses the LANE and router capabilities. The shortcut path uses LANE plus NHRP for address resolution and a shortcut for data transfer.

Table 8 Inter-ELAN Scenarios

	To MPOA Host	To LAN Host
From MPOA Host	(E)	(F)
From LAN Host	(G)	(H)

II.2 Flows

II.2.1 Intra-ELAN

Intra-ELAN flows are illustrated in Figure 34.

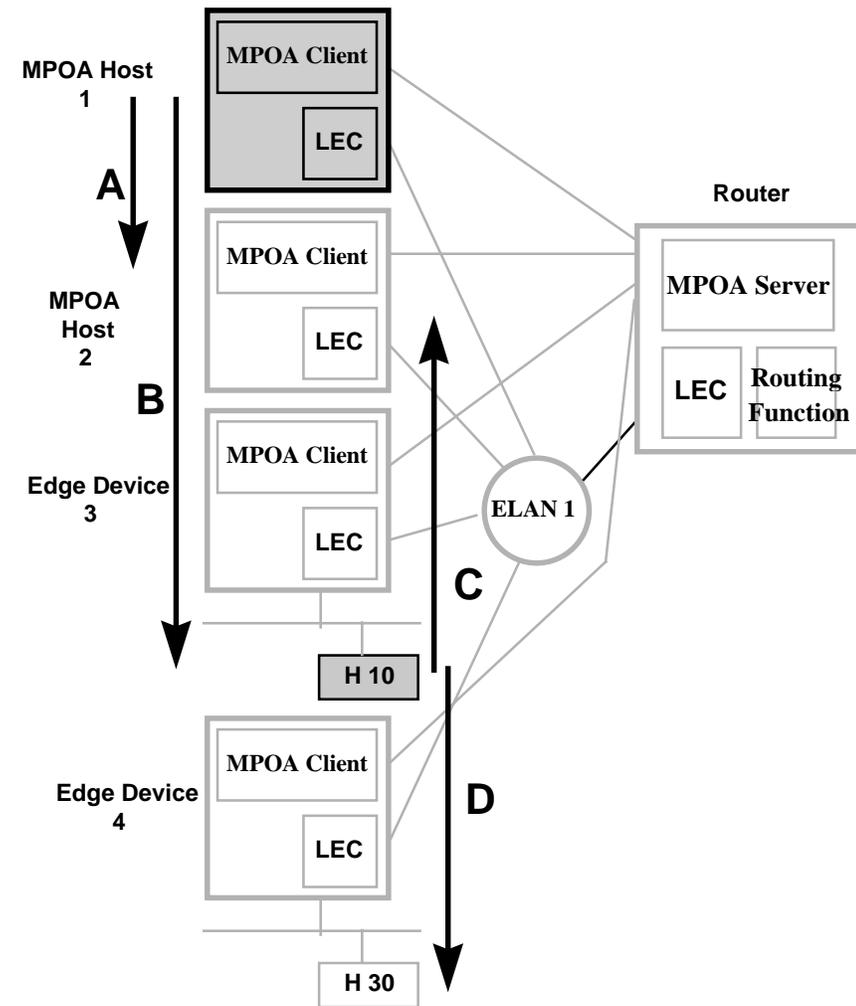


Figure 34 Intra-ELAN Flows

II.2.1.1 From MPOA Host

II.2.1.1.1 Scenario (A): MPOA Host 1 to MPOA Host 2

Figure 35 shows the data path for data originating from MPOA Host 1 and destined to MPOA Host 2 within the same ELAN. LANE is used for such a flow and a Data Direct VCC will carry the LANE frames.

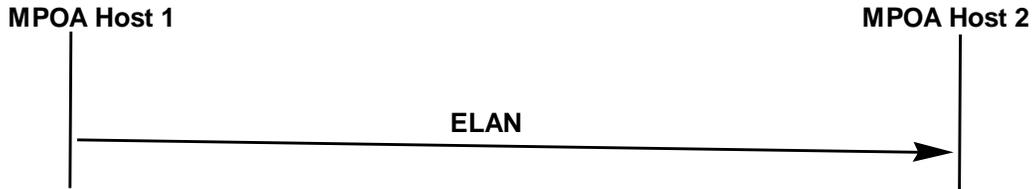


Figure 35 MPOA Host to MPOA Host Data Flow

II.2.1.1.2 Scenario (B): MPOA Host 1 to LAN Host H 10

Figure 36 shows the data path for data originating from MPOA Host 1 and destined to LAN Host H 10 within the same ELAN. LANE is used for such a flow and a Data Direct VCC between MPOA Host 1 and Edge Device 3 will carry the LANE frames.

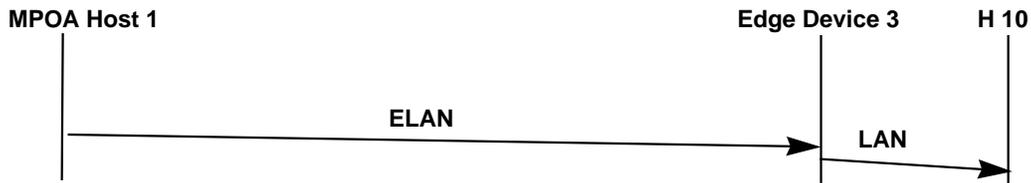


Figure 36 MPOA Host to LAN Host Data Flow

II.2.1.2 From LAN Host

II.2.1.2.1 Scenario (C): LAN Host H 10 to MPOA Host 2

Figure 37 shows the data path for data originating from LAN Host H 10 and destined to MPOA Host 2 within the same ELAN. LANE is used for such a flow and a Data Direct VCC between Edge Device 3 and MPOA Host 2 will carry the LANE frames.

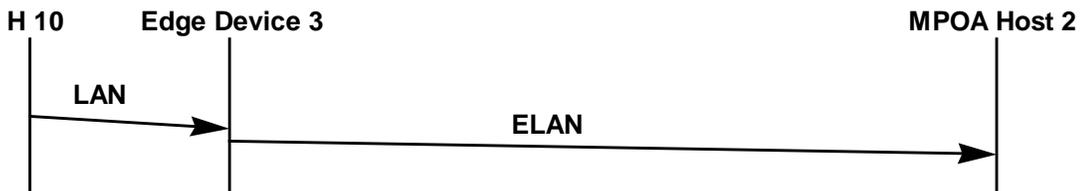


Figure 37 LAN Host to MPOA Host Data Flow

II.2.1.2.2 Scenario (D): LAN Host H 10 to LAN Host H 30

Figure 38 shows the data path for data originating from LAN Host H 10 and destined to LAN Host H 30 within the same ELAN. LANE is used for such a flow and a Data Direct VCC between Edge Device 3 and Edge Device 4 will carry the LANE frames.



Figure 38 LAN Host to LAN Host Data Flow

II.2.2 Inter-ELAN

Inter-ELAN flows are illustrated in Figure 39. For the sake of simplicity, the flows shown in this section only involve a single router/MPS; however, in practice, there can be an arbitrary number of routers on the default routed path between the ingress MPS and egress MPS. The ingress MPS actions (MPOA Resolution Request/Reply) and egress MPS actions (MPOA Cache Imposition Request/Reply) are independent, and in a more complex scenario where multiple routers exist on the default path, the ingress and egress MPSs communicate via NHRP.

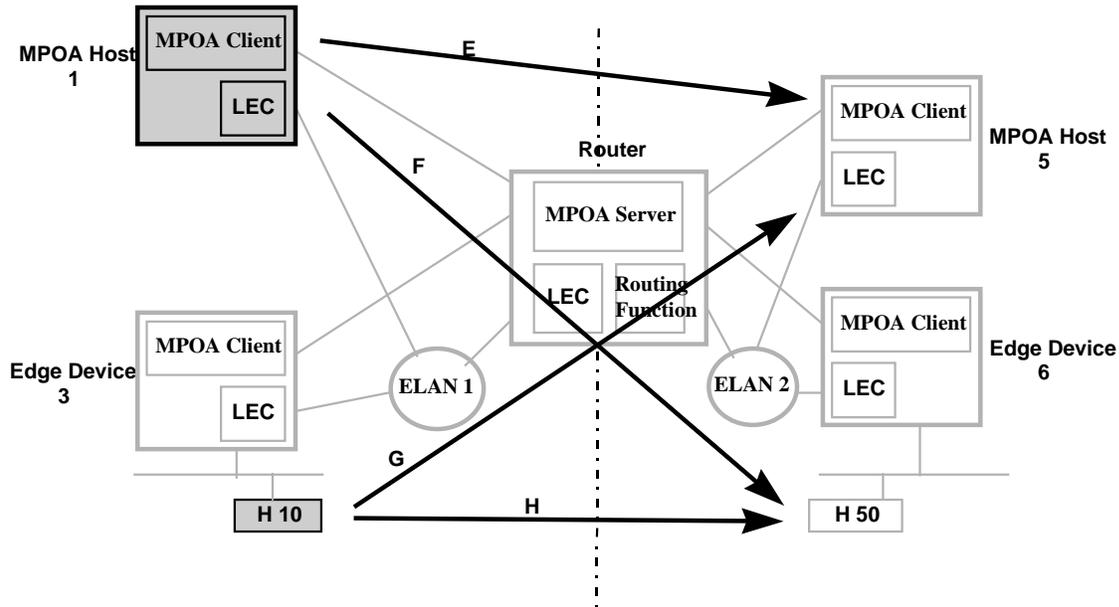


Figure 39 Inter-ELAN Flows

II.2.2.1 From MPOA Host

II.2.2.1.1 Scenario (E): MPOA Host 1 to MPOA Host 5

Figure 40 shows the default and shortcut data paths for data originating from MPOA Host 1 and destined to MPOA Host 5 within a different ELAN.

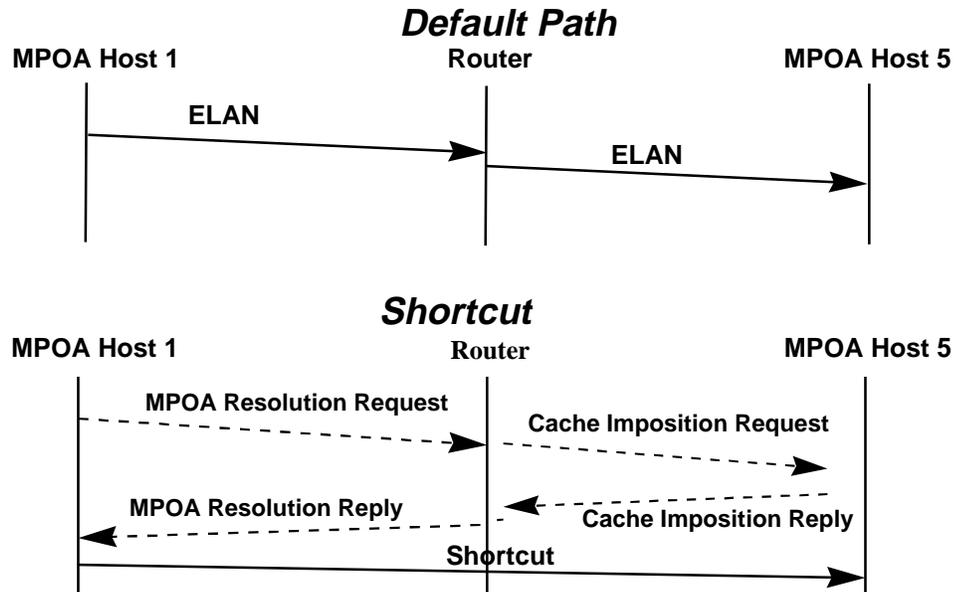


Figure 40 MPOA Host to MPOA Host

Default Path:

MPOA Host 1 sends the packet in a LANE frame to the router via a Data Direct VCC. The router forwards the packet in a LANE frame to MPOA Host 5 via another Data Direct VCC.

Shortcut:

If MPOA Host 1 detects a flow to the internetwork layer address of MPOA Host 5, it sends an MPOA Resolution Request to the MPS to get the corresponding ATM address. The router sends an MPOA Cache Imposition Request to MPOA Host 5 to provide the egress cache entry. MPOA Host 5 sends an MPOA Cache Imposition Reply to the MPS indicating that it can accept the shortcut. The router sends an MPOA Resolution Reply back to MPOA Host 1 with the ATM address of MPOA Host 5. MPOA Host 1 may then update its ingress cache and establish a shortcut to MPOA Host 5.

For subsequent data destined to MPOA Host 5, MPOA Host 1 encapsulates the internetwork layer protocol packet with the appropriate encapsulation for the shortcut. The packets are then sent to MPOA Host 5 using the VCC specified in the ingress cache entry.

II.2.2.1.2 Scenario (F): MPOA Host 1 to LAN Host H 50

Figure 41 shows the default and shortcut data paths for data originating from MPOA Host 1 and destined to LAN Host H 50 within a different ELAN.

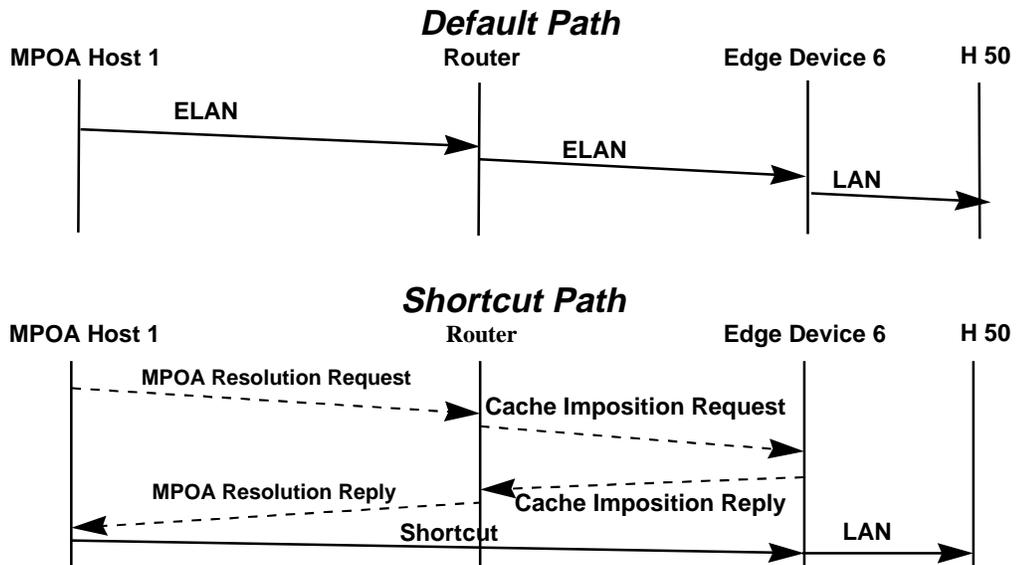


Figure 41 MPOA Host to LAN Host

Default Path:

MPOA Host 1 sends the packet in a LANE frame to the router via a Data Direct VCC. The router forwards the packet in a LANE frame to Edge Device 6 via another Data Direct VCC. Edge Device 6 sends the MAC frame to the LAN Host 50.

Shortcut:

If MPOA Host 1 detects a flow to the internetwork layer address of LAN Host H 50, it sends an MPOA Resolution Request to the MPS to get the corresponding ATM address. The router sends an MPOA Cache Imposition Request to Edge Device 6 to provide the egress cache entry. Edge Device 6 sends an MPOA Cache Imposition Reply to the MPS indicating that it can accept the shortcut. The router sends an MPOA Resolution Reply to MPOA Host 1 with the ATM address of Edge Device 6. MPOA Host 1 may then update its ingress cache and establish a shortcut to Edge Device 6.

For subsequent data destined to LAN Host H 50, MPOA Host 1 encapsulates the internetwork layer protocol packet with the appropriate encapsulation for the shortcut. The packets are then sent to Edge Device 6 using the VCC specified in the cache entry. Edge Device 6 receives the encapsulated packets, makes the MAC frames and sends them to LAN Host 50.

II.2.2.2 From LAN Host**II.2.2.2.1 Scenario (G): LAN Host H 10 to MPOA Host 5**

Figure 42 shows the default and shortcut data paths for data originating from LAN Host H 10 and destined to MPOA Host 5 within a different ELAN.

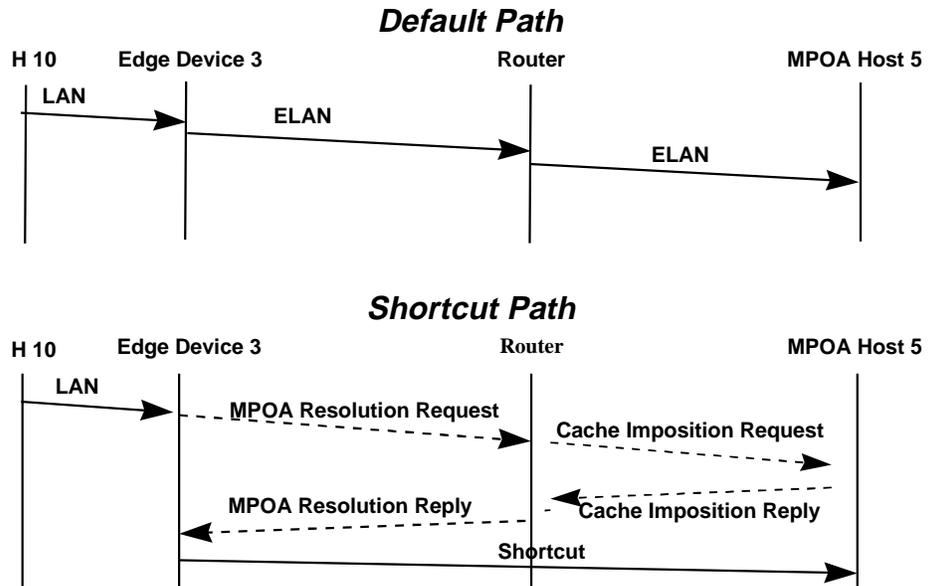


Figure 42 LAN Host to MPOA Host

Default Path:

LAN Host 10 sends the MAC frame to Edge Device 3. Edge Device 3 sends the packet in a LANE frame to the router via a Data Direct VCC. The router forwards the packet in a LANE frame to MPOA Host 5 via another Data Direct VCC.

Shortcut:

LAN Host 10 sends the MAC frame to Edge Device 3. If Edge Device 3 detects a flow to the internetwork layer address of MPOA Host 5, it sends an MPOA Resolution Request to the MPS to get the corresponding ATM address. The router sends an MPOA Cache Imposition Request to MPOA Host 5 to provide the egress cache entry. MPOA Host 5 sends an MPOA Cache Imposition Reply to the MPS indicating that it can accept the shortcut. The router sends an MPOA Resolution Reply to Edge Device 3 with the ATM address of MPOA Host 5. Edge Device 3 may then update its ingress cache and establish a shortcut to MPOA Host 5.

For subsequent data destined to MPOA Host 5, Edge Device 3 encapsulates the internetwork layer protocol packet with the appropriate encapsulation for the shortcut. The packets are then sent to MPOA Host 5 using the VCC specified in the cache entry.

II.2.2.2.2 Scenario (H): LAN Host H 10 to LAN Host H 50

Figure 43 shows the default and shortcut data path for data originating from LAN Host H 10 and destined to LAN Host H 50 within a different ELAN.

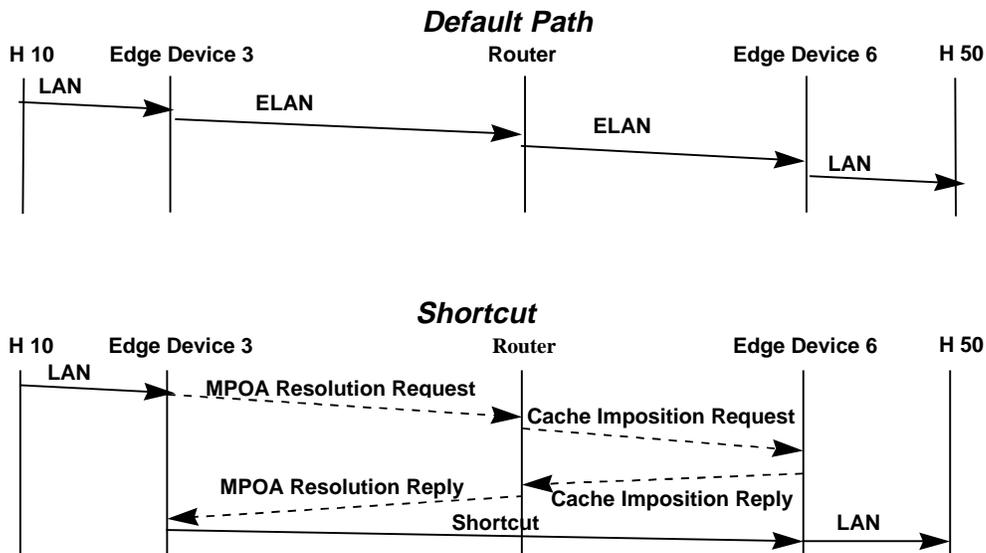


Figure 43 LAN Host to LAN Host

Default Path:

LAN Host 10 sends the MAC frame to Edge Device 3. Edge Device 3 sends the packet in a LANE frame to the router via a Data Direct VCC. The router forwards the packet in a LANE frame to Edge Device 6 via another Data Direct VCC. Edge Device 6 sends the MAC frame to the LAN Host 50.

Shortcut:

LAN Host 10 sends the MAC frame to Edge Device 3. If Edge Device 3 detects a flow to the internetwork layer address of LAN Host H 50, it sends an MPOA Resolution Request to the MPS to get the corresponding ATM address. The router sends an MPOA Cache Imposition Request to Edge Device 6 to provide the egress cache entry. Edge Device 6 sends an MPOA Cache Imposition Reply to the MPS indicating that it can accept the shortcut. The router sends an MPOA Resolution Reply to Edge Device 3 with the ATM address of Edge Device 6. Edge Device 3 may then update its ingress cache and establish a shortcut to MPOA Host 5.

For subsequent data destined to LAN Host H 50, Edge Device 3 encapsulates the internetwork layer protocol packet with the appropriate encapsulation for the shortcut. The packets are then sent to Edge Device 6 using the VCC specified in the cache entry. Edge Device 6 receives the encapsulated packets, makes the MAC frames and sends them to LAN Host 50.

Appendix III. Related Work

[Informative]

The rapid and wide acceptance of ATM has stimulated enormous activity in the communications industry to standardize ATM interfaces. One of the principal objectives of this activity is to enable protocols and applications at the internetwork layer and above to operate effectively over an ATM transport network. Enabling existing protocols and applications to operate over ATM is generally viewed as one of the final, necessary steps to allow the benefits of ATM to be brought gradually into existing networks. Several industry projects address different pieces of the internetwork layer problem, including the LAN Emulation over ATM (LANE) specification under the auspices of the ATM Forum, Classical IP and ARP over ATM defined in RFC 2225 ([IPOA]), the Next Hop Resolution Protocol (NHRP), the Multicast Address Resolution Protocol (MARS), RFC 1483 and other projects under the auspices of the Internet Engineering Task Force (IETF). The APPN Implementers Workshop (AIW) has addressed extensions for the High Performance Routing (HPR) protocol for ATM networks in [AIW]. These projects have all taken a somewhat similar technical approach. The existing protocol stack is either left unchanged (e.g., LANE), or is modified in only minor ways (e.g., Classical IP). These projects have additionally required that any changes or additions can be made only to protocol stacks in systems that have a direct ATM interface, that no changes can be made to the protocol stacks on "LAN systems" (i.e., systems attached to existing subnetworks), and that ATM-attached systems and LAN systems must be fully interoperable.

III.1 LANE

The LANE specification defines a method for an ATM network to emulate an Ethernet or Token-Ring LAN. Protocols, such as IP's Address Resolution Protocol (ARP), that are dependent on the availability of a broadcast function, are supported by LANE over ATM which, due to its connection-oriented nature, is inherently non-broadcast. A host on a LAN that wishes to send data to another host on that LAN using an internetwork layer protocol must determine the MAC (medium access control) address of the destination host prior to data transfer. Protocols such as ARP use broadcast to resolve internetwork layer addresses to MAC Addresses by querying all end-stations on the LAN. On an Emulated LAN (ELAN), LANE supports this broadcast with a Broadcast/Unknown Server (BUS). However, to effect the actual data transfer over an ATM network, a further mapping from MAC Address to ATM address is necessary. Another server, the LANE server (LES), provides this mapping. The originating host then transfers the data by setting up an ATM VCC to the target ATM address.

III.2 Classical IP

The MAC-to-ATM address resolution provided by LANE, while allowing Internetwork and higher-layer protocols to operate as they do on an Ethernet or token ring LAN, involves two levels of resolution prior to data transfer. An internetwork layer address must first be resolved to a MAC Address, and then the MAC Address is mapped to an ATM address. RFC 2225 is the definition of an enhanced IP ARP procedure that resolves internetwork layer addresses directly to ATM addresses. A server, known as the ATMARP server, responds to queries from hosts for internetwork layer addresses with an ATM address. By reducing one step in the process of setting up an ATM connection for data transfer, RFC 2225 helps to minimize broadcast traffic in the subnet. The ATMARP server provides this service to all IP end-stations that are directly attached to the ATM network in a Logical IP Subnet (or LIS, the scope of which is defined in RFC 2225). RFC 2225 applies only to IP, while LANE supports all internetwork layer protocol.

III.3 MARS

Rounding out the suite of protocols for ATM internetwork layer is the IETF's MARS (Multicast Address Resolution Server) specification. MARS is used to resolve internetwork layer multicast and broadcast addresses to either a list of ATM addresses, or to the ATM address of a Multicast Server (MCS) that is responsible for distributing the data to the appropriate end-stations. A MARS serves end-stations in a MARS *cluster*, which is currently equivalent to a LIS. Further study is required before the scope of a cluster is extended beyond a LIS.

III.4 RFC 1483

RFC 1483, Multiprotocol Encapsulation over AAL5, describes encapsulation mechanisms that higher layer protocols can use for transport using ATM Adaptation Layer 5. Data on ATM VCCs established using any of the above methods may be encapsulated using the formats described in RFC 1483.

Appendix IV. Ambiguity at the Edge

[Informative]

IV.1 Ambiguous Encapsulation Information At The Egress MPC

In certain network configurations, traffic associated with two or more distinct flows of data can converge at a single node within the network and be expected to diverge again on leaving this node. In a typical store-and-forward device in a network, this would present no problems. The device would either make forwarding decisions based on internetwork-layer information (e.g., in a router or layer-3 aware switch), or it would leave layer 2 headers intact (e.g., in a bridge).

Because data arriving on an MPOA shortcut VCC does not include an ISO layer 2 header, the impact of temporarily merging distinct data flows may result in a need for distinct cache impositions for each data flow. Merging of flows at the last-hop ATM/MPOA router (MPS) prior to the egress MPC, this router's next hop(s) or only at the egress MPC itself would result in the use of multiple distinct DLL headers for a given internetwork-layer destination. Because the egress MPC is required to prepend the correct DLL header to each data packet received on a shortcut VCC prior to forwarding it on an appropriate port, the egress MPC must be able to distinguish flows using more than the internetwork-layer destination.

IV.2 Resolving Egress Ambiguity

An egress MPC is able to detect when such an ambiguity occurs because it receives a new cache imposition (with a new cache ID) that has the same layer 3 destination and source ATM address as one of its existing cache entries, but a different next hop DLL header. At this point, an MPC implementation should assume that the ingress MPC (or NHC) will re-use a shortcut VCC associated with the existing cache entry. Therefore, the egress MPC must take some action to ensure that it will be able to distinguish packets arriving on such VCCs and make the correct association of cache and flow.

Assuming that the egress MPC is not an active router or layer 3 switch (i.e. it does not have co-resident MPS), the actions that it might take are:

- refuse the cache imposition (force the flow associated with the newer cache to continue to use default forwarding),
- return a distinct ATM address for the new cache imposition or
- assign a tag value in the cache acknowledgment.

The latter option may be used only if the cache imposition includes an MPOA egress cache tag extension indicating that the ingress is prepared to receive a tag in its response and use the tag for all frames transmitted on the shortcut. This is not the case, for example, if the "ingress" is a standard NHC.

IV.3 Ambiguity At The Ingress

The combination of ambiguous use of shortcut VCCs and data-plane purge results in the potential for ambiguous purge messages being received over a shortcut VCC. This will result either when an egress MPC is using tags to distinguish flows or when a single VCC is used to carry several flows and the VCC terminates at an ATM router (NHS or MPS) - which may or may not use tags to distinguish flows. In the worst case, the ingress MPC is forced to be conservative in purging cache entries and reissue MPOA Resolution Request(s) for the layer 3 destination address associated with the purge message received.

Appendix V MPOA-friendly NHRP implementations

[Informative]

This appendix lists some optional extensions and procedures that an NHC/NHS may wish to implement for more efficient interoperation with MPOA devices. These features are not needed for interoperation between NHRP and MPOA devices, but in some cases efficiency may be improved.

1. support the use of the MPOA egress cache tag extension
2. support the use of the ATM Service Category extension
3. always include a non-zero value for MTU size in a Resolution Reply
4. support CPCS-SDU size negotiation during signalling
5. use the same address for Source Protocol address in Resolution Request and shortcut VCC
6. accept control packets on shortcuts (for example, the data plane purge can be sent to an NHC).

Note that the NHRP specification requires correct processing of a purge message received on any VCC; therefore, MPOA data plane purges will be handled correctly.

Appendix VI. MPOA Requirements for Co-Located LEC

[Informative]

To support the MPOA device discovery mechanism described in Section 4.2, LECs must support the functionality described in this section that is not defined in [LANE].

VI.1 Support MPOA Device Type TLV Association

LANE allows the use of multiple ATM addresses by a LEC. The MPOA Device Type TLV is associated with an ATM address of the LEC and, therefore, all MAC addresses behind it. The LE_ASSOCIATE.request interface defined in LANE only supports association of TLVs with single LAN_Destination address. For an MPS in a router with a limited number of MAC addresses, it may be feasible to individually associate the MPOA device type TLV with each of these MAC addresses; however, for an MPC in a bridge that dynamically learns a large number of MAC addresses, it is not likely feasible to individually associate the MPOA device type TLV with each of these dynamically learned MAC addresses. To support the MPOA learning, a LEC should allow the MPOA component to associate the MPOA Device Type TLV with an ATM address and all of the LAN Destinations behind it.

To further facilitate learning, the MPOA Device Type TLV should be carried in all LE_ARP requests originating from a LEC that uses the ATM address associated with the MPOA Device Type TLV as the source ATM address in the LE_ARP.

VI.2 Support for LECs that do Source Learning

It is possible for LECs to learn MAC-to-ATM address mappings by observing data flowing over a LANE Data Direct VCC. When the LEC sees a new source MAC address, it can add the MAC-to-ATM address mapping to its LE_ARP cache. This type of source learning can defeat the MPOA device discovery process unless the MPOA Device Type TLV is associated with the learned MAC addresses properly.

To properly associate the Device Type TLV with learned MAC addresses, a LEC needs to abide by the following rules:

1. For each source ATM address from which a LEC is performing source learning, the LEC must issue at least one LE_ARP_REQUEST for a learned MAC address to determine the MPOA device type of the LEC with the ATM address.
2. The MPOA Device Type TLV received in an LE_ARP_RESPONSE must be associated with all MAC addresses subsequently learned to be associated with the ATM address in the LE_ARP_RESPONSE and not just the MAC address in the LE_ARP_RESPONSE.
3. An LEC that receives an indication of a change of MPOA device type for one MAC address must assume that this change effects all the MAC addresses learned from the same ATM address.

VI.3 Support for LLC Multiplexing

When LLC multiplexing is used, the MPOA device type is associated with the <ELAN-ID,LLC-MUXED-ADDRESS> pair, instead of just the ATM address as described above.

Appendix VII MPOA 1.1 Tip Sheet [informative]

This tip sheet is intended to give some further useful information for implementors of MPOA 1.1.

VII.1 MPS Multinetting

After the completion of the MPOA 1.0 MIB as af-mpoa-0092.000, it was noted that the MIB does not allow a particular configuration (in the following referred to as "MPS Multinetting"), which is not explicitly precluded by MPOA 1.0. It was therefore agreed to correct the MIB accordingly within the context of MPOA 1.1. This Appendix gives some background on the particular "MPS Multinetting" configuration. It should be noted that the use of MPOA 1.1 together with MPOA 1.0 MIB (according to af-mpoa-0092.000) will result in related restrictions for MPS multinetting as shown below.

MPS Multinetting refers to a configuration, where a single MPS with one control ATM address uses multiple MAC addresses on which MPCs can perform flow detection. Such a configuration may be useful in certain scenarios, e.g. in cases where an MPS uses two different IP addresses, but only one control ATM address. A typical scenario is shown by the following Figure:

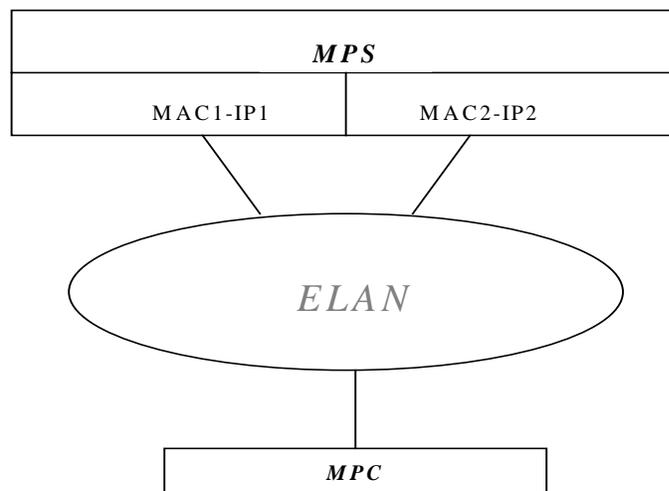


Figure 44: MPS Multinetting

The MPOA 1.0 MIB [af-mpoa-0092.000] precludes such a configuration since the `mpcMpsMacAddressTable` therein is used to only represent a SINGLE MPS MAC address which is used by the MPC in flow detection. In order to no longer preclude MIB support for MPS multinetting, i.e. that MPCs can use more than one MPS MAC address, the MPOA 1.1 MIB therefore corrects this as follows:

- The entire `mpcMpsMacAddressTable` of MPOA 1.0 MIB is obsoleted, i.e. the status for the Table, Entry, Index and Objects will be changed from "current" to "obsolete". This means that the `mpcMpSMacAddressTable` should no longer be supported by existing applications.

- A new table with more appropriate descriptions and the ability to have multiple MPS MAC Addresses for the same MPC, its MPS, and LecIndex has been added. This table is called: "mpcMpsMultipleMacAddressTable".

VII.2 Address Indications in NHRP Ingress Cache Purge

This section outlines why the NHRP Ingress Cache Purge Request / Reply (between I-MPS and I-MPC) indicates the I-MPS internetwork layer address in the "Source Protocol Address Field" (see section 5.3.12, and Annex B). This indication is related to the following scenario / task:

An MPS (MPS-b) is connected to a co-located MPC/MPS with the same ATM address for MPC and MPS, and with shared VCCs between the colocated MPC and MPS. This MPS (MPS-b) shall be able to distinguish between an NHRP Ingress Cache Purge Reply (from the MPC) and an NHRP Purge Reply (from the peer MPS/NHS = MPS-c).

The following figure illustrates this scenario which might have occurred in MPOA 1.0:

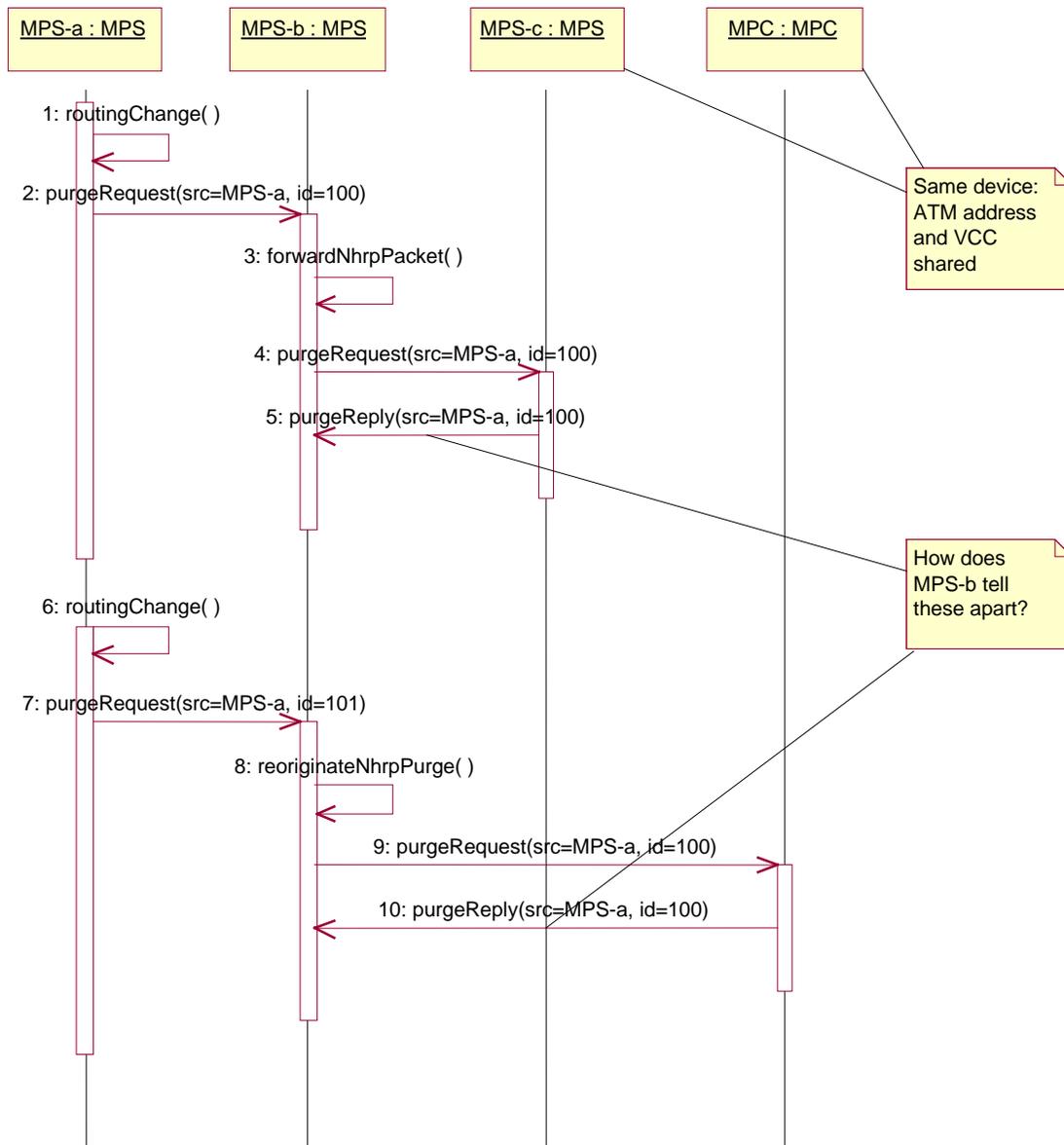


Figure 45: Simultaneous NHRP Purge Replies with same Request ID in MPOA 1.0

(Acronyms: "src" – source address information, "id" – Request ID)

In this example, MPS-b has received an NHRP purge request (7) with a request ID = 101 from its peer MPS (MPS-a). Acting as Ingress MPS for this request, it reoriginates the request as an NHRP Ingress Cache Purge request (9) to the MPC and selects Request ID = 100. MPS-b may also have received another NRHP purge request (2) with the Request ID = 100 which it forwards (4) to MPS-c. Acting as transit NHS in this case, it does not change the Request ID and also uses Request ID = 100 for the forwarded NHRP purge request (4).

Since the Request ID = 100 is used in both cases, the Request ID can't be used by MPS-b to distinguish between the NHRP purge replies it gets back from the MPC and MPS-c. Further, since the co-located MPC and MPS-c use the same ATM address and share VCCs, neither the ATM address nor the VCC can be used for this purpose. Therefore,

the indication of its own (I-MPS-) internetwork layer address as the Source Protocol address in case of the reoriginated NHRP Ingress Cache Purge Request (9) is a means to allow MPS-b to distinguish the related reply of the MPC (10) from the NHRP purge reply of MPS-c (5) where the E-MPS internetwork layer address (in this case: of MPS-a) is indicated instead as the Source Protocol address.

Within MPOA 1.1, it is therefore required that the I-MPS Protocol Address is indicated as the Source Protocol Address in the NHRP Ingress Cache Purge Request / Reply messages to/from the MPC (src=MPS-b, instead of src=MPS-a in MPOA 1.0, for messages (9) and (10) in the example above). Implementors should note that in sections 5.3.12 and Annex B.2 and B.3, not only the Source Protocol Address field has been changed from MPOA 1.0; in order to achieve a harmonized approach, also the Destination Protocol Address field and the Source NBMA Address field have been modified accordingly.

In general, it is required that an MPS allocates unique Request IDs for all NHRP Purge Requests for which it indicates its own internetwork layer address as the Source Protocol address (i.e. in its role as either E-MPS or I-MPS).

Since an NHRP Ingress Cache Purge Reply – unlike in MPOA 1.0 - does not include the internetwork layer address of the E-MPS, it is also recommended that an I-MPS stores the internetwork layer address of an E-MPS which was included in a received NHRP purge request for the case that it has to map a related NHRP Ingress Cache Purge Reply from an MPC to an NHRP purge reply towards that E-MPS.
