Data-Over-Cable Service Interface Specifications

IPv4 and IPv6 eRouter Specification

CM-SP-eRouter-I14-150305

ISSUED

DISCLAIMER

This document is furnished on an "AS IS" basis and neither CableLabs nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein. Any use or reliance on the information or opinion in this document is at the risk of the user, and CableLabs and its members shall not be liable for any damage or injury incurred by any person arising out of the completeness, accuracy, or utility of any information or opinion contained in the document.

CableLabs reserves the right to revise this document for any reason including, but not limited to, changes in laws, regulations, or standards promulgated by various entities, technology advances, or changes in equipment design, manufacturing techniques, or operating procedures described, or referred to, herein.

This document is not to be construed to suggest that any company modify or change any of its products or procedures, nor does this document represent a commitment by CableLabs or any of its members to purchase any product whether or not it meets the characteristics described in the document. Unless granted in a separate written agreement from CableLabs, nothing contained herein shall be construed to confer any license or right to any intellectual property. This document is not to be construed as an endorsement of any product or company or as the adoption or promulgation of any guidelines, standards, or recommendations.

Document Status Sheet

Document Control Number:	CM-SP-eRouter-I14-150305			
Document Title:	IPv4 and IPv6 eRouter Specification			
Revision History:	I01 – 12/7/06 I02 – 2/23/07 I03 – 5/18/07 I04 – 6/11/10 I05 – 2/10/11 I06 – 6/23/11 I07 – 11/17/11 I08 – 03/29/12 I09 – 04/04/13 I10 – 8/08/13 I11 – 11/20/13 I12 – 04/3/14 I13 – 07/29/14 I14 – 03/05/15			
Date:	March 5, 2015			
Status:	~~Work in Progress~~	~~Draft~~	**Issued**	~~Closed~~
Distribution Restrictions:	~~Author Only~~	~~CL/Member~~	~~CL/ Member/ Vendor~~	**Public**

Key to Document Status Codes:

Work in Progress An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.

Draft A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.

Issued A generally public document that has undergone Member and Technology Supplier review, cross-vendor interoperability, and is for Certification testing if applicable. Issued Specifications are subject to the Engineering Change Process.

Closed A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Trademarks:

CableLabs® is a registered trademark of Cable Television Laboratories, Inc. Other CableLabs marks are listed at http://www.cablelabs.com/certqual/trademarks. All other marks are the property of their respective owners.

Contents

Figures

Tables

This page has been left blank intentionally.

1 SCOPE[1]

This specification is part of the DOCSIS family of specifications developed by Cable Television Laboratories, Inc. (CableLabs). This specification was developed for the benefit of the cable industry, and includes contributions by operators and vendors from North America, Europe, and other regions.

1.1 Introduction and Purpose

This specification defines a core set of features that enable multiple subscriber devices to gain access to operator provided high-speed data service using DOCSIS. This core set of features allows for both IPv4- and IPv6-enabled devices to gain connectivity to the Internet.

The eRouter is specified as an Embedded Service/Application Functional Entity (eSAFE) device as defined in [eDOCSIS] that is implemented in conjunction with an embedded DOCSIS cable modem device.

The core set of features defined in this specification includes the ability to provision multiple CPE devices, a description of how to forward data to and from CPE devices, and also the ability to forward IP Multicast traffic to and among CPE devices.

1.2 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

"MUST" This word means that the item is an absolute requirement of this specification.

"MUST NOT" This phrase means that the item is an absolute prohibition of this specification.

"SHOULD" This word means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.

"SHOULD NOT" This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

"MAY" This word means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.

[1] Revised per eRouter-N-13.1127-4 on 3/18/14 by PO.

2 REFERENCES

2.1 Normative References[2]

In order to claim compliance with this specification, it is necessary to conform to all or part of the following standards and other works as indicated, in addition to the other requirements of this specification. Notwithstanding, intellectual property rights may be required to use or implement such normative references.

[CANN DHCP] CableLabs DHCP Options Registry Specification, CL-SP-CANN-DHCP-Reg-I11-141218, December 18, 2014, Cable Television Laboratories, Inc.

[eDOCSIS] eDOCSIS™ Specification, CM-SP-eDOCSIS-I28-150305, March 5, 2015, Cable Television Laboratories, Inc.

[ISO/IEC 29341] Universal Plug and Play Architecture Version 1.1, September 12, 2011.

[MULPIv3.0] DOCSIS MAC and Upper Layer Protocol Interface Specification, CM-SP-MULPIv3.0-I26-150305, March 5, 2015, Cable Television Laboratories, Inc.

[OSSIv3.0] DOCSIS Operations Support System Interface Specification, CM-SP-OSSIv3.0-I25-150305, March 5, 2015, Cable Television Laboratories, Inc.

[RFC 1122] IETF RFC 1122, Requirements for Internet Hosts - Communication Layers, R. Braden, October, 1989.

[RFC 1157] IETF RFC 1157, Simple Network Management Protocol (SNMP, J.D. Case, M. Fedor, M.L. Schoffstall, J. Davin, Simple Network Management Protocol (SNMP), May 1990.

[RFC 1812] IETF RFC 1812, Requirements for IP Version 4 Routers, F. Baker, June 1995.

[RFC 1918] IETF RFC 1918, Address Allocation for Private Internets, Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear, February 1996.

[RFC 2131] IETF RFC 2131, Dynamic Host Configuration Protocol, R. Droms, March, 1997.

[RFC 2132] IETF RFC 2132, DHCP Options and BOOTP Vendor Extensions, S. Alexander, R. Droms, March 1997.

[RFC 2710] IETF RFC 2710, Multicast Listener Discovery (MLD) for IPv6, S. Deering, W. Fenner, B. Haberman, October 1999.

[RFC 2827] IETF RFC 2827, Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing, P. Ferguson, D. Senie, May 2000.

[RFC 2784] IETF RFC 2784, Generic Routing Encapsulation (GRE), D. Farinacci, T. Li, S. Hanks, D. Meyer, P. Traina, March 2000.

[RFC 2863] IETF RFC 2863, The Interfaces Group MIB, K. McCloghrie, F. Kastenholz, June 2000.

[RFC 2890] IETF RFC 2890, Key and Sequence Number Extensions to GRE, G. Dommety, September 2000.

[RFC 3022] IETF RFC 3022, Traditional IP Network Address Translator (Traditional NAT), P. Srisuresh, K. Egevang, January 2001.

[RFC 3203] IETF RFC 3203, DHCP reconfigure extension, Y. T'Joens, C. Hublet, P. De Schrijver, December, 2001.

[2] Revised per eRouter-N-09.0877-2 on 5/11/10 by JB, revised per eRouter N-11.1014-3 on 10/27/11, revised per eRouter N-11.1015-2 on 11/04/11 by JB, revised by eRouter-N-12.1080-4 on 1/4/13 by PO, revised by eRouter-N-13.1108-2 on 7/8/13 by PO; updated per eRouter-N-13.1127-4 on 3/18/14 by PO, updated per eRouter N-14.1150-2 and eRouter N-14.1156-2 on 7/17/14 by PO, revised per eRouter-N-15.1235-3 on 2/13/15 by JB.

[RFC 3315] IETF RFC 3315, Dynamic Host Configuration Protocol for IPv6 (DHCPv6), R. Droms, Ed., J. Bound, B. Volz, T. Lemon, C. Perkins, M. Carney, July 2003.

[RFC 3319] IETF RFC 3319, Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers, H. Schulzrinne, B. Volz, July 2003.

[RFC 3376] IETF RFC 3376, Internet Group Management Protocol, Version 3, B. Cain, S. Deering, I. Kouvelas, B. Fenner, A. Thyagarajan, October, 2002.

[RFC 3412] IETF RFC 3412, Message Processing and Dispatching for the Simple Network Management Protocol (SNMP), J. Case, D. Harrington, R. Presuhn, B. Wijnen, December 2002.

[RFC 3413] IETF RFC 3413, Simple Network Management Protocol (SNMP) Applications, D. Levi, P. Meyer, B. Stuwart, December 2002

[RFC 3415] IETF RFC 3415, View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP), B. Wijnen, R. Presuhn, K. McCloghrie, December 2002.

[RFC 3417] IETF RFC 3417, Transport Mappings for the Simple Network Management Protocol (SNMP), R. Presuhn, December 2002.

[RFC 3419] IETF RFC 3419, Textual Conventions for Transport Addresses, M. Daniels, J. Schoenwaelder, TU Braunschweig, December 2002.

[RFC 3584] IETF RFC 3584, Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework, R. Frye, D. Levi, S. Routhier, B. Wijnen, August 2003.

[RFC 3633] IETF RFC 3633, IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6, O. Troan, R. Droms, December 2003.

[RFC 3646] IETF RFC 3646, DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6), R. Droms, December 2003.

[RFC 3736] IETF RFC 3736, Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6, R. Droms, April 2004.

[RFC 3810] IETF RFC 3810, Multicast Listener Discovery Version 2 (MLDv2) for IPv6, R. Vida, Ed., L. Costa, Ed., June 2004.

[RFC 4022] IETF RFC 4022, Management Information Base for the Transmission Control Protocol (TCP), R. Raghunarayan, March 2005

[RFC 4075] IETF RFC 4075, Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6, V. Kalusivalingam, Cisco Systems, May 2005.

[RFC 4113] IETF RFC 4113, Management Information Base for the User Datagram Protocol (UDP), B. Fenner, J. Flick, June 2005.

[RFC 4191] IETF RFC 4191, Default Router Preferences and More-Specific Routes, R. Draves, D. Thaler, November 2005.

[RFC 4242] IETF RFC 4242, Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6), S. Venaas, T. Chown, B. Volz, November 2005.

[RFC 4291] IETF RFC 4291, IP Version 6 Addressing Architecture, R. Hinden, S. Deering, February 2006.

[RFC 4293] IETF RFC 4293, Management Information Base for the Internet Protocol (IP), S. Routhier, (Editor), Bill Fenner, Brian Haberman, Dave Thaler, April 2006.

[RFC 4361] IETF RFC 4361, Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4), T. Lemon, B. Sommerfeld, February 2006.

[RFC 4443] IETF RFC 4443, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, A. Conta, S. Deering, M. Gupta, Ed., March 2006.

[RFC 4632] IETF RFC 4632, Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan, V. Fuller, T. Li, August 2006.

[RFC 4861] IETF RFC 4861, Neighbor Discovery for IP Version 6 (IPv6), T. Narten, E. Nordmark, W. Simpson, H. Soliman, September 2007.

[RFC 4862] IETF RFC 4862, IPv6 Stateless Address Autoconfiguration, S. Thomson, T. Narten, T. Jinmei, September 2007.

[RFC 4884] IETF RFC 4884, Extended ICMP to Support Multi-Part Messages, R. Bonica, D. Gan, D. Tappan, C. Pignataro, April 2007.

[RFC 5389] IETF RFC 5389, Session Traversal Utilities for NAT (STUN), J. Rosenberg. R, Mahy, P. Matthews, D. Wing, October 2008.

[RFC 5494] IETF RFC 5494, IANA Allocation Guidelines for the Address Resolution Protocol (ARP), J. Arkko Ericsson, C. Pignataro, April 2009

[RFC 5942] IETF RFC 5942, IPv6 Subnet Model: The Relationship Between Links and Subnet Prefixes, H. Singh, W. Beebee, E. Nordmark, July 2010.

[RFC 6092] IETF RFC 6092, Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service, J. Woodyatt, Ed., January 2011.

[RFC 6106] IETF RFC 6106, IPv6 Router Advertisement Options for DNS Configuration, November 2010.

[RFC 6298] IETF RFC 6298, Computing TCP's Retransmission Timer, V. Paxson, M. Allman, J. Chu, M. Sargent, June 2011

[RFC 6333] IETF RFC 6333, Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion, A. Durand, R. Droms, J. Woodyatt, Y. Lee, August 2011.

[RFC 6334] IETF RFC 6634, Dynamic Host-Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite, D.Hankins, T. Mrugalski, August 2011.

[RFC 6540] IETF RFC 6540, IPv6 Support Required for All IP-Capable Nodes. W. George, C. Donley, C. Liljenstolpe, L. Howard, April 2012.

[RFC 6633] IETF RFC 6633, Deprecation of ICMP Source Quench Messages, F. Gont, May 2012

[RFC 6644] IETF RFC 6644, Rebind Capability in DHCPv6 Reconfigure Messages. D. Evans, R. Droms, S. Jiang, July 2012.

[RFC 6762] IETF RFC 6762, Multicast DNS, S. Cheshire, M. Krochmal, Apple Inc., February 2013.

[RFC 6918] IETF RFC 6918, Formally Deprecating Some ICMPv4 Message Types, F. Gont, C. Pignataro, April 2013

[RFC 7083] IETF RFC 7083, Modification to Default Values of SOL_MAX_RT and INF_MAX_RT, R. Droms, November 2013.

[RFC 7084] IETF RFC 7084, Basic Requirements for IPv6 Customer Edge Routers, H. Singh, W. Beebee, C. Donley, B. Stark, November 2013.

[RFC 792] IETF RFC 792, Internet Control Message Protocol, J. Postel, September 1981.

[RFC 826] IETF RFC 826, An Ethernet Address Resolution Protocol, David C. Plummer, November, 1982.

[RFC 868] IETF RFC 868, Time Protocol, J. Postel & K. Harrenstien, May 1983.

[SECv3.0] DOCSIS Security Specification, CM-SP-SECv3.0-I15-130808, August 8, 2013, Cable Television Laboratories, Inc.

[TR-064] TR-064 LAN-Side DSL CPE Configuration Specification, May 2004, Broadband Forum Technical Report.

[TR-069a4] TR-069 CPE WAN Management Protocol v1.2, Issue 1 Amendment 2, July 2011, Broadband Forum Technical Report.

[TR-143 Corrigendum1] TR-143 Enabling Network Throughput Performance Tests and Statistical Monitoring, Issue 1, Corrigendum 1, December 2008, Broadband Forum Technical Report.

[TR-181] TR-181 Device Data Model for TR-069, Issue 2 Amendment 8, November 2013, Broadband Forum Technical Report.

2.2 Informative References[3]

This specification uses the following informative references.

[MR-230] TR-069 Deployment Scenarios, Issue 1, MR-230, August 2010, Broadband Forum Marketing Report.

[RFC 1323] IETF RFC 1323, TCP Extensions for High Performance, Jacobson, V., Braden, B., and D. Borman, May 1992.

[RFC 2460] IETF RFC 2460, Internet Protocol, Version 6 (IPv6) Specification, Deering, S. and R. Hinden, December 1998.

[RFC 3775] IETF RFC 3775, Mobility Support in IPv6, Johnson, D., Perkins, C., and J. Arkko, June 2004.

[RFC 3828] IETF RFC 3828, The Lightweight User Datagram Protocol (UDP-Lite), Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E., and G. Fairhurst, July 2004.

[RFC 3879] IETF RFC 3859, Deprecating Site Local Addresses, C. Huitema, B. Carpenter, September 2004.

[RFC 4007] IETF RFC 4007, IPv6 Scoped Address Architecture, Deering, S., Haberman, B., Jinmei, T., Nordmark, E., and B. Zill, March 2005.

[RFC 4193] IETF RFC 4193 Unique Local IPv6 Unicast Addresses. R. Hinden, B. Haberman, October 2005.

[RFC 4302] IETF RFC 4302, IP Authentication Header, Kent, S., December 2005.

[RFC 4303] IETF RFC 4303, IP Encapsulating Security Payload (ESP). S. Kent. December 2005.

[RFC 4340] IETF RFC 4340, Datagram Congestion Control Protocol (DCCP), Kohler, E., Handley, M., and S. Floyd, March 2006.

[RFC 4960] IETF RFC 4960, Stream Control Transmission Protocol, Stewart, R., September 2007.

[RFC 5095] IETF RFC 5095, Deprecation of Type 0 Routing Headers in IPv6, Abley, J., Savola, P., and G. Neville-Neil, December 2007.

[RFC 5156] IETF RFC 5156, Special-Use IPv6 Addresses, Blanchet, M., April 2008.

[RFC 5201] IETF RFC 5201, Host Identity Protocol, Moskowitz, R., Nikander, P., Jokela, P., and T. Henderson, April 2008.

[3] Revised per eRouter-N-12.1081-2 on 1/4/13 by PO, revised per eRouter-N-13.1088-2 on 3/5/13 by PO.

[RFC 5382] IETF RFC 5382, NAT Behavioral Requirements for TCP, Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, October 2008.

[RFC 5996] IETF RFC 5996, Internet Key Exchange Protocol Version 2 (IKEv2), Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, September 2010.

[RFC 793] IETF RFC 793/STD-7, Transmission Control Protocol, Postel, J., September 1981.

[TR-106a5] TR-106 Data Model Template for TR-069-Enabled Devices, Issue 1, Amendment 5, November 2010, Broadband Forum Technical Report.

[WI-FI MGMT] Wi-Fi Provisioning Framework Specification, WR-SP-WiFi-MGMT-I05-141201, December 1, 2014, Cable Television Laboratories, Inc.

2.3 Reference Acquisition[4]

- Broadband Forum, 48377 Fremont Blvd, Suite 117 Fremont, CA 94538 USA;
  Phone: +1-510-492-4020; Fax: +1-510-492-4001; http://www.broadband-forum.org
- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027 USA;
  Phone:+1-303-661-9100; Fax:+1-303-661-9199; http://www.cablelabs.com
- Internet Engineering Task Force (IETF) Secretariat, 48377 Fremont Blvd., Suite 117, Fremont, CA 94538 USA;
  Phone: +1-510-492-4080; Fax: +1-510-492-4001, http://www.ietf.org/
- International Organization for Standardization, http://www.iso.org/iso

[4] Revised per eRouter N-11.1014-3 on 10/27/11 by JB.

3 TERMS AND DEFINITIONS[5]

This specification uses the following terms:

Customer-Facing Interface An eRouter interface used for connecting CPE devices. Defined in [RFC 7084] as a Local Area Network (LAN) Interface, the Customer-Facing Interface is represented by a physical port.

Customer-Facing IP Interface An IP Interface connected to the eRouter that is not necessarily mapped one-to-one with the number of Customer-Facing ports on the eRouter. As defined in [RFC 7084], this is an IP LAN interface in which one or many physical ports are associated with an IP address.

Customer-Facing Logical Interface A logical Interface connected to the eRouter that is not necessarily mapped one-to-one with the number of Customer-Facing ports on the eRouter. As defined in [RFC 7084], this is a LAN interface in which one or more physical ports are associated with a logical interface, such as a VLAN.

Down Interface An interface on a router that is further away from the ISP network than the ‘Up’ interface on that same router.

eRouter An eSAFE device that is implemented in conjunction with the DOCSIS embedded cable modem.

Hard Reset Describes a full reset of the eDOCSIS device and its constituent eSAFE application elements (such as the eRouter) and embedded CM.

Internet Gateway Device A remotely managed gateway device as defined in CPE WAN Management Protocol [TR-069a4].

Link ID 16 bits of both IPv4 and IPv6 addresses chosen to uniquely identify each "link" or LAN segment (Customer-Facing IP Interface) within the home network. Counting from the left, the Link ID includes bits 49 - 64 (fourth 16-bit block) in an IPv6 address and bits 9 - 24 (middle two octets) in an IPv4 address.

Multicast Subscription Database A simple table of entries for the IPv4 or IPv6 Multicast Group Membership information maintained by the eRouter on respective interfaces. Implementation details for storage of records are completely vendor-defined.

Operator-Facing Interface The eRouter interface that is connected to the Embedded cable modem. As defined in [RFC 7084], this is a Wide Area Network (WAN) interface. In CPE WAN Management Protocol (CWMP) this is called an upstream interface.

Operator-Facing IP Interface IP Interface that is connected to the Embedded Cable Modem and is provisioned with an IP Address provided by the Operator. As defined in [RFC 7084], this is a WAN interface.

Prefix A common address component, which defines a portion of a network. The meanings of the terms Prefix and Subnet are interchangeable. The term Prefix is favored in this document.

Reset Describes a routine in which the operational state is interrupted by the instruction to shutdown and restart. The term is synonymous with the terms re-initialization and reboot. The term can describe either a full device reset (a Hard Reset) or the re-initialization of an individual eSAFE's software application (a Soft Reset) and any associated routines necessary to notify connected clients or other nodes of the device becoming temporarily unavailable.

[5] Revised per eRouter-N-09.0877-2 on 5/11/10 by JB, revised per eRouter N-11.1014-3 on 10/27/11 and by per eRouter N-11.1015-2 on 11/04/11 by JB; revised per eRouter-N-13.1108-2 on 7/10/13 by PO; revised per eRouter-N-13.1127-4 on 3/20/14 by PO. Revised per eRouter-N-14.1155-3 on 2/10/15 by JB.

Soft Reset Describes a reset operation in which the software layer of the eRouter eSAFE application is re-initialized without impacting other eSAFEs or the embedded CM within an eDOCSIS device.

Subnet A portion of a network that shares a common address component. The meanings of the terms Prefix and Subnet are interchangeable. The term Prefix is favored in this document.

TR-069 Term used to refer to the CPE WAN Management Protocol suite defined in [TR-069a4].

TR-069 CPE Term used to refer to the CPE managed using the CPE WAN Management Protocol suite defined in [TR-069a4].

Up Interface A router interface that connects to another router that is closer to the ISP network. For example, the ‘Up Interface’ of an Internal router is the port used to connect to the CFI (down interface), of the eRouter.

4 ABBREVIATIONS AND ACRONYMS[6]

This specification uses the following abbreviations:

ACS	Auto-configuration Server
AFTR	Address Family Translation Router
AH	Authentication Header
ALG	Application Layer Gateway
ARP	Address Resolution Protocol
ASN.1	Abstract Syntax Notation One
B4	Basic-Bridging BroadBand
BCP	Best Current Practice
CER	Customer Edge Router
CER-ID	Customer Edge Router-Identification
CM	Cable Modem
CMTS	Cable Modem Termination System
CPE	Customer Premises Equipment (includes internal routers)
CWMP	CPE WAN Management Protocol
DAD	Duplicate Address Detection
DCCP	Datagram Congestion Control Protocol
DHCPv4	Dynamic Host Configuration Protocol version 4
DHCPv6	Dynamic Host Configuration Protocol version 6
DNS	Domain Name Service
DUID	DHCP unique identifier
EAE	Early Authentication and Encryption
DUID-EN	DUID Enterprise Number
DUID-LL	DUID Link Layer address
DUID-LLT	DUID Link Layer plus Time
eCM	embedded Cable Modem
eSAFE	Embedded Service/Application Functional Entity
ESP	Encapsulating Security Protocol
EUI	Extended Unique Identifier
FTP	File Transfer Protocol
GNAP	Global Network Address Port
GUA	Global Unique Address
IANA	Internet Assigned Numbers Authority

[6] Revised per eRouter N-11.1014-3 on 10/27/11 by JB, updated per eRouter N-14.1150-2 and eRouter N-14.1156-on 7/17/14 by PO. Revised per eRouter-N-14.1155-3 on 2/10/15 and per eRouter-N-14.1233-1 on 2/13/15 by JB.

IA_NA	Identity Association for Non-temporary Addresses
IA_PD	Identity Association for Prefix Delegation
ICMP	Internet Control Message Protocol
ID	Identifier
IGMP	Internet Group Management Protocol
IKE	Internet Key Exchange
IPTV	Internet Protocol Television
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IRT	Initial Retransmission Times
LAN	Local Area Network
LLC	Logical Link Control
MAC	Media Access Control
MIB	Management Information Base
mDNS	Mulitcast Domain Name System
MLD	Multicast Listener Discovery
MoCA	Mulitmedia over Coax Alliance
MRC	Maximum Retransmission Count
MRD	Maximum Retransmission Duration
MRT	Maximum Retransmission Time
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
NAPT	Network Address Port Translation
NAT	Network Address Translation
ND	Neighbor Discovery
OID	Object ID
ORCHIDv2	Overlay Routable Cryptographic Task Identifiers version 2
OUI	Organization Unique Identifier
PNAP	Private Network Address Port
QoS	Quality of Service
RA	Router Advertisement
RD	Router Discovery
RFC	Request For Comment
RG	Residential Gateway
SCTP	Stream Control Transmission Protocol
SIP	Session Initiation Protocol

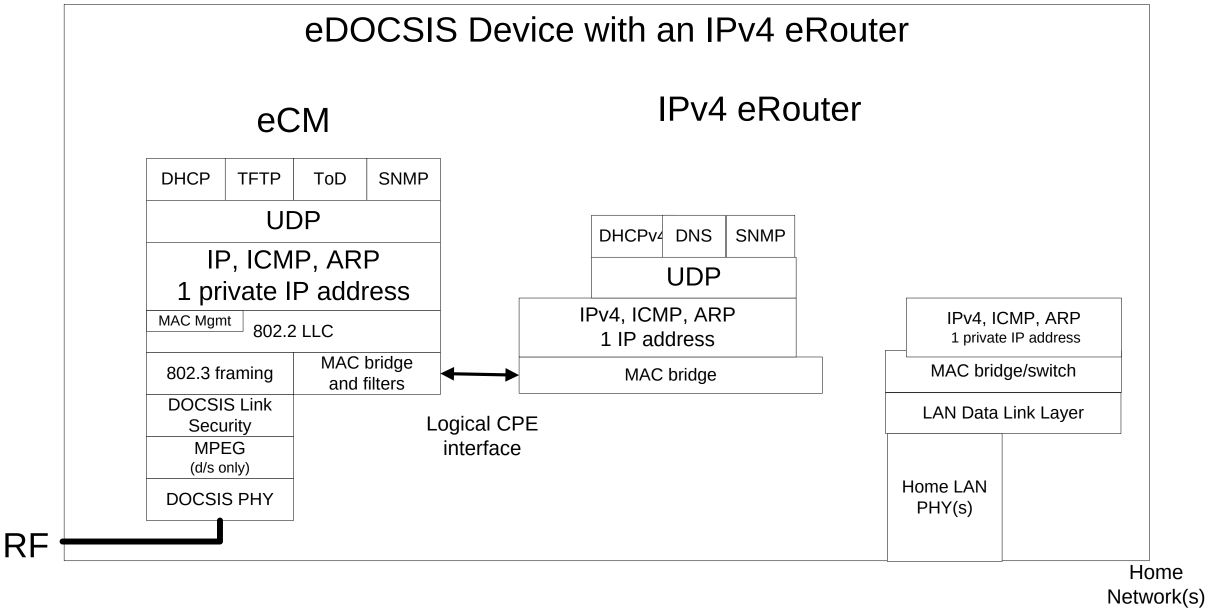
SLAAC	Stateless Address Autoconfiguration
SNMP	Simple Network Management Protocol
SNTP	Simple Network Time Protocol
SSDP	Simple Service Delivery Protocol
TCP	Transmission Control Protocol
TLV	Type/Length/Value
ToS	Type of Service
TTL	Time To Live
UDP	User Datagram Protocol
ULA	Unique Local Addresses
USB	Universal Serial Bus
VLAN	Virtual Local Area Network
WAN	Wide Area Network

5 THEORY OF OPERATION[7]

The eRouter device is intended to provide networking functionality in conjunction with an embedded DOCSIS eCM in an eDOCSIS device.

This specification defines a set of features for an eRouter that is in one of three protocol enabled modes: IPv4 protocol enabled, IPv6 protocol enabled, or dual protocol enabled.

The figures below depict implementations of an eDOCSIS eRouter device in each of the three enabled modes.



[7] Revised per eRouter-N-09.0877-2 on 5/11/10 by JB, revised per eRouter N-11.1014-3 on 10/27/11 by JB; revised per eRouter-N-13.1127-4 on 3/20/14 by PO. Revised per eRouter-N-14.1155-3 on 2/10/15 and per eRouter-N-14.1162-2 by JB.

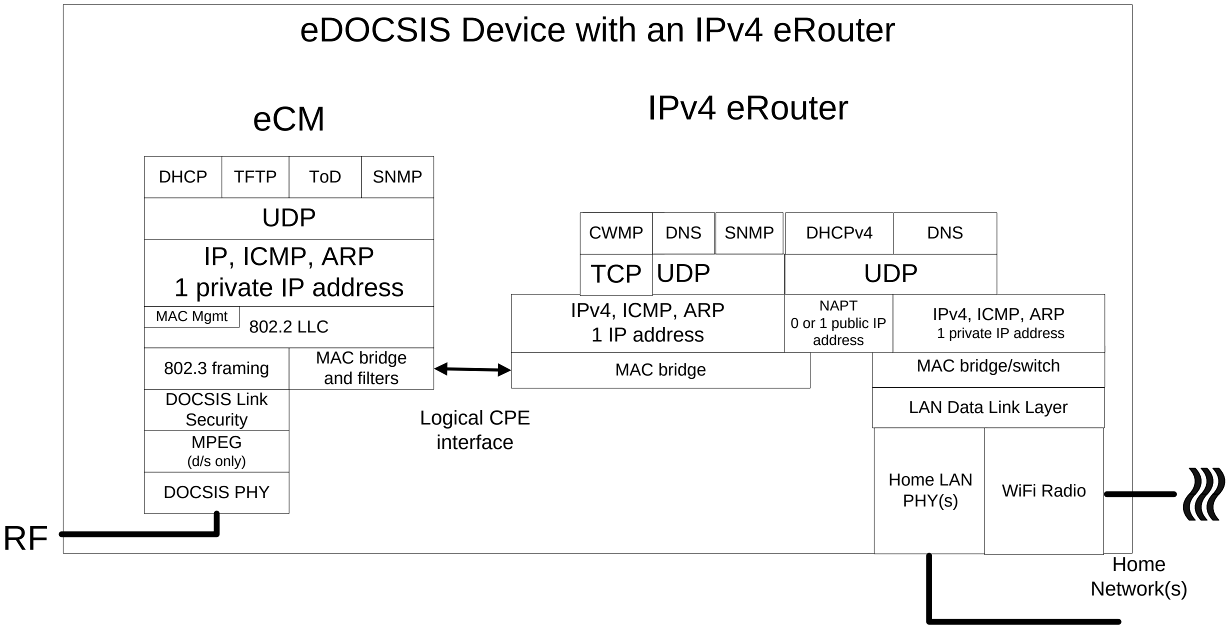


Figure 5-1 - Logical Components of an eDOCSIS device with an IPv4 Protocol Enabled eRouter

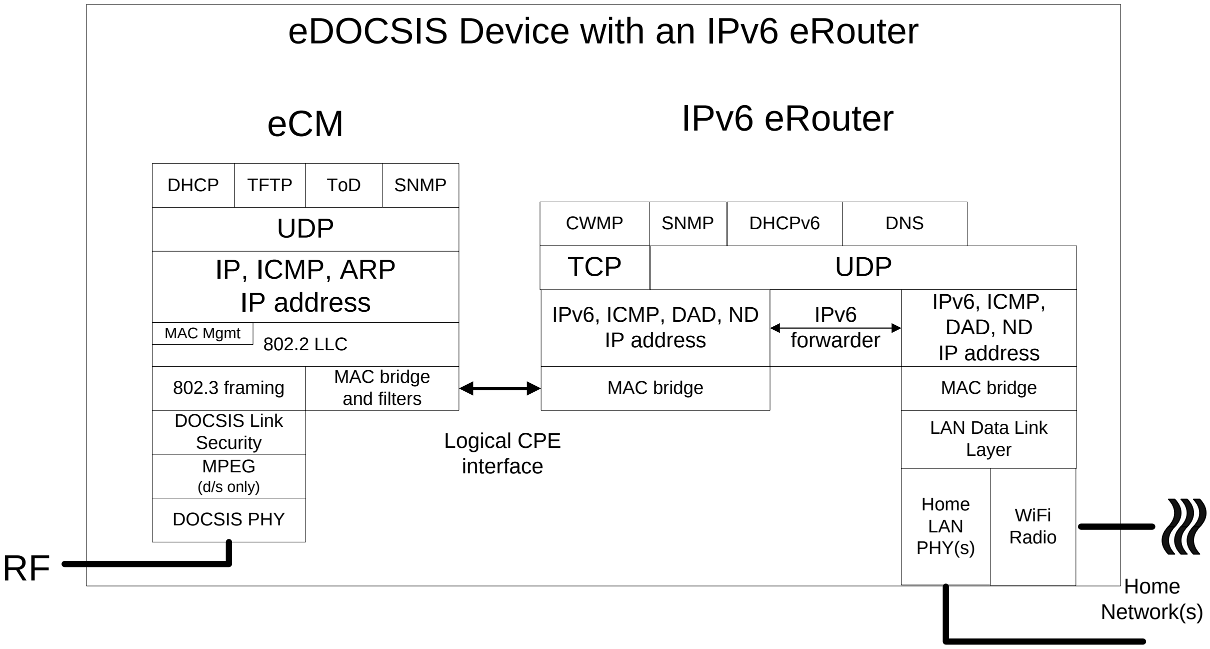


Figure 5-2 - Logical Components of an eDOCSIS device with an IPv6 Protocol Enabled eRouter

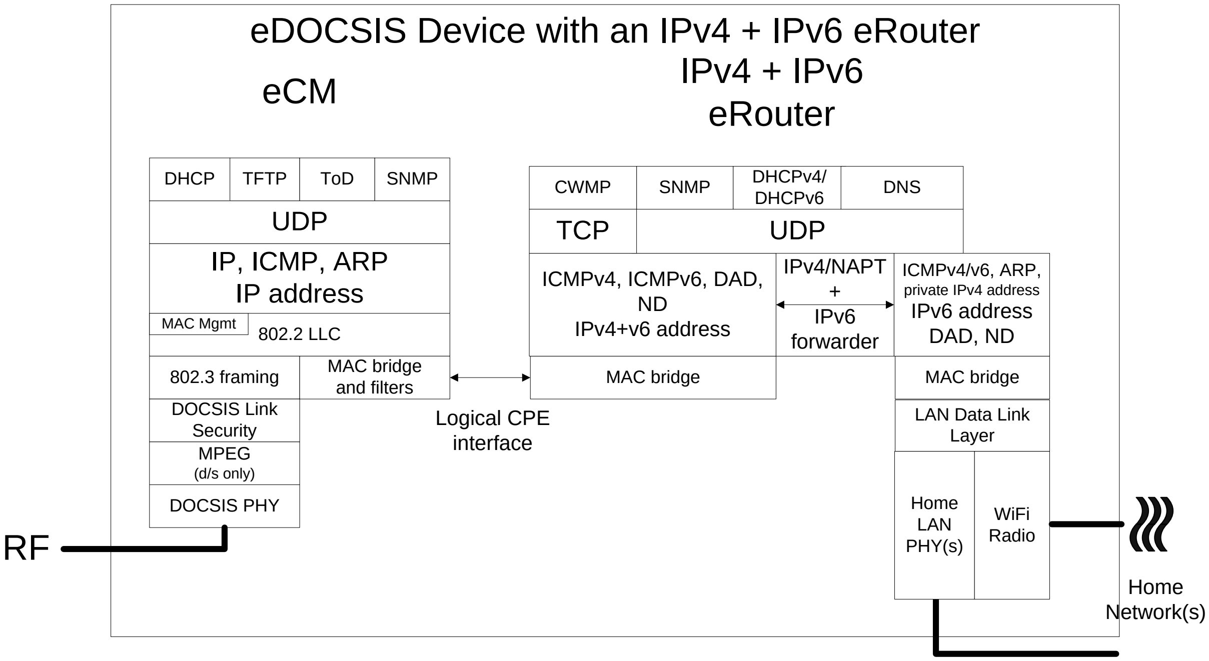


Figure 5-3 - Logical Components of an eDOCSIS device with a Dual Protocol Enabled eRouter

The primary function of the eRouter device is to allow subscribers to connect multiple CPE devices to the operator-provided DOCSIS high-speed Internet service. DOCSIS specifications allow subscribers to directly connect multiple CPE devices to the cable modem; however, that requires operators to provide IP provisioning to each of the CPE devices. The eRouter is delegated the responsibility of provisioning multiple CPE devices at the subscriber end. Depending on which IP Protocols are enabled, the eRouter allows provisioning of IPv4 CPEs, IPv6 CPEs, or both IPv4 and IPv6 CPEs simultaneously.

This specification defines the core set of functions that are performed by the eRouter; however, in most implementations, vendors include additional features and functionality that enhance the eRouter device.

The specification defines: a) CPE provisioning with IPv4 and IPv6 addresses, b) IPv4 data forwarding with and without NAPT and IPv6 data forwarding, c) ability to forward IP Multicast traffic, and d) preserving IP QoS markings on IP data to and from the CPE devices.

The eRouter specification defines two methods that an eRouter uses when assigning IP Addresses to Customer Facing Interfaces. The methods are Link-ID and non Link-ID. Both methods implement IPv4 addressing using RFC-1918 IPv4 address ranges, but in different ways.

The method, called Link-ID, provides a predictable IPv4 addressing scheme, where IPv6 link bits are reflected in IPv4 octets 2 and 3, and enabling native IPv4 routing within the home. This functionality allows for routing across multiple routers without the need for routing protocols or multiple routers running NAPT.

However, if Link-ID is not enabled or an IPv6 prefix is not available from which to generate a Link-ID, then IPv4 routing across multiple routers will not be possible without manual configuration.

One overall goal, is that after a reset or reboot, CFI IPv4 addressing should follow the address scheme implemented by the eRouter before the reset/reboot event.

Another overall goal is to bring up the IPv4 LAN interfaces quickly after reboot/reset/power-cycle to provide some LAN functionality even if the OFI is non-operational for any reason.

This specification uses the terms Customer-Facing Interface and Operator-Facing Interface as defined in Section 3.

This specification defines requirements for an eRouter device with a single Operator-Facing IP Interface. This specification defines requirements for an eRouter device with a single one or more Customer-Facing logical IP Interfaces that are not necessarily mapped one-to-one with the number of Customer-Facing ports on the eRouter.

This specification defines SNMP [RFC 3412] and TR-069 CWMP [TR-069a4] as the Operator-Facing management interface options for eRouter.

5.1 eDOCSIS eRouter and TR-069 Architecture[8]

This section defines TR-069 requirements for the eRouter management architecture, which are derived from the [eDOCSIS] specification.

The TR-069 specification suite defines the Device 2.x entity in [TR-181]. It refers to a CPE device management space for holding the device itself and root of other services specifications (e.g., VoIP, Storage, IPTV, etc.). See [MR-230] for more details on TR-069 deployment scenarios.

Both eDOCSIS and TR-069 architectures define two equivalent components:

- Access Modem: eDOCSIS defines the eCM; TR-069 may accommodate any access technology.
- Services: eDOCSIS defines eSAFEs; TR-069 defines CPE Services.

The cable modem is not referred to as a CPE in [MULPIv3.0] and [eDOCSIS]. Only devices attached to the Customer-Facing Interfaces of a cable modem are termed CPEs in [MULPIv3.0] and [eDOCSIS]. In TR-069, all devices located in the customer premises are considered CPEs. For the eRouter case, the term CPE has the same meaning within DOCSIS and TR-069; that is, the eRouter eSAFE is considered to be a 'CPE' under both the DOCSIS and TR-069 definitions, whereas the eCM is not. However, in this specification the eSAFE term is used when referring to the eRouter.

The main differences between both architectures are:

- A TR-069 Device 2.x [TR-181] is a TR-069 enabled CPE such as Residential Gateways (RGs) and other type of network devices (e.g., Access Modem). Different services can be implemented on a TR-069 Device. The Access Modem could be part of the device itself, by modeling it as an upstream interface of the entire TR-069 CPE, or the device contains only CPE services. In eDOCSIS the eCM is the Access Modem and eRouter is an application or functional entity (eSAFE). DOCSIS specifications define CMs and eSAFEs such as eRouters (embedded eRouters within an eCM).
- The management of eSAFEs in eDOCSIS is separated from the eCM. In TR-069 the management of services is integrated with the CPE device management.

TR-069 allows the transparent integration of access network technologies within the RG and CPE Services by combining the multiple components and their respective management data. The TR-069 device can either configure and monitor the Access Modem managed elements, simply report the Access Modem status and configuration, or do nothing with the Access Modem. The latter is the case of TR-069 in the context of eDOCSIS where the eCM is managed and provisioned independently of any eSAFE supporting TR-069 management.

Figure 5-4 shows the alignment of eDOCSIS, eRouter, and TR-069 device architectures where the reuse of the TR-069 protocol stack and data models for eDOCSIS devices such as eRouter can be seen. A general purpose eSAFE is shown for illustration purposes. The main difference between both models is the separation of the CM bridge of the internal WAN/LAN bridging function at the eRouter compared to the integrated TR-069 Device 2.x.

Figure 5-4 is based on the "Simple Router Example (Interfaces Visualized)" figure of [TR-181]. In Figure 5-4, the stack layers are seen as interfaces per [TR-181], physical interfaces (e.g., Ethernet, SSID, Wi-Fi Radio), bridges, ports, Bridges, Ethernet Link interfaces (LLC), and IP Interfaces. The Operator-Facing TR-069 Etherlink Interfaces correspond to the eDOCSIS Logical CPE Interfaces (LCI). IP additional interfaces can represent IP Tunnels and other IP forwarding models.

[8] Added per eRouter N-11.1014-3 on 10/27/11 by JB; revised per eRouter-N-13.1127-4 on 3/20/14 by PO.

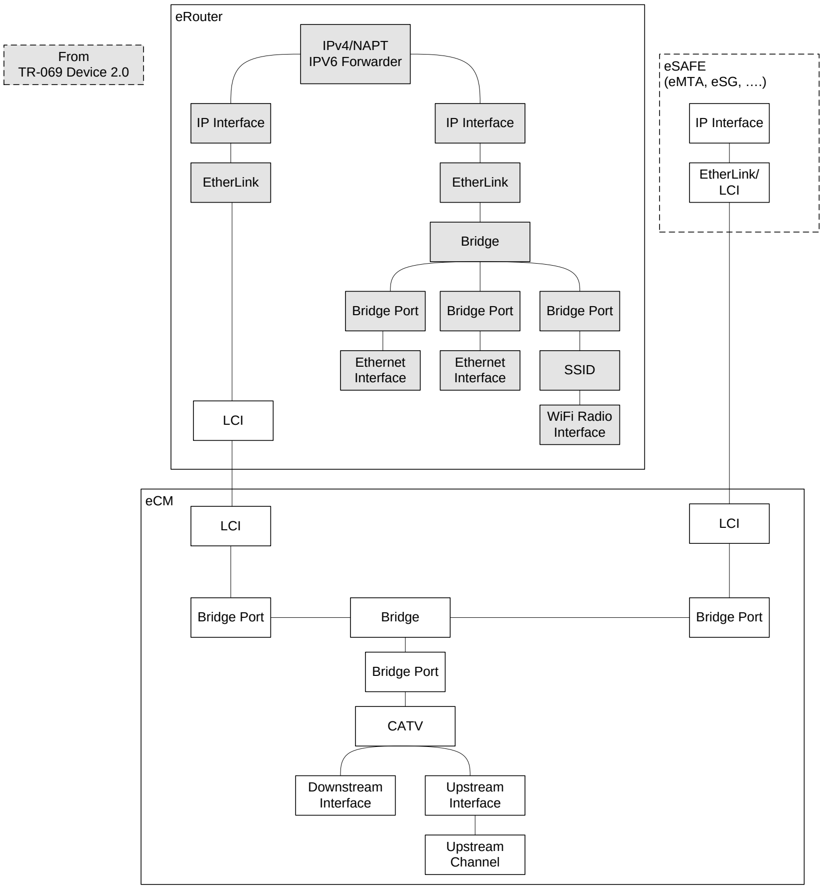


Figure 5-4 - TR-069 Interface Model Applied to eDOCSIS eRouter

5.2 eRouter Device Management[9]

The eRouter device that supports TR-069 implies the support of dual stack management, SNMP for the eCM component, and TR-069 for the eRouter component, as shown in Figure 5-4.

eRouter eSAFEs can be modeled as a stack of interfaces, and, in the future, other eSAFEs might support TR-069 protocol. This specification does not address architecture requirements such those listed in section 5.1 of [eDOCSIS], specifically, whether two TR-069-capable eSAFEs share the same TR-069 management stack or have separate stacks (as in the SNMP model). This is outside of the scope of this specification and within the scope of [eDOCSIS].

5.3 Service Discovery[10]

Service discovery will allow devices with services to announce their presence and allow a query / response method for discovering and choosing a service from a list of possible candidates that provide that service. An example is a network-based printer, mDNS, as defined in [RFC 6762], and UPnP, as defined by the UPnP forum and specified in [ISO/IEC 29341] are used to implement Service Discovery in the eRouter.

[9] Added per eRouter N-11.1014-3 on 10/27/11 by JB.
[10] Section added per eRouter-N-14.1156-2 on 7/18/14 by PO.

5.3.1 mDNS (multicast Domain Name System)[11]

The mDNS protocol provides both an announcement and a query / response mechanism to provide a list of devices that offer services on the home network. The mDNS protocol is link-scoped only but can be enhanced to provide service to multiple networks in the home. Therefore, a method to relay announcements and queries/responses between different networks is needed. The requirements to accomplish this are listed in the sections for IPv4 and IPv6 Service Discovery implementation.

The eRouter will take an announcement, query or response packet from one link/subnet and relay it onto another link/subnet but replace the IP source address from the originating link/subnet with the eRouter's egress IP address on the other link/subnet. Additionally, if the payload of the mDNS packet contains a link-local or Auto-IP resource record, those address records will be removed before the packet is placed onto the new link/subnet.

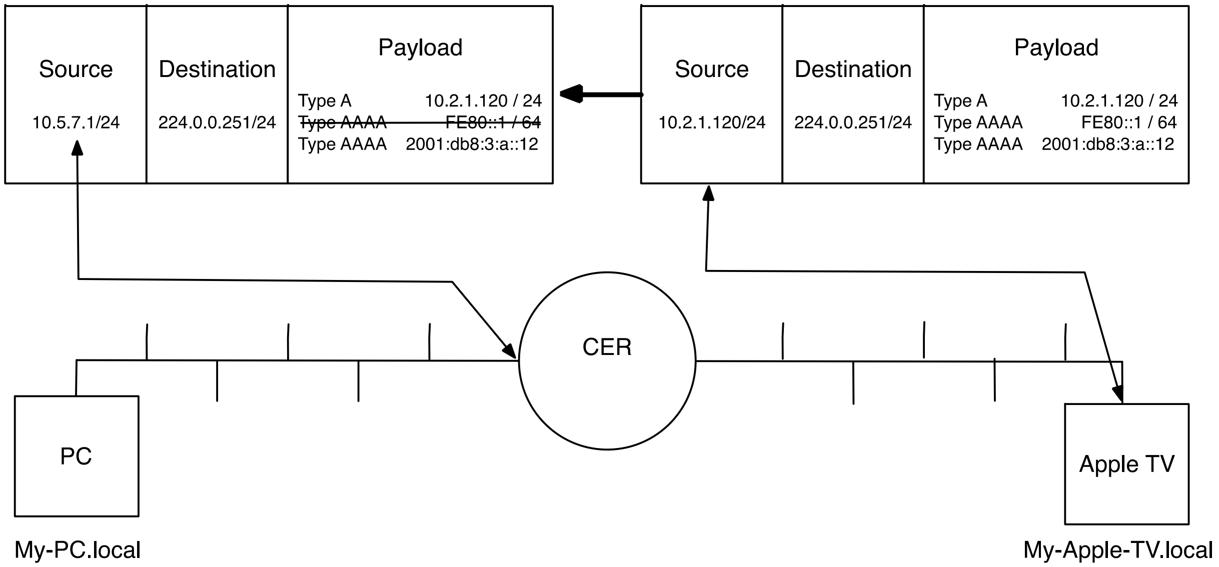


Figure 5-5 - mDNS Source IP and Payload Changes Going Through a Router

Hosts implementing the mDNS protocol might silently discard any frame with a source IP address that is not part of the receiver's network. When the eRouter receives a packet with an address that is not on the receiver's network, the eRouter MUST replace the source address in the packet with the IP address of the eRouter's egress interface to the receiver's network. This ensures that the packet is properly relayed to other hosts and not dropped by the host.

mDNS forwards as described above to multicast addresses (FF02::FB and 224.0.0.251).

The eRouter will not keep state information, but simply relays the packet and make the appropriate changes to the IP source address and payload.

5.3.2 UPnP (Universal Plug and Play)

The UPnP architecture makes use of a number of protocols including IP, TCP, UDP, HTML, and XML to enable peer-to-peer networking and service discovery. Most of the protocols used by UPnP will function across network segments in the home network. The one exception and the focus of this section is the UPnP Device Discovery Protocol, which uses the Simple Service Discovery Protocol (SSDP). UPnP was designed to function in an unmanaged network environment by having UPnP controllers (control points) automatically discover UPnP devices. SSDP is used by UPnP controllers to search for UPnP devices and is also used by UPnP devices to announce themselves and their services to UPnP controllers.

[11] Revised per eRouter-N-14.1233-1 on 2/13/15 by JB.

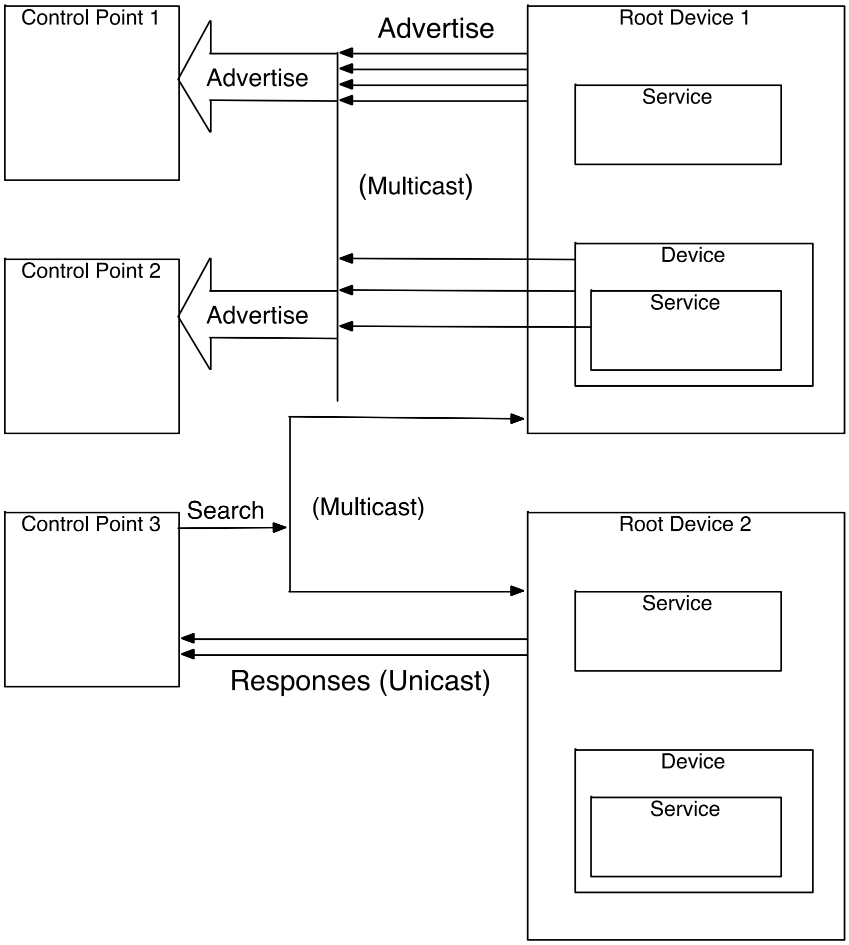


Figure 5-6 - UPnP General Architecture

UPnP forwards as described above to multicast addresses (FF02::C, FF05::C, FF02::130 and 239.255.255.250). eRouter is expected to forward to the site scoped address only.

The eRouter should not forward or listen for SSDP messages, for either IPv4 or IPv6, on the Operator Facing interface. IPv6 support was added to the [ISO/IEC 29341] in annex A.

5.4 CER-ID (Customer Edge Router – Identification)[12]

A home network may contain one or more edge routers and one or more internal routers providing connectivity to home devices. An eRouter, by default, demarcates the edge of the customer network and provides a method to assist internal routers in determining which device(s) reside at the edge of the customer network using a DHCPv6 CER-ID option encoding. The CER-ID is a 128-bit string that is usually set to an IPv6 interface address that is reachable on the customer LAN, though other values can be set to establish the role of the edge router

It is desirable to have services such as firewall functions and NAT/NAPT employed by the device(s) at the edge of the home network and not by other internal routers.

[12] Section added per eRouter-N-14.1152-3 on 7/18/14 by PO.

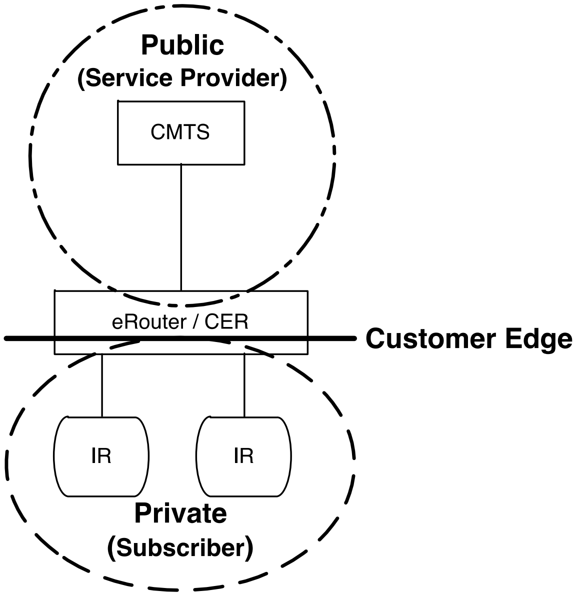


Figure 5-7 - Example of a CER Demarcation Boundary

6 eROUTER INITIALIZATION[13]

The eRouter operates in any one of three possible modes – 'IPv4 Protocol Enabled', 'IPv6 Protocol Enabled', or 'Dual IP Protocol Enabled', as summarized in Table 6-1. The eRouter can also be set to 'Disabled', which turns the eRouter into a bridging device. The eRouter MUST support all three modes of operation, and the ability to be set to 'Disabled'. The eRouter MUST default to 'Dual IP Protocol Enabled' mode in conformance with [RFC 6540].

The eRouter Mode is controlled via the eRouter Initialization Mode Encoding, the eRouter Initialization Mode Override Encoding and the esafeErouterInitModeControl object, all defined in Annex B. Prior to its initialization, the eRouter is enabled or disabled through the eRouter Initialization and eRouter Initialization Mode Override encodings in the cable modem configuration file. The esafeErouterInitModeControl object is used to change (override) the eRouter Mode after eRouter initialization completes. The eRouter ignores the esafeErouterInitModeControl object if it is present in a DOCSIS cable modem configuration file.

There are two means of overriding the eRouter Initialization Mode Encoding, the eRouter Initialization Mode Override Encoding and the esafeErouterInitModeControl object.

The esafeErouterInitModeControl object is used to change the eRouter Mode after the eRouter has initialized. Whenever the value of esafeErouterInitModeControl is changed from the default of honoreRouterInitMode(5) via an SNMP SET, the eRouter MUST override the eRouter Initialization Mode encoding encapsulated in the eCM configuration file and use the value of the esafeErouterInitModeControl.

For an eRouter which has been set to 'Disabled', the eRouter Initialization Mode Override Encoding is used to force the eRouter to remain 'Disabled' and ignore the value of the eRouter Initialization Mode TLV. If the eRouter is 'Disabled' and the esafeErouterInitModeControl object is set to honoreRouterInitMode(5), the eRouter MUST follow the eRouter Initialization Mode Override Encoding to determine whether it is to continue to remain 'Disabled" or whether it is to obey the eRouter Initialization Mode Encoding. If the eRouter is not 'Disabled' or the esafeErouterInitModeControl object is not set to honoreRouterInitMode(5), the eRouter MUST ignore the eRouter Initialization Override Encoding.

The eRouter MUST evaluate Initialization Mode configuration controls in the following order of precedence:

1. The stored esafeErouterInitModeControl object written via an SNMP management station SET prior to a reset
2. The eRouter Initialization Mode Override [TLV 202.3 in the cable modem configuration file]
3. eRouter Initialization Mode [TLV 202.1 in the cable modem configuration file]

The eRouter MUST persist its initialization mode across reinitializations. The eRouter MUST permit an SNMP SET to the esafeErouterInitModeControl object upon completing initialization via the TLV encodings.

When the eRouter is 'Disabled', the eRouter MUST NOT enable either IPv4 or IPv6 services or route IP between the Customer-Facing Interfaces and Operator-Facing Interfaces. When the eRouter is 'Disabled', it transparently bridges all traffic directly between its Customer-Facing Interfaces and its Operator-Facing Interface. When configured in this way, it will appear as if there is no eRouter present. The CM bridges all traffic (regardless of IP protocol version) to the CPE ports that would have been behind the eRouter had it been enabled. When configured as 'Disabled', the eRouter specification becomes irrelevant – the interfaces become part of the cable modem. All behavior will occur according to the DOCSIS specifications.

When the eRouter is in 'IPv4 Protocol Enabled' mode, the eRouter performs IPv4 provisioning as described in Section 7 and IPv4 data forwarding and NAPT according to Section 9. The eRouter operating in 'IPv4 Protocol Enabled' mode does not perform any IPv6 provisioning. When the eRouter is in 'IPv4 Protocol Enabled' mode, the eRouter MUST NOT forward IPv6 traffic between the Operator-Facing Interface and the Customer-Facing Interfaces.

When the eRouter is in 'IPv6 Protocol Enabled' mode, the eRouter performs IPv6 provisioning according to Section 8 and IPv6 data forwarding according to Section 10. The eRouter operating in 'IPv6 Protocol Enabled' mode

[13] Revised per eRouter-N-09.0877-2 on 5/11/10 by JB, revised per eRouter N-11.1015-2 on 11/04/11 by JB, and revised per eRouter-N-13.1091-4 on 3/7/13 by PO; revised per eRouter-N-13.1127-4 on 3/20/14 by PO.

does not perform any IPv4 provisioning. When the eRouter is in IPv6 Protocol Enabled' mode, the eRouter MUST NOT forward IPv4 traffic between the Operator-Facing Interface and the Customer-Facing Interfaces.

When the eRouter is in 'Dual IP Protocol Enabled' mode, the eRouter performs IPv4 provisioning as described in Section 7 and IPv6 provisioning according to Section 8. Once an eRouter in 'Dual IP Protocol Enabled' mode acquires an IPv4 address per Section 7, the eRouter performs IPv4 data forwarding and NAPT according to Section 9. Once an eRouter in 'Dual IP Protocol Enabled' mode acquires an IPv6 address and prefix per Section 8, the eRouter performs IPv6 data forwarding according to Section 10.

When the eRouter is enabled in any of the IP Protocol Enabled Modes, the eRouter MUST forward IP traffic between the Customer-Facing Interfaces, regardless of which IP Protocol Mode is enabled.

Table 6-1 provides a summary of the eRouter behavior based upon the configured mode of operation as well as when it is 'Disabled'.

Table 6-1 - eRouter Modes

Mode	IPv4	IPv6
Disabled	Disables the eRouter resulting in a bridge. No IPv4 provisioning. CM bridges all traffic per [MULPIv3.0] spec.	Disables the eRouter resulting in a bridge. No IPv6 provisioning. CM bridges all traffic per [MULPIv3.0] spec.
IPv4 Protocol Enabled	IPv4 Provisioning (Section 7). IPv4 data forwarding using NAPT (Section 9).	No IPv6 provisioning. No IPv6 data forwarding between Operator-Facing Interface and the Customer-Facing Interfaces.
IPv6 Protocol Enabled	No IPv4 provisioning. No IPv4 data forwarding between Operator-Facing Interface and the Customer-Facing Interfaces.	IPv6 Provisioning (Section 8). IPv6 data forwarding (Section 10).
Dual IP Protocol Enabled	IPv4 Provisioning (Section 7). IPv4 data forwarding using NAPT (Section 9).	IPv6 Provisioning (Section 8). IPv6 data forwarding (Section 10).

7 IPV4 PROVISIONING[14]

The normative requirements of this section are mandatory for an eRouter that implements the 'IPv4 Protocol Enabled' mode and/or the 'Dual IP Protocol Enabled' mode as defined in Section 6.

After the CM has reached operational state, if the eRouter is configured to route IPv4 packets, the eRouter MUST use DHCPv4 [RFC 2131] via its Operator-Facing Interface in order to obtain an IP address and any other parameters needed to establish IP connectivity, as illustrated in Figure 7-1.

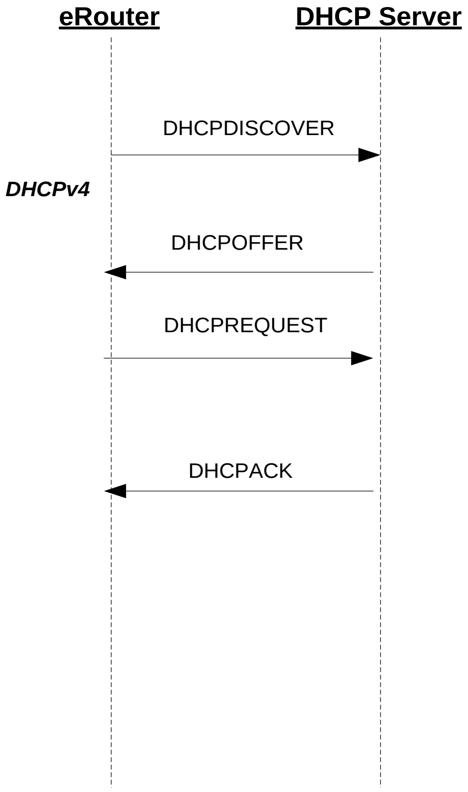


Figure 7-1 - IPv4 Provisioning Message Flow

The eRouter may receive multiple DHCPOFFER messages in response to its DHCPDISCOVER message. If a received DHCPOFFER message does not include all of the required DHCPv4 fields and options as described in Section 7.2, the eRouter MUST discard the DHCPOFFER message and wait for another DHCPOFFER message. If none of the received DHCPOFFER messages contain all the required DHCPv4 fields and options, the eRouter MUST retransmit the DHCPDISCOVER message.

The backoff values for retransmission of DHCPDISCOVER messages SHOULD be chosen according to a uniform distribution between the minimum and maximum values in the rows of Table 7-1.

Table 7-1 - eRouter DHCP Retransmission Interval

Backoff Number	Minimum (sec.)	Maximum (sec.)
1	3	5
2	7	9

[14] Section updated per eRouter-N-13.1127-4 on 3/21/14 by PO.

Backoff Number	Minimum (sec.)	Maximum (sec.)
3	15	17
4	31	33
5	63	65

The eRouter SHOULD also implement a different retransmission strategy for the RENEWING and REBINDING states, as recommended in [RFC 2131], which is based on one-half of the remaining lease time.

The eRouter MUST limit the number of retransmissions of the DHCPDISCOVER and DHCPREQUEST messages to five or fewer. The eRouter MUST NOT forward IPv4 traffic between its Customer-Facing Interface and its Operator-Facing Interface until it has completed IPv4 provisioning, including the successful receipt of a DHCPACK message. The eRouter MUST NOT forward IPv4 traffic if, at any time, it does not have an IPv4 address for its Operator-Facing Interface.

The eRouter MUST be able to accept a unicast response from the DHCP server/relay agent.

[RFC 3203] describes an extension to DHCPv4 that allows a server to send a FORCERENEW message that forces a client to renew its lease. The eRouter MUST ignore all received FORCERENEW messages.

7.1 DHCPv4 Fields Used by the eRouter[15]

The eRouter MUST include the following fields in the DHCPDISCOVER and DHCPREQUEST messages:

- The hardware type (htype) MUST be set to 1.
- The hardware length (hlen) MUST be set to 6.
- The client hardware address (chaddr) MUST be set to the 48-bit MAC address associated with the IPv4 CM-facing interface of the eRouter.
- The Broadcast bit MUST NOT be set.
- The client-identifier option MUST be included, using the format defined in [RFC 4361].
- The parameter request list option MUST be included.
- The following option codes (defined in [RFC 2132] and [RFC 4361]) MUST be included in the list
  - Option code 1 (Subnet Mask)
  - Option code 3 (Router Option)
  - Option code 6 (DNS Server Option)
  - Option code 60 (Vendor Class Identifier) [eRouter1.0]
  - Option Code 43 (see [eDOCSIS])
  - Option 55 Parameter Request List:

The following fields are expected in the DHCPOFFER and DHCPACK messages returned to the eRouter. The eRouter MUST configure itself with the listed fields from the DHCPACK:

- The IP address to be used by the eRouter (yiaddr) (critical).
- The IP Address lease time, option 51 (critical).
- The Server identifier, option 54 (critical).
- The subnet mask to be used by the eRouter (Subnet Mask, option 1) (critical).

[15] Updated by eDOCSIS-N-13.1127-4 on 3/21/14 by PO

- A list of addresses of one or more routers is to be used for forwarding eRouter-originated IP traffic (Router Option, option 3 (critical).

NOTE: The eRouter is not required to use more than one router IP address for forwarding.

- A list of DNS Server addresses (critical).
- A list of options under the CL_V4EROUTER_CONTAINER_OPTION option which are passed on to CPE devices as defined in the [CANN DHCP] (non-critical).

If a critical field is missing or invalid in the DHCPACK received during initialization, the eRouter MUST restart the DHCP cycle, beginning with an initial DHCPDISCOVER.

If a non-critical field is missing or invalid in the DHCPACK received during initialization, the eRouter MUST ignore the field, and continue the provisioning process.

If the yiaddr, Server Address, or Lease Time field is missing or invalid in the DHCPACK received during a renew or rebind operation, the eRouter MUST retry the renew or rebind operation until either: (1) it receives a response containing valid values of the yiaddr, Server Address, and Lease Time fields; or (2) the lease expires. If the lease expires, the eRouter MUST restart the DHCP cycle, beginning with an initial DHCPDISCOVER.

If any field other than the yiaddr, Server Address or Lease Time is missing, or is invalid in the DHCPACK received during a renew or rebind operation, the eRouter MUST ignore the field if it is invalid and remain operational.

7.2 eRouter Interface Addressing Using Link-ID[16]

When operating in 'Dual IP Protocol Enabled' mode and Link-ID is enabled, the eRouter MUST generate a unique /24 prefix for each Customer-Facing IP Interface using the 10.0.0.0/8 aggregate prefix and the Link ID generated from the appropriate IPv6 prefix assigned to the Customer-Facing IP Interface.

A unique IPv4 prefix is created using two steps:

1. use the decimal value 10 for the first octet,
2. convert IPv6 Link octets to their decimal equivalents for IPv4 octets 2 and 3.

Step #2 is explained in both the following text and diagram. For example, if an eRouter assigns IPv6 prefix 2001:db8:1234:5601::/64 to a Customer-Facing IP Interface, the Link ID for that interface will be hex 5601(bits 49-64 of the IPv6 prefix). The eRouter will use this Link ID to construct the second and third octets of its /24 IPv4 prefix. The second octet for this example is decimal 86 (equivalent of 0x56, the first octet of the Link ID), and the third octet is decimal 1 (equivalent of 0x01, the second octet of the Link ID). Thus, in this example, the eRouter will assign an IPv4 prefix of 10.86.0.0/24 to the Customer-Facing IP interface. These requirements enable native routing of IPv4 without the need for a routing protocol or NAPT when the eRouter is operating in Dual IP Protocol Enabled mode. Refer to the Figure 7-2 below and Annex E for further details.

[16] Added per eRouter-N-14.1155-3 on 2/10/15 by JB.

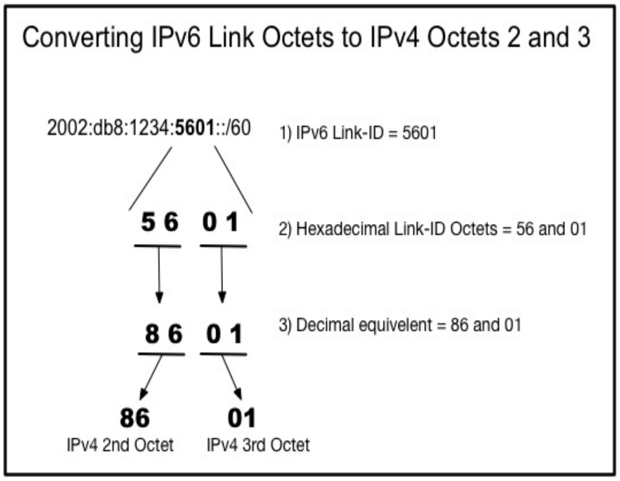

Converting IPv6 Link Octets to IPv4 Octets 2 and 3
2002:db8:1234:5601::/60
1) IPv6 Link-ID = 5601
5 6 0 1
2) Hexadecimal Link-ID Octets = 56 and 01
8 6 0 1
3) Decimal equivelent = 86 and 01
86
IPv4 2nd Octet
01
IPv4 3rd Octet

Figure 7-2 - Example Deriving IPv4 Octets 2 and 3 from an IPv6 Prefix

Using the above Link-ID example, the IPv6 address of 2001:db8:1234:5601::1/64 results in a corresponding IPv4 address of 10.86.1.1/24.

When operating in IPv4 Protocol Enabled mode only or when operating in ‘Dual IP Protocol Enabled’ mode with Link-ID disabled, the eRouter MUST generate each unique /24 IPv4 prefix from one of the three blocks of address space reserved for private internets per [RFC 1918].

7.3 Router DHCPv4 Server Sub-element

The DHCP server is responsible for assigning network address leases to LAN IP devices associated with Customer-Facing Interfaces. It is also responsible for providing LAN IP devices with configuration information via DHCP Option codes, as specified in [RFC 2132].

7.3.1 DHCPv4 Server Function Goals

Goals for the DHCP server include the following:

- Assign network address leases to CPE devices according to [RFC 2131].
- Assign private CPE addresses according to [RFC 1918].
- Assign configuration information according to [RFC 2132].

7.3.2 DHCPv4 Server Function System Description

The eRouter DHCPv4 server responsibilities include:

- Assigning IP Addresses and delivering DHCP configuration parameters to CPE Devices. The server relies on built-in default values for initial IP Address pool configuration, lease parameter configuration, and DHCP options values.
- Optional logging of DHCPv4 server errors to a local event log.

7.3.3 DHCPv4 Server Function Requirements[17]

```
eRouter Operator Facing Interface Provisioned Prefix: 2001:db8:1234:5600::/56

Customer-Facing IP Interface br(0)
     IPv6 Prefix:         2001:db8:1234:5600::/64
     Link ID Conversion:  56 00 -> 86 00
     IPv4 Network:   10.86.0.0/24   Gateway:  10.86.0.1

Customer-Facing IP Interface br(1)
     IPv6 Prefix:         2001:db8:1234:5600::/64
     Link ID Conversion:  56 00 -> 86 00
     IPv4 Network:   10.86.1.0/24 Gateway:  10.86.1.1

Customer-Facing IP Interface br(2)
     IPv6 Prefix:         2001:db8:1234:5601::/64
     Link ID Conversion:  56 01 -> 86 01
     IPv4 Network:   10.86.1.0/24 Gateway:  10.86.1.1

Customer-Facing IP Interface br(3)
     IPv6 Prefix:         2001:db8:1234:5601::/64
     Link ID Conversion:  56 01 -> 86 01
     IPv4 Network:   10.86.1.0/24 Gateway:  10.86.1.1


Topology Mode Encoding: Favor Width
Calculated on 3-Bit Boundary
Calculated subdelegation prefix length: /59
Number /59 available for prefix sub-delegation: 7
-- Sub-delegated prefix: 2001:db8:1234:5620::/59
-- Sub-delegated prefix: 2001:db8:1234:5640::/59
-- Sub-delegated prefix: 2001:db8:1234:5660::/59
-- Sub-delegated prefix: 2001:db8:1234:5680::/59
-- Sub-delegated prefix: 2001:db8:1234:56a0::/59
-- Sub-delegated prefix: 2001:db8:1234:56c0::/59
-- Sub-delegated prefix: 2001:db8:1234:56e0::/59

Sub-delegation:

The CER in this example has (4) Customer Facing IP Interfaces - represented below as
br(0) -> br(3).
Assume there are (4) IRs with one attached to each of those Customer Facing IP
Interfaces of the CER.
Further assume each IR is assigned an IPv6: IA_NA and IA_PD and IPv4 address in the
following way by the CER.


IR(0) on br(0) is  assigned via DHCPv6  IA_NA =  2001:db8:1234:9a00::100 and IA_PD =
2001:db8:1234:9a20::/59 and via DHCPv4 IP Address = 10.154.0.100
IR(1) on br(1) is  assigned via DHCPv6  IA_NA =  2001:db8:1234:9a01::100 and IA_PD =
2001:db8:1234:9a40::/59 and via DHCPv4 IP Address = 10.154.1.100
IR(2) on br(2) is  assigned via DHCPv6  IA_NA =  2001:db8:1234:9a02::100 and IA_PD =
2001:db8:1234:9a60::/59 and via DHCPv4 IP Address = 10.154.2.100
IR(3) on br(3) is  assigned via DHCPv6  IA_NA =  2001:db8:1234:9a03::100 and IA_PD =
2001:db8:1234:9a80::/59 and via DHCPv4 IP Address = 10.154.3.100

Route Table:

Network Destination                Next Hop                        Interface
--------------------------------------------------------------------------------------
----------
2001:db8:1234:9a20::/59    2001:db8:1234:9a00::100    br(0)
```

[17] Updated by eRouter-N-13.1127-4 on 3/21/14 by PO; per eRouter-N-14.1155-3 on 2/10/15 by JB.

```
10.154.32.0/19            10.154.0.100              br(0)
2001:db8:1234:9a40::/59   2001:db8:1234:9a01::100   br(1)
10.154.64.0/19            10.154.1.100              br(1)
2001:db8:1234:9a60::/59   2001:db8:1234:9a02::100   br(2)
10.154.96.0/19            10.154.2.100              br(2)
2001:db8:1234:9a80::/59   2001:db8:1234:9a03::100   br(3)
10.154.128.0/19           10.154.3.100              br(3)
```

The eRouter MUST include a DHCPv4 server compliant with [RFC 2131].

In addition, the following requirements apply to the DHCPv4 Server function:

- When the DHCP server assigns an active lease for an IP address to a CPE Device, the server MUST remove that IP address from the pool of IP addresses available for assignment.
- The requirements in this section use an appropriate address space as defined in [RFC 1918], and overrides the requirements in Section 9.3.1 when using IPv4-only mode (not Dual-Stack)
- The DHCP server function of the eRouter MUST support the DHCP options indicated as mandatory in Table 7-2.
- The DHCP server function of the eRouter MUST NOT respond to DHCP messages that are received through the Operator-Facing Interface, nor originate DHCP messages from the Operator-Facing Interface.
- The DHCP server function of the eRouter MUST NOT deliver any DHCP option with null value to any CPE device.
- The DHCP server function SHOULD be operational independent of the eRouter Operator-Facing Interface connectivity state.
- If the eRouter Operator-Facing Interface is not successfully provisioned, the eRouter DHCP server function SHOULD assign a short lease time to CPE devices and may omit options it has not acquired.
- The DHCP server function MUST assign private IP address space as defined in [RFC 1918].
- The DHCP server function SHOULD log errors to a local event log.

Table 7-2 - DHCPv4 Server Options

Option Number	Option Function
0	Pad
255	End
1	Subnet Mask
3	Router Option
6	Domain Name Server
50	Requested IP Address
51	IP Address Lease Time
54	Server Identifier
55	Parameter Request List
4491.3	Option(s) acquired under CL_V4EROUTER_CONTAINER_OPTION from the Operator

7.4 Operator-Facing IPv4 Address Release Behavior[18]

There are a number of situations in which it is desirable for eRouter to release its associated IPv4 address leases in order to protect the integrity of the DHCP database. Examples of such circumstances include situations in which the eRouter needs to be administratively reset (i.e., for configuration change, software update, or other reasons), or a change to the IPv4 address during DHCPv4 renewal. Due to the eRouter's dependency on the eCM for maintaining operator-facing connectivity, the eRouter MUST release its lease information prior to an SNMP or administratively imposed re-initialization of the embedded CM in order to prevent loss of the communications path with the DHCP server.

Whenever the eRouter is instructed to reset, the eRouter MUST send a DHCP_RELEASE message [RFC 2131] for the IPv4 public address assigned by the DHCPv4 server to the eRouter's Operator-Facing Interface. The eRouter MUST send the DHCP_RELEASE message [RFC 2131] for the IPv4 public address assigned by DHCPv4 to the eRouter's Operator-Facing Interface whenever the eRouter receives a DHCPv4 server renewal response contains a different IPv4 address. The eRouter MUST NOT wait for a confirmation of the receipt of the release by the DHCPv4 server in order to re-initialize.

7.5 Customer-Facing IPv4 Address Release Behavior[19]

After initiating an administrative device reset in which the public address has been released, the eRouter customer-facing interfaces will be limited to inter-LAN forwarding until the device completes any necessary resets and a new address lease is acquired. Prior to the operator-facing interface acquiring an IPv4 address from the operator's DHCPv4 server, local network services and data forwarding of the customer-facing LAN interfaces will continue so long as the DHCPv4 server of the eRouter is enabled.

[18] Added per eRouter N-11.1015-2 on 11/04/11 by JB.

[19] Added per eRouter N-11.1015-2 on 11/04/11 by JB.

8 OPERATOR-FACING IPV6 PROVISIONING[20]

IPv4 address space is nearly exhausted. The IANA pool of free IPv4 address space is completely depleted and customers have yet to be fully migrated to IPv6. The features necessary to facilitate transition to IPv6 are described in the following sections.

The normative requirements of this section are mandatory for an eRouter that implements the IPv6 Protocol.

After the CM has completed provisioning, if the eRouter is operating in either 'IPv6 Protocol Enabled' mode or 'Dual IP Protocol Enabled' mode as defined in Section 6, the eRouter MUST use DHCPv6 [RFC 3315] in order to obtain an IP address for its Operator-Facing IP Interface and any other parameters needed to establish IP connectivity, as illustrated in Figure 8-1. The eRouter MUST use DHCPv6 prefix delegation [RFC 3633] in order to obtain an IPv6 prefix for the eRouter's Customer-Facing IP Interfaces and any downstream internal routers (IRs), as well as any other parameters needed to establish IPv6 connectivity within the home or office network.

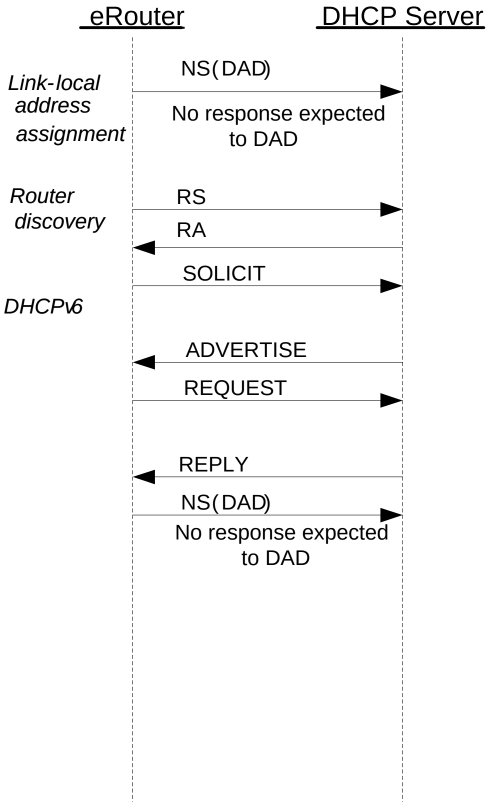


Figure 8-1 - IPv6 Provisioning Message Flow

The eRouter establishes IPv6 connectivity including assignment of:

- Link-local IPv6 address
- IPv6 address of a Default router
- Operator-Facing Interface IPv6 address (used for both management access to the eRouter and data forwarding)
- Other IPv6 configuration

[20] Revised per eRouter-N-09.0877-2 on 5/11/10 by JB, revised per eRouter-N-13.10905 on 3/5/13 by PO; updated by eRouter-N-13.1127-4 on 3/21/14 by PO. Revised per eRouter-N-14.1233-1 on 2/13/15 by JB.

These steps are described in the following subsections.

8.1 Obtain Link-Local Address[21]

The eRouter MUST construct a link-local address for its Operator-Facing Interface and each of its Customer-Facing Interface(s) according to the procedure in section 5.3 of [RFC 4862]. The eRouter MUST use the EUI-64 identifier as a link-local address for each of its interfaces as described in [RFC 4291]. For each of its interfaces, the eRouter MUST join the all-nodes multicast address and the solicited-node multicast address of the corresponding link-local address [RFC 4862], [RFC 2710]. The eRouter MUST use Duplicate Address Detection (DAD), as described in section 5.4 of [RFC 4862], to confirm that the constructed link-local addresses are not already in use prior to sending any Router Solicitations on the interface. If the eRouter determines that the constructed link-local address is already in use, the eRouter MUST terminate IPv6 operation on that interface.

8.2 Perform router discovery[22]

The eRouter MUST perform router discovery as specified in section 6.3 of [RFC 4861] on its Operator-Facing Interface. The source address used in the Router Solicitation MUST be the link-local address on the Operator-Facing Interface. The eRouter identifies neighboring routers and default routers from the received RAs.

8.3 Obtain IPv6 address and other configuration parameters[23]

An eRouter MUST examine the contents of RAs it receives on the Operator-facing interface and obeys the following rules:

- If the M bit in the RA is set to 1, the eRouter MUST use stateful DHCPv6 to obtain its IA_NA IPv6 address and other configuration information (and ignore the A and O bits).
- If the M bit is set to 1 in the RA, the eRouter MUST use stateful DHCPv6 to obtain its IA_PD.
- If the M bit is set to 0 and the O bit is set to 1, then the eRouter MUST NOT perform stateful or stateless DHCPv6 to obtain its IA_NA, IA_PD or other configuration information.
- If both the M bit and the O bit in the RA are set to 0, the eRouter MUST NOT attempt to use DHCPv6 to obtain its IPv6 address and other configuration information.
- The eRouter MUST NOT support SLAAC on its Operator-facing interface.

If the eRouter receives an RA where the M bit is set to zero then the eRouter considers provisioning to have failed.

If an RA contains a prefix advertisement for an IPv6 network prefix on which the eRouter does not have an address and the M bit in the RA is set to 1, the eRouter MUST use DHCPv6 to obtain its IPv6 address for its Operator-Facing Interface and renew any current IA_PD lease(s).

Table 8-1 - eRouter Behavior

The following table depicts eRouter behavior based on the values present in the M and O bits.

M Bit	O Bit	Stateful DHCPv6	Stateless DHCPv6	Prefix Delegation
1	1	Yes	No	Yes
1	0	Yes	No	Yes
0	1	No	No	No
0	0	No	No	No

[21] Revised per eRouter-N-09.0877-2 on 5/11/10 and per eRouter-N-11.1015-2 on 11/4/11 by JB.

[22] Revised per eRouter-N-09.0877-2 on 5/11/10 and per eRouter-N-11.1015-2 on 11/4/11 by JB.

[23] Revised per eRouter-N-09.0877-2 on 5/11/10, per eRouter-N-10.0974-2 on 1/24/11 per eRouter-N-11.1015-2 on 11/4/11 by JB, eRouter-N-12.1084-2 on 3/5/13 by PO, per eRouter-N-13.1090-5 on 3/5/13 by PO, per eRouter-N-13.1127-4 on 3/24/14 by PO, also changed by eRouter-N-13.1134-2 on 3/25/14 by PO.

The eRouter MUST follow the recommendations in section 4 of [RFC 5942], and in particular the handling of the L flag in the Router Advertisement Prefix Information option.

The eRouter MUST act as a requesting router for the purposes of DHCPv6 prefix delegation ([RFC 3633]). DHCPv6 address assignment (IA_NA) and DHCPv6 prefix delegation (IA_PD) SHOULD be done as a single DHCPv6 session.

The eRouter sends a DHCPv6 Solicit message as described in section 17.1.1 of [RFC 3315]. The Solicit message MUST include:

1. A Client Identifier option containing the DHCP Unique Identifier (DUID) for this eRouter (as specified by [RFC 3315]), the DUID should be formatted as follows:
   a. The eRouter MUST use a DUID that is one of DUID-LL, DUID-EN or DUID-LLT type and;
   b. The eRouter MUST use a DUID that is persistent across administrative reset or reboot following a loss of power per [RFC 7084] W-6.
2. An IA_NA option to obtain its IPv6 address.
3. An IA_PD option (as specified in [RFC 3633]) to obtain its delegated IPv6 prefix.
4. A Reconfigure Accept option to indicate the eRouter is willing to accept Reconfigure messages.
5. An Options Request option, which MUST include the following options:
   - DNS Recursive Name Server option [RFC 3646].
   - DNS Domain Search List option as per [RFC 3646].
   - OPTION_SOL_MAX_RT (82) as per [RFC 7083].
6. A Vendor Class option containing 32-bit number 4491 (the Cable Television Laboratories, Inc., enterprise number) and the string "eRouter1.0".
7. A DOCSIS Device Identifier Option, as defined in [CANN DHCP].
8. A Vendor-specific option, containing
   a. The 32-bit number 4491 (the Cable Television Laboratories, Inc., enterprise number).
   b. A CableLabs Vendor Specific Option Request Option CL_OPTION_ORO as defined in [CANN DHCP].
   c. A CL_EROUTER_CONTAINER_OPTION requested inside CL_OPTION_ORO.

The eRouter MUST use the delegated prefix assigned by the most recent DHCPv6 operation even if the new prefix differs from the prefix previously assigned. This new prefix will overwrite any stored prefix information preserved across resets by the eRouter.

If the eRouter does not have a previously assigned delegated prefix, the eRouter MUST indicate a non-zero prefix size as DHCPv6 "hint" information [RFC 3633]. The eRouter MUST ask for a prefix large enough to assign one /64 for each of its Customer-Facing Logical Interfaces rounded up to the nearest nibble. The eRouter MUST be able to accept a delegated prefix length different from what was provided in the hint. If the delegated prefix is too small to address all of its interfaces, the eRouter SHOULD assign a single /64 for all Customer-Facing Logical Interfaces and log an error message.

Any packet received from the Operator-facing interface by the eRouter with a destination address in the prefix(es) delegated to the eRouter but not in the set of prefix(es) assigned by the eRouter to the Customer-facing interface MUST be dropped. For example, if the delegated prefix is a /56 but only 12 /64 are in active use, the eRouter should discard all traffic destined to the 242 unused /64. This is necessary to prevent forwarding loops and is also helpful in preventing malicious (DoS, network scanning, etc.) traffic from entering the LAN or using eRouter resources.

The eRouter MUST use the following values for retransmission of the Solicit message (see section 14 of [RFC 3315] for details)

IRT (Initial Retransmission Time)	= SOL_TIMEOUT
MRT (Maximum Retransmission Time)	= SOL_MAX_TIMEOUT

- MRC (Maximum Retransmission Count) = 0
- MRD (Maximum Retransmission Duration) = 0

The eRouter MUST use the following value for the Max Solicit timeout value as per [RFC 7083] in preference to any value shown in [RFC 3315]:

- SOL_MAX_RT = 3600 secs

The DHCP server responds to Solicit messages and Request messages with Advertise and Reply messages. The Advertise and Reply messages may include other configuration parameters, as requested by the eRouter, or as configured by the administrator, to be sent to the eRouter. If any of the following options are absent from the Advertise message, the eRouter MUST discard the message and wait for another Advertise message. If any of the following options are absent from the Reply message, the eRouter MUST consider IPv6 provisioning to have failed, discard the Reply, and continue transmitting Solicit messages. In addition the eRouter MAY log an event.

1. The IA_NA option containing the eRouter's IPv6 address;
2. The IA_PD option containing the delegated IPv6 prefix for use by the eRouter;
3. Reconfigure Accept option;
4. The DNS Recursive Name Server Option.

The eRouter MAY log an event if IPv6 provisioning has failed.

The eRouter interface MUST join the All-Nodes multicast address and the Solicited-Node multicast address of the IPv6 address acquired through DHCPv6. The eRouter MUST perform Duplicate Address Detection (DAD) with the IPv6 address acquired through DHCPv6.

If the eRouter determines through DAD that the IPv6 address assigned through DHCPv6 is already in use by another device, the eRouter MUST:

- Send a DHCP Decline message to the DHCP server, indicating that it has detected that a duplicate IP address exists on the link.
- Discontinue using the duplicate IP address.
- Consider the IPv6 provisioning process to have failed,log the event in the local log, and re-initiate the DHCP process.

The eRouter MUST support the Reconfigure Key Authentication Protocol, as described in section 21.5 of [RFC 3315].

The eRouter MUST NOT forward any IPv6 traffic between its Customer-Facing Interface and its Operator-Facing Interface until it has successfully completed the IPv6 provisioning process. The eRouter MUST NOT forward any IPv6 traffic between its Customer-Facing Interface and its Operator-Facing Interface if, at any time, it does not have a Globally-assigned IPv6 address on its Operator-Facing Interface. The eRouter MUST NOT forward any IPv6 traffic between its Customer-Facing interface and its Operator-Facing interface if it has not completed the delegated prefix acquisition process.

If DHCPv6 provisioning fails on the Operator-Facing interface for any reason, then the eRouter MUST transmit a Router Advertisement on the Customer-Facing interface with the router lifetime equal to zero.

8.4 Use of T1 and T2 Timers[24]

The eRouter MUST initiate the lease renewal process when timer eRouter-T1 expires. The eRouter MUST initiate the lease rebinding process when timer eRouter-T2 expires. Timers eRouter-T1 and eRouter-T2 are called T1 and T2, respectively, in the DHCP specifications. If the DHCP server sends a value for eRouter-T1 to the eRouter in a DHCP message option, the eRouter MUST use that value. If the DHCP server does not send a value for eRouter-T1, the CM MUST set eRouter-T1 to 0.5 times the duration of the lease [RFC 3315]. If the DHCP server sends a

[24] Revised per eRouter-N-09.0877-2 on 5/11/10 by JB.

value for eRouter-T2 to the eRouter in DHCP message options, the eRouter MUST use that value. If the DHCP server does not send a value for eRouter-T2, the eRouter MUST set eRouter-T2 to 0.875 times the duration of the lease [RFC 3315].

8.5 Customer-Facing IPv6 Provisioning of CPE Devices[25]

An eRouter that has no default routers on its Operator-facing interface MUST NOT advertise itself as an IPv6 default router on its Customer-facing interfaces. If an eRouter is serving as an advertising router and subsequently detects loss of connectivity on its Operator-facing interface it MUST invalidate itself as an IPv6 default router on each of its customer-facing interfaces. The eRouter MUST then transmit one or more Router Advertisement messages with the Router Lifetime field set to zero.

The eRouter MUST divide the MSO delegated prefix acquired from the IA_PD option per Section 8.3 during the provisioning process into several sub-prefixes to be used for its Customer-Facing IP Interfaces and any downstream internal routers (IRs).

By default, the eRouter MUST divide the delegated prefix based on the MSO provisioned prefix size and the configurable Topology mode (Section B.4.9) as follows:

- If the provisioned MSO assigned IA_PD is smaller than a /56 (e.g., a /60) and the Topology mode is set to "favor depth", the eRouter MUST divide the delegated prefix on two (2)-bit boundaries into four (4) sub-prefixes by default.
- If the provisioned MSO assigned IA_PD is smaller than a /56 (e.g., a /60) and the Topology mode is set to "favor width", the eRouter MUST divide the delegated prefix on three (3)-bit boundaries into eight (8) sub-prefixes by default.
- If the provisioned MSO assigned IA_PD is a /56 or larger and the Topology mode is set to "favor depth", the eRouter MUST divide the delegated prefix on three (3)-bit boundaries into eight (8) sub-prefixes by default.
- If the provisioned MSO assigned IA_PD is a /56 or larger and the Topology mode is set to "favor width", the eRouter MUST divide the delegated prefix on four (4)-bit boundaries into sixteen (16) sub-prefixes by default.
- If the provisioned MSO assigned IA_PD is too small to divide in the manner described, the eRouter MUST divide the delegated prefix into as many /64 sub-prefixes as possible and log an error message indicating the fault.

For example, if eRouter set to "favor width" receives a /56 IA_PD from the MSO during the provisioning process, the eRouter will split the /56 delegated prefix into sixteen /60 sub-prefixes for use within the home or office. In another scenario where an eRouter set to "favor depth" receives a /62 IA_PD from the MSO during the provisioning process, it would split that /62 delegated prefix into four /64 prefixes for use within the home or office network.

The eRouter MAY support other methods of dividing the provisioned MSO assigned IA_PD; any such methods would have to be configured by the MSO or its customer.

The eRouter MUST generate and assign a globally unique /64 prefix for each Customer-Facing IP Interface before sub-delegating any prefixes to downstream routers within the home.

The eRouter MUST allocate these /64 interface prefixes starting from the numerically lowest sub-prefix generated from the division of the MSO assigned IA_PD (as described above). If the sub-prefix is too small to address all of the Customer-Facing IP Interfaces, the eRouter MUST allocate additional /64 interface prefixes from the next, numerically consecutive sub-prefix.

The eRouter MAY reserve additional /64 interface-prefixes for Customer-Facing Logical Interfaces that could be enabled in the future.

After all of the eRouter's Customer-Facing IP Interfaces have been assigned a globally unique /64 prefix, the eRouter MUST delegate sub-prefixes to directly attached downstream routers starting from the numerically highest sub-prefix and working down in reverse numerical order. The prefix assignment in reverse order allows for the

[25] Revised per eRouter-N-11.1015-2 on 11/4/11 by JB, and eRouter-N-13.1090-5 on 3/5/13 by PO. Revised per eRouter-N-13.1101-3 and eRouter-N-13.1102-3 on 5/24/13 by JB, changed by eRouter-N-13.1134-2 on 3/25/14 by PO.

flexibility of having a contiguous Customer-Facing IP Interface prefix assignment for interfaces that may be enabled after the initial prefix assignment. This includes the most common use case of additional SSID interfaces that may be administratively disabled at the time the eRouter initializes that are later enabled.

If there are not enough sub-prefixes remaining to delegate to all downstream routers, the eRouter MUST log an error message indicating the fault.

For example, if there is an eRouter set to "favor depth" configured with two (2) Customer-Facing IP Interfaces that receives a MSO provisioned prefix of 3900:1234:5678:9ab0::/60, the prefix assignment would be as follows:

- Customer-Facing Logical Interface #1 would be assigned with the prefix: 3900:1234:5678:9ab0::/64
- Customer-Facing Logical Interface #2 would be assigned with the prefix: 3900:1234:5678:9ab1::/64

The eRouter would delegate sub-prefixes to the directly attached downstream routers starting first with the 3900:1234:5678:9abc::/62 sub-prefix, and next with 3900:1234:5678:9ab8::/62 sub-prefix, and so on.

If the MSO prefix is too small to address all of its interfaces, the eRouter MUST collapse the Customer-Facing IP Interfaces into a single Interface and assign a single /64,logging an error message indicating the fault. For example, if eRouter with eight (8) Customer-Facing (physical) Interfaces receives a single /64 prefix from the MSO during the provisioning process, the eRouter will be forced to bind all eight (8) interfaces into the lowest numbered, or primary LAN, creating a single flat network and a single Customer-Facing IP Interface, regardless of the existing LAN or VLAN configuration(s).

The eRouter MUST assign a global IPv6 address to each Customer-Facing IP Interface. The eRouter SHOULD generate each Customer-Facing IP Interface Identifier using the Modified EUI-64 process as described per [RFC 4291]. The Modified EUI-64 IPv6 Interface Identifier is created by converting the IEEE 802 MAC address assigned to each Customer-Facing IP Interface to an EUI-64 formatted 64-bit address, and complementing the U/L bit; then, pre-pending 64 bits of the prefix acquired under IA_PD in Section 8.3 to create the 128-bit IPv6 Interface Identifier address.

This entire process can be illustrated in the following way:

1. The aggregate MSO prefix is acquired per Section 8.3.
2. The eRouter then breaks this aggregate MSO prefix into sub-prefixes, based on the Topology Mode, Section B.4.9.
    a. If the MSO prefix is not large enough, it is broken into as many /64 sub-prefixes as possible and logs an error message.
3. The first of these sub-prefixes is further broken into /64 interface-prefixes for use one on each of the eRouter's Customer-Facing Logical Interfaces.
    a. If the sub-prefix is too small to number all Customer-Facing Logical Interfaces, the eRouter uses additional sub-prefixes as needed (in numerical order).
    b. If the aggregate MSO prefix is too small to number all Customer-Facing Logical Interfaces, the eRouter collapses them into a single interface, assigns a single /64 to that interface, and logs an error message.
4. Each Customer-Facing IP Interface is assigned an IP address from the corresponding interface-prefix.
5. The remaining sub-prefixes are delegated via DHCPv6 to directly downstream routers as needed, in reverse numerical order.

The eRouter MUST support SLAAC [RFC 4862] on all Customer-Facing Interfaces. This requirement satisfies IP address allocation on the Customer-Facing Interfaces for any host that does not implement a full DHCP client.

The eRouter MUST support a DHCPv6 server [RFC 3315] on all Customer-Facing Interfaces. This requirement provides the Customer-Facing Interface with the ability to allocate IP addresses to hosts that implement a DHCP client.

The eRouter MUST support Delegating Router behavior for the IA-PD Option [RFC 3633] on all Customer-Facing Interfaces. This requirement provides the means to delegate sub-prefixes to routers within the customer's network from the aggregate, delegated prefix assigned by the operator to the eRouter.

The eRouter MUST support Neighbor Discovery for IPv6 as defined in [RFC 4861].

The eRouter MUST advertise itself as a router for its delegated prefix(es) using the Route Information Option, as specified in section 2.3 of [RFC 4191].

The eRouter's Router Advertisement (RA) transmission period MUST be configurable from 3 seconds to 1800 seconds for each Customer Facing Logical Interface. This configuration flexibility is necessary to adapt to conditions for which the [RFC 4862] defined default of a 120 second interval is inadequate. For example, when prefixes are changed and timely notification of such change is essential to maintaining network continuity.

The eRouter MUST implement a 30 second Router Advertisement (RA) transmission interval by default for each of its Customer Facing Logical Interfaces. If the prefix information contained in an RA changes, the eRouter MUST immediately generate and transmit an updated RA.

8.5.1 Additional Customer-Facing IP Interfaces Enabled After Initial Provisioning [26]

If an eRouter Customer-Facing IP Interface is enabled after initial provisioning and the initial prefix delegation, the eRouter MUST continue prefix assignment for this interface from the next available lowest numbered /64 prefix available. To illustrate using the same example as above, if an additional Customer-Facing IP Interface is enabled after the initial prefix assignment, the eRouter would assign this interface with the prefix of 3900:1234:5678:9ab2::/64.

When the eRouter has used all of its sub-prefixes for any reason, the eRouter MUST NOT enable any new Customer-Facing IP Interfaces. When an attempt to enable a Customer-Facing IP Interface fails because there are no available prefixes, the eRouter MUST log an error message indicating the fault.

8.5.2 SLAAC Requirements for eRouter[27]

SLAAC is required for hosts that do not implement a DHCPv6 client.

The /64 prefix length is required for the dynamic numbering of CPE devices using SLAAC [RFC 4862]. The eRouter MUST generate Router Advertisements (RA) on each Customer-Facing Interface as per [RFC 4862].

The eRouter MUST include the following in its RA by default:

- A Prefix Information Option with a prefix derived from the prefix acquired under IA_PD in Section 8.3 and both the ICMPv6 options 'flags' L-Bit (On-link) bit and A-Bit (Autonomous) bit set to 1,
- Preferred and Valid lifetimes in the Prefix Information Option set equal to or less than the Preferred and Valid lifetimes communicated in the IA-PD option received on the Operator-Facing Interface. This requirement ensures prefix lifetime synchronization between the eRouter aggregate prefix and the prefix/es assigned to each Customer-Facing Interface.

The above L, and A settings in the RA will cause CPE devices to use auto-configuration by default for assigning their global IPv6 address.

Once the eRouter has completed Operator-facing DHCPv6 provisioning:

- The eRouter MUST include DNS configuration option RDNSS in its RA messages as specified in [RFC 6106].
- The eRouter MUST include DNS configuration option DNSSL in its RA messages as specified in [RFC 6106] if OPTION_DOMAIN_LIST (24) option is acquired via Operator-facing DHCPv6 provisioning.
- The eRouter MUST include the list of DNS servers specified in the OPTION_DNS_SERVERS (23).
- The eRouter MUST include the list of domain names specified in the OPTION_DOMAIN_LIST (24) option, if acquired via Operator-facing DHCPv6 provisioning.

[26] Section added per eRouter-N-13.1102-3 on 5/24/13 by JB, per eRouter-N-13.1127-4 on 3/24/14 by PO.

[27] Revised per eRouter-N-13.1090-5 on 3/7/13 by PO, per eRouter-N-14.1134-2 on 3/25/14 by PO.

8.5.2.1 Local Configuration of SLAAC Options[28]

The eRouter MAY provide a mechanism for local configuration of SLAAC for CPE devices. If local configuration is used, the eRouter MUST override the pass through of options received from the Cable Operator and provide the locally configured options to CPEs.

8.5.3 DHCPv6 Requirements for eRouter[29]

The eRouter MUST provide a DHCPv6 server on Customer-Facing Interfaces as described in:

- [RFC 3315] Dynamic Host Configuration Protocol for IPv6 (DHCPv6).
- [RFC 3736] Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6.

The eRouter DHCPv6 server MUST support providing the following DHCPv6 Options:

- [RFC 3646] DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6).
- [RFC 3633] IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6.
- [RFC 4075] Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6.
- [RFC 4242] Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6).

The DHCPv6 server MUST be able to manage at least one IA_NA for each client, and at least one address in each IA_NA.

The DHCPv6 server MUST be able to manage at least one IA_PD for each client and at least one delegated prefix in each IA_PD. The sub-prefix delegated to the client is derived from the aggregate prefix delegated to the eRouter from the Cable Operator as described in Section 8.5.

The eRouter DHCPv6 server MUST derive the valid and preferred lifetimes for Customer-facing IA_NA and IA_PD leases from the valid and preferred lifetimes acquired in the IA_PD on the Operator Facing Interface. The DHCPv6 messages sent on Customer-facing interface MUST contain the lifetime value greater than zero and equal to or less than the IA_PD lifetime acquired on the Operator Facing Interface.

IA_NA and IA_PD T1 and T2 values are supplied to the Customer-facing interface(s) in accordance with [RFC 3315] section 22.4 and [RFC 3633] section 9, respectively.

The eRouter MUST generate Router Advertisements (RA) on each Customer-Facing IP Interface as per [RFC 4862]. The RA MUST include the following by default:

- have the M bit set to 1,
- have the O bit set to 1,
- contain a Prefix Information Option with a prefix derived from the prefix acquired under IA_PD in Section 8.3 and both the ICMPv6 options 'flags' L-Bit (On-link) bit and A-Bit (Autonomous) bit set to 1,
- set the Preferred and Valid lifetimes in the Prefix Information Option equal to the Preferred and Valid lifetimes communicated in the IA-PD option on the Operator-Facing Interface. This requirement ensures prefix lifetime synchronization between the eRouter aggregate prefix and the prefix(es) assigned to each Customer-Facing Interface.

These settings in the RA will direct CPE devices to use DHCPv6 configuration for assigning their global IPv6 address. In most scenarios, an eRouter would make DHCPv6 services available concurrently with SLAAC in order to supply address and other information to hosts of varying capability. Hosts will be presented with a Router Advertisement that includes the M-bit set to indicate DHCPv6 operation in addition to the A-bit set to indicate SLAAC operation and the O-bit set to support stateless DHCPv6 clients.

[28] Added by eRouter-N-14.1134-2 on 3/25/14 by PO.

[29] Revised per eRouter-N- 13.1090-5 on 3/7/13 by PO, per eRouter-N-13.1127-4 on 3/24/14 by PO.

NOTE: Recent testing shows operating systems will perform both DHCPv6 and SLAAC for address acquisition when the operating system includes a DHCP client and both methods of address acquisition are made available.

The eRouter MUST be able to pass the following set of options received from the Cable Operator to the DHCPv6 server for configuration of CPEs.

- DNS Recursive Name Server option as specified in [RFC 3646]
- DNS Domain Search List option as specified in [RFC 3646]
- The list of options under the CL_EROUTER_CONTAINER_OPTION option, as defined in [CANN DHCP], which are passed to the eRouter by the operator.

The eRouter MAY relax the requirements on non-volatile storage of assigned addresses and delegated prefixes and MAY glean information about assigned addresses and delegated prefixes from Advertise, Renew, and Rebind messages received from clients.

8.5.3.1 Local Configuration of DHCPv6 Options[30]

The eRouter MAY provide a mechanism for local configuration of DHCPv6 options for CPE devices. If local configuration is used, the eRouter MUST override the pass through of options received from the Cable Operator and provide the locally configured options to CPEs.

8.5.4 Prefix Changes[31]

An eRouter might receive a replacement prefix from the DHCP server (e.g., during a renewal operation on the Operator-Facing Interface). Due to the global nature of IPv6 addressing of CPEs, the eRouter is required to deprecate the previously acquired prefix and allocate addressing from the newly acquired prefix whenever this happens.

The eRouter MUST perform CPE provisioning as per Section 8.5 immediately upon receiving a new prefix.

The eRouter deprecates the previously acquired prefix via routines defined in Soft Reset (Annex B.5). These steps include immediately sending an RA message that indicates the prefix to be deprecated, sending a DHCP-RECONFIGURE message prompting DHCP clients to renew their IP information, shutting down and restarting all Customer-Facing Interfaces, as well as clearing of the ND cache and any other procedures that are specific to the implementation. When CPEs receive the updated RA, RECONFIGURE message, or notice a state change in the link-state of the Customer-Facing Interface, they are compelled to discard their current IPv6 addresses and restart the address acquisition process. A majority of CPEs do not yet properly respond to the DHCP-RECONFIGURE message with a 'Rebind' per [RFC 6644] at the present time. We are anticipating this will change as compliance improves.

When a prefix previously assigned to the eRouter is no longer available for any reason (e.g., a prefix change during renewal), the eRouter MUST deprecate that prefix. When the eRouter deprecates a prefix, it MUST follow Soft Reset steps 2, 3, and 6 from Annex B.5. The eRouter MAY implement vendor-specific techniques that supplement those defined in Annex B.5.

When an eRouter receives updated information for a currently assigned prefix, the eRouter MUST immediately send Router Advertisements (RAs) with the updated prefix information and IPv6 DHCP RECONFIGURE (type 6, Rebind) on all Customer-Facing Interfaces.

8.6 Operator-Facing IPv6 Address Release Behavior[32]

There are a number of situations in which it is desirable for the eRouter to release its associated IPv6 address leases in order to ensure the integrity of the DHCP database. Examples of such circumstances include situations in which

[30] Added per eRouter-N-14.1134-2 on 3/25/14 by PO.
[31] Revised per eRouter N-13.1121-1 on 11/5/13 by PO.
[32] Revised per eRouter-N-11.1015-2 on 11/4/11 by JB.

the eRouter needs to be administratively reset (say for configuration change, software update or other reason) or a change to the IPv6 address during DHCPv6 renewal.

Due to the eRouter's dependency on the eCM for maintaining operator-facing connectivity, the eRouter MUST release its lease information prior to an SNMP or administratively imposed re-initialization of the embedded CM in order to prevent loss of the communications path with the DHCP server. The eRouter MUST NOT wait for confirmation of receipt of the release by the DHCPv6 server in order to re-initialize.

The eRouter MUST send a DHCP_RELEASE message [RFC 3315] for the IPv6 IA_NA and IA_PD assigned by the DHCPv6 server to the eRouter's Operator-Facing Interface for the following events:

- whenever the eRouter is instructed to reset,
- whenever the eRouter receives a DHCPv6 Reply message containing a different IPv6 prefix or IPv6 address.
- whenever the IA_PD is not renewed for any reason.

8.7 Customer-Facing IPv6 Address Release Behavior[33]

After initiating an administrative device reset in which the IA_NA and IA_PD addresses have been released, the eRouter customer-facing interfaces will be limited to inter-LAN forwarding until the device completes any necessary resets and new address and prefix leases are acquired. Prior to the eRouter's operator-facing interface acquiring IPv6 prefix information from the operator's DHCPv6 server, the eRouter will retain the IPv6 prefix information beyond the remaining valid lifetime of the prefix for the purposes of customer-facing LAN addressing. This additional measure on the customer-facing interfaces will insure that local network services and data forwarding of the customer-facing LAN interfaces will continue.

The eRouter MUST send an ICMPv6 'destination unreachable' message (code 5) for packets forwarded to it that use an address from a prefix that has been invalidated.

After initiating a Reset in which the Operator-Facing Interface's IA_NA and IA_PD addresses have been released, the eRouter MUST declare that it is no longer a Default Router by setting the Router Lifetime field to zero in the Router Advertisement.

8.8 CER-ID Requirements[34]

- The eRouter MUST assign the CER-ID value for each of its Customer-Facing IP Interfaces for which an IPv6 prefix has been assigned, by using the corresponding GUA assigned to the Customer-Facing IP Interface.
- The eRouter MUST include the DHCPv6 CL_CER_ID option [CANN DHCP] in Advertise or Reply messages containing an IA_PD.
- If the IPv6 address of the eRouter’s Customer Facing IP Interface that established the CER-ID changes for any reason, the eRouter MUST assign a new value for CER-ID to be included in subsequent DHCPv6 messages.
- The value of the CER-ID MAY be configurable by the subscriber. The exact mechanism is out of scope for this document.

[33] Revised per eRouter-N-11.1015-2 on 11/4/11 by JB, per eRouter-N-13.1108-3 on 7/16/13 by PO, per eRouter-N-13.1127-4 on 3/24/14 by PO.Revised per eRouter-N-14.1178-1

[34] Section added per eRouter-N-14.1152-3 on 7/18/14 by PO.

9 IPV4 DATA FORWARDING AND NAPT OPERATION

9.1 Introduction

The normative requirements of this section are mandatory for an eRouter that implements the 'IPv4 Protocol Enabled' mode and/or the 'Dual IP Protocol Enabled' mode as defined in Section 6.

9.1.1 Assumptions

- There is only a single Operator-Facing IP Interface on the eRouter.
- There is typically a single Customer-Facing IP interface on the eRouter.
- At least one globally-routable IPv4 address is available to the eRouter's Operator-Facing IP Interface.
- The Operator-Facing IP Interface is Ethernet encapsulated.
- The Customer-Facing IP interface is Ethernet encapsulated.

9.1.2 Overview

IPv4 Forwarding in the eRouter consists of three logical sub-elements:

- IPv4 Router
- NAPT (Network Address Port Translation)
- ARP (Address Resolution Protocol)

The IPv4 Router sub-element is responsible for forwarding packets between the Operator-Facing IP Interface and the Customer-Facing IP interfaces. This includes looking up the IPv4 Destination address to make a forwarding decision on whether to forward the packet from one of its interfaces to another one of its interface or to its internal stack.

Packet handling in the eRouter for NAPT includes:

- Providing a form of IPv4 address translation that allows for multiple IPv4 hosts on the Customer-Facing IP interfaces while presenting a small number of IPv4 addresses on the Operator-Facing IP Interface.
- Preventing unnecessary traffic on the Customer-Facing IP interfaces.
- Preventing traffic from one CPE device to another CPE device from traversing to the Operator-Facing Interface.

The ARP protocol on the eRouter provides a mechanism for converting IPv4 network addresses to Ethernet MAC addresses on both Customer-Facing IP interfaces and the Operator-Facing IP Interface.

9.2 System Description

9.2.1 Overview

Some eRouters may have multiple customer ports that are connected to the same logical IP router interface. One scenario would be when the eRouter has an 802.11 wireless port and an 802.3 Ethernet port on the single Customer-Facing logical IP interface. The text in this section uses the term "Customer-Facing IP interface" to refer to a single Customer-Facing logical IP router interface connected to the eRouter that is not necessarily mapped one-to-one with the number of Customer-Facing ports on the eRouter. This text documents the behavior of a single Customer-Facing IP interface, though it is possible that an eRouter could have multiple Customer-Facing IP interfaces. It is vendor-specific how to route between Customer-Facing Interfaces and the Operator-Facing IP Interface when there are multiple Customer-Facing IP interfaces.

Packets need to be processed by each of the three sub-elements in a very specific order (see Figure 9-1). The order is different depending on whether packets are received from a Customer-Facing IP interface or the Operator-Facing IP Interface.

When receiving packets from the Customer-Facing IP interface, the eRouter first attempts to route the packet through the router sub-element. If the router sub-element forwards the packet to the Operator-Facing Interface, the packet is passed to the NAPT sub-element to see if the packet requires NAPT translation. Once the NAPT sub-element has completed its work, the packet is sent to the ARP sub-element to resolve the IPv4 network address to Ethernet MAC. Then the packet is encapsulated in an Ethernet header and sent out the operator interface. If the router sub-element forwards the packet back out the Customer-Facing IP interface (perhaps because the client is on a different private subnet), the packet is sent to the ARP sub-element to resolve the IPv4 network address to Ethernet MAC. Then the packet is encapsulated in an Ethernet header and sent out the appropriate interface. No NAPT processing is necessary for packets routed back out the Customer-Facing IP interface.

When packets are received from the Operator-Facing Interface, they are immediately sent to the NAPT sub-element to translate the IPv4 network addresses back to addresses within the domain of the router sub-element. Once the NAPT has been performed on the packet, it is then sent to the router sub-element. If the router sub-element forwards the packet to the Customer-Facing IP interface, it sends the packet to the ARP sub-element to resolve the IPv4 network address to Ethernet MAC, encapsulates the packet in an Ethernet header, and sends the packet out the appropriate interface. If the router sub-element forwards the packet back to the Operator-Facing IP Interface, it is vendor-specific how to deal with the packet. Some implementations may choose to forward the packet back to the operator network; some may choose to drop the packet. Regardless, traffic should not be sent to a given eRouter from the operator network unless it is destined for a subnet known to the Customer-Facing IP interface.

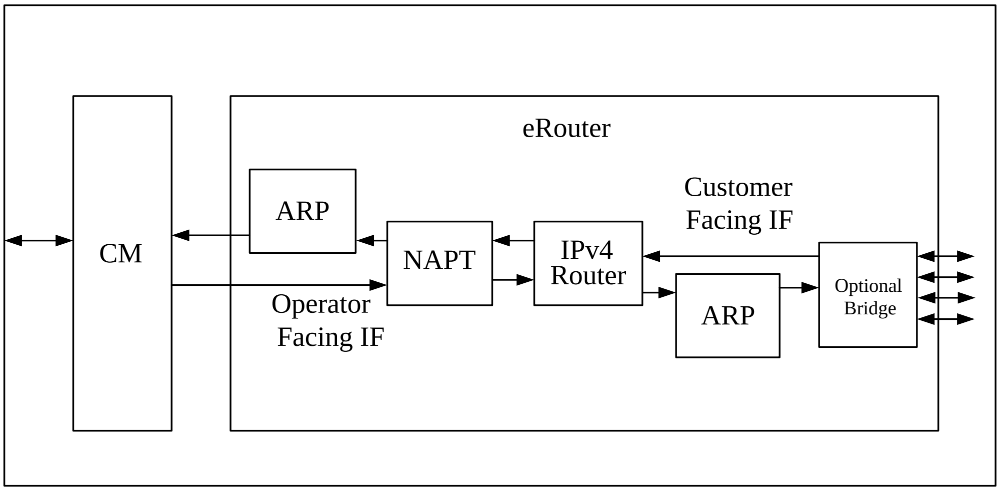


Figure 9-1 - eRouter IPv4 Forwarding Block Diagram

9.3 IPv4 Router[35]

When the eRouter's IPv4 Router sub-element receives a packet from its NAPT sub-element (received initially by its Operator-Facing IP Interface), it validates the IPv4 header in the packet. The eRouter MAY validate the IPv4 header in accordance with [RFC 1812], section 5.2.2. As defined in [RFC 1812], section 5.3.1, the eRouter MUST decrement the IP TTL field by at least one when forwarding the packet either back to the Customer-Facing IP interface, or out the Operator-Facing Interface. Packets forwarded to the eRouter's local IP stack for processing, MUST NOT decrement the TTL. Once the IPv4 header has been validated, the eRouter processes the destination IPv4 address of the packet. If the destination IPv4 address matches the eRouter's public address assigned to its Operator-Facing IP Interface, the eRouter sends the packet to its local IP stack for processing. If the destination IPv4 address does not match this address, the eRouter determines the next-hop address of the destination in order to forward the packet. The next-hop can be another router or a client directly connected to its Customer-Facing IP interface. The next-hop is determined by comparing the destination IPv4 address to the subnets assigned to its Customer-Facing IP interface. If the destination IPv4 address matches any of the prefixes assigned to the Customer-

[35] Revised per eRouter-N-13.1127-4 on 3/24/14 by PO, and per eRouter-N-14.1155-3 on 2/10/15 by JB.

Facing IP interface, the destination is considered directly connected, or "on-link", and the next-hop to use for ARP purposes is the destination IPv4 address. If it does not match, the destination is considered remote or "not on-link", and the next-hop to use for ARP purposes is the address of the internal router. Discovering other routers on Customer-Facing IP Interfaces, aside from knowledge derived via the use of Link ID when operating in Dual IP Protocol Enabled mode Section 9.3.1 is vendor-specific. If the eRouter cannot determine the next-hop of the IPv4 destination, then it MUST drop the packet.

When the eRouter's IPv4 Router sub-element receives a packet from its Customer-Facing IP interface, it validates the IPv4 header in the packet. The eRouter MAY validate the IPv4 header in accordance with [RFC 1812], section 5.2.2. As defined in [RFC 1812], section 5.3.1, the eRouter MUST decrement the IP TTL field by at least one when forwarding the packet, either back to the Customer-Facing IP Interface, or out the Operator-Facing Interface. Packets forwarded to the eRouter's local IP stack for processing MUST NOT decrement the TTL. Once the IPv4 header has been validated, the eRouter processes the destination IPv4 address of the packet. If the destination IPv4 address matches one of the private addresses assigned to the eRouter, it sends the packet to its local IP stack for processing. If the destination IPv4 address does not match one of these addresses, the eRouter determines the next-hop address of the destination in order to forward the packet. The next-hop can be another router or a client directly connected to either its Operator-Facing IP Interface, or back out its Customer-Facing IP Interface. The next-hop is determined by comparing the destination IPv4 address to the subnets assigned to the IP interface on which the eRouter is transmitting. If the destination IPv4 address matches a sub-net prefix, the destination is considered directly connected or "on-link", and the next-hop to use for ARP purposes is the destination IPv4 address. If it does not match, the destination is considered remote or "not on-link", and the next-hop to use for ARP purposes is the address of the intermediate router. The typical scenario for packets routed to the Operator-Facing IP Interface is that the next-hop router will be the eRouter's default, learned via DHCP, Section 7.2, which will be the CMTS. Discovering other routers, aside from the CMTS (or routing delegate chosen by the DHCP server if the CMTS is a bridge) on the Operator-Facing IP Interface, is vendor-specific. Discovery of other directly connected devices on the Operator-Facing IP Interface is also vendor-specific. The typical scenario for packets routed back out the Customer-Facing IP Interface is that the next-hop is a local host on a different subnet than that of the source, but directly connected to the eRouter. If the eRouter cannot determine the next-hop of the IPv4 destination address, it MUST drop the packet.

Regardless of whether the packet was received from the Customer-Facing IP Interface or the Operator IP Interface, the eRouter MUST generate an appropriate ICMP error message as described in [RFC 792] to identify the reason for dropping an IPv4 datagram, except in the follow cases

- The drop is due to congestion.
- The packet is itself an ICMPv4 error message.
- The packet is destined for an IPv4 broadcast or multicast address.
- The source IPv4 address of the packet is invalid as defined by [RFC 1812], section 5.3.7.
- The packet is a fragment and is not the first fragment (i.e., a packet for which the fragment offset in the IPv4 header is nonzero).

The eRouter's IPv4 router sub-element MUST process and/or generate the following ICMPv4 messages when appropriate:

0	Echo Reply	[RFC 792]
3	Destination Unreachable	[RFC 792]
11	Time Exceeded	[RFC 792]

NOTE: It is considered inappropriate for the eRouter's IPv4 router sub-element to generate ICMPv4 Destination Unreachable messages on the operator-facing interface.[36]

The eRouter MUST have at least one MAC address for its Operator-Facing IP Interface and one MAC address for its Customer-Facing IP Interface. The eRouter MUST share these source MAC addresses for IPv4 and IPv6. The eRouter MUST use the MAC address assigned to its Operator-Facing IP Interface as the source MAC address for all

[36] Revised per eRouter-N-11.0991-1 on 5/6/11 by JB.

packets that it sends out its Operator-Facing IP Interface. The eRouter MUST use the MAC address assigned to the Customer-Facing IP Interface as the source MAC address for all packets that it sends out its Customer-Facing IP Interfaces.

The eRouter MUST forward broadcast packets received on either interface only to the eRouter's IP stack. The eRouter MUST NOT forward broadcast packets received on either interface to any interface other than the eRouter's IP stack.

9.3.1 Dual IP Protocol and Link-ID Enabled Mode IPv4 Routing[37]

This section describes the requirements for IPv4 routing when the eRouter is in ‘Dual IP Protocol Enabled’ mode with Link ID enabled.

In order to install an IPv4 route to an IR, the eRouter does the following:

1. Calculates the IPv4 prefix to be used for the IR ‘Down’ interface(s) (LAN)
2. Uses the IR’s DHCPv4 assigned address as the next hop route address

Using Link-ID the eRouter MUST construct the IR route destination prefix using the first octet from the 10.0.0.0/8 aggregate prefix, the 16-bit Link ID from the IPv6 prefix delegated to the IR, and a prefix length that aligns with the length of the IA_PD. To align an IPv4 prefix length to an IPv6 prefix length, the eRouter subtracts 40-bits from the IPv6 prefix length delegated to that link and uses the result.

A unique IPv4 prefix is created using threes steps:

1. Use the decimal value 10 for the first octet
2. Convert IPv6 Link octets to their decimal equivalents for IPv4 octets 2 and 3
3. Determine the appropriate IPv4 prefix (subnet mask), from the given IPv6 prefix. For example, if the eRouter delegates prefix 2001:db8:1234:5601::/60 (Link ID 5601) to an internal router, the eRouter will assign the IPv4 prefix 10.86.01.0/20 for that IR. The decimal values 86 and 1 are equivalent to the values 0x56 and 0x01 from the Link ID). Setting the prefix size to /20 exactly maps the number of possible IPv6 links (16) to the number of possible IPv4 subnets (16), as illustrated in Figure 7-2 and in Annex E.

It is expected that IR¹s that support operations behind an eRouter will use the same Link-Id prefix calculation methods for IPv4 prefixes described for the eRouter when they receive a delegated IPv6 prefix so that the IPv4 addressing on the IR ‘Down’ interface(s) matches the expected values derived by the eRouter. Using methods other than those described in this section could result in unpredictable behavior.

9.4 NAPT

The eRouter MUST implement a NAPT function compliant with traditional Network Address Port Translation (NAPT) [RFC 3022], section 2.2. Per [RFC 3022], NAPT "is a method by which many network addresses and their TCP/UDP (Transmission Control Protocol/User Datagram Protocol) ports, are translated into a single network address and its TCP/UDP ports". Also, per [RFC 3022], the purpose of NAPT functionality is to "provide a mechanism to connect a realm with private addresses to an external realm, with globally unique registered addresses". The text in the NAPT sections below uses the term "public address(es)" to refer to the addresses reachable by the eRouter on its Operator-Facing IP Interface, assuming that they are globally-unique registered addresses. Note that an IP address that the eRouter views as globally unique, may be private to the operator's network. However, from the eRouter's perspective, these addresses are unique enough to ensure proper delivery to the next router upstream, and assumed to be globally unique.

Traditional NAPT is the simplest and most straightforward version of NAPT. Other versions that allow for mixtures of public and private network addresses on the Customer-Facing IP interface, or that allow users from the Operator-Facing IP Interface to establish translations to the Customer-Facing IP Interface, are not required by the eRouter and not discussed in this specification. Traditional NAPT requires that addresses used within the private network on

[37] Revised per eRouter-N-14.1155-3 on 2/10/15 by JB.

Customer-Facing IP Interfaces cannot overlap with any public addresses reachable by the Operator-Facing IP Interface. Therefore, the eRouter MUST use any of the private IPv4 network addresses described in [RFC 1918] for its Customer-Facing IP interface. The Customer-Facing IP Interface is considered to be a member of one private realm. A private realm is a single domain of private addresses. This means that an eRouter cannot connect to multiple private realms or private address domains.

The eRouter MAY advertise routes to destinations on the Operator-Facing IP Interface on the private network. The eRouter MUST NOT advertise routes to private destinations on the Customer-Facing IP Interface. Destinations on the Customer-Facing IP Interface MUST NOT be propagated onto the Operator-Facing IP Interface.

The eRouter MUST create NAPT translations dynamically based on receiving a packet from a private source on the Customer-Facing IP Interface attempting to access a public address on the Operator-Facing IP Interface, as described in Section 9.4.1.

For packets that traverse the NAPT function, the eRouter MUST always map a combination of private IPv4 address and port number to the same combination of public IPv4 address and port number. That is, the eRouter does not implement a symmetric Network Address Translation (NAT) as defined in[RFC 5389].

The eRouter MUST NOT create NAPT translations when public sources on the Operator-Facing IP Interface attempt to access private destinations on the Customer-Facing IP Interface. Connectivity between two devices that both live on the Customer-Facing IP Interface, but on different subnets, do not require NAPT translations, as they are required to be part of the same private realm. Therefore, the eRouter MUST NOT create NAPT translations to allow connectivity between CPEs that live on the Customer-Facing IP Interface.

In the following sections, the term Private Network Address Port (PNAP) refers to the network address and TCP/UDP port of a device on Customer-Facing IP interface that is using a private network address. The term Global Network Address Port (GNAP) refers to the network address and TCP/UDP port of that same device on Operator-Facing IP Interface after it has been translated by NAPT.

9.4.1 Dynamically Triggered NAPT Translations

Dynamically-triggered NAPT is invoked when a device on the Customer-Facing IP Interface with a private network address attempts to initiate one or more sessions to a public destination on the Operator-Facing IP Interface. In this case, the eRouter creates a mapping of source PNAP to GNAP and simultaneously creates a mapping of destination GNAP to PNAP for the return packets. The eRouter then replaces the source PNAP fields of the packet with its corresponding GNAP fields and forwards the packet out the Operator-Facing IP Interface. Once the external destination responds, the eRouter intercepts the reply and changes the previously inserted GNAP fields (now destination) back to the original PNAP values.

The eRouter MUST timeout dynamically-created NAPT translations to ensure that stale entries get removed. This timeout value MUST default to 300 seconds. This time value MAY be configurable. Other mechanisms can be used (like analyzing TCP session state) to time out the translations sooner, but the eRouter MUST still time out translations based on the timeout time in case the more advanced mechanism fails (e.g., because packet loss occurred and the eRouter did not see the final packets of a TCP flow).

9.4.2 Application Layer Gateways (ALGs)

Many applications are hampered by NAPT for various reasons. A common problem is the appearance of IPv4 address and/or port information inside the application payload that is too deep into the packet to be manipulated by NAPT, which operates at the network and transport layers. ALGs can be deployed to work around some of the problems encountered, but if the payload of such packets is secured, (by secure transport or application level security) the application cannot work. Another common reason NAPT causes problems is when applications exchange address/port information to establish new connections, creating interdependencies that NAPT cannot know about. The subsections following describe specific ALGs required by the eRouter.

9.4.2.1 ICMP Error Message ALG

ICMP error messages are required for the well-known trace-route network debugging tool to work across the eRouter. This ALG is described in detail in [RFC 3022], section 4.3. The ICMP error message ALG MUST be implemented by the eRouter. Briefly stated, the eRouter MUST translate both the outer and inner IPv4 headers in the ICMP error message in order for the protocol to work correctly, when packets traverse through the NAPT sub-element.

9.4.2.2 FTP ALG

FTP is a fairly widely-used protocol, so the FTP ALG is one of the most important ALGs. The issue with FTP is that it uses the body of the control session packets to signal the data session parameters, including the new TCP ports, to use for the data session. Since NAPT relies heavily on the TCP port field in order to translate between the private and public realm, this ALG is necessary to understand the new ports to be used by the ensuing data session. This ALG is described in detail in [RFC 3022], section 4.4. The FTP ALG MUST be implemented by the eRouter.

9.4.3 Multicast NAPT

IPv4 Multicast packets are a special case for NAPT and will need special handling at the eRouter. One scenario where forwarding of IP Multicast packets at the eRouter will need special handling is when a video source is using a private network address on a Customer-Facing IP Interface. In general, for video sources on the Customer-Facing IP Interface to work, the eRouter would be required to run at least one industry-standard multicast routing protocol to advertise the flows.

Since the eRouter will support IGMP proxy for IGMP v2 and v3, there is no reason to support a special translation for multicast packets in the eRouter for IGMP messages from private network addresses arriving on the Customer-Facing IP interface, as they will be consumed by the eRouter and new IGMP messages will be sent by the proxy agent from a public source network address on the Operator-Facing IP Interface.

9.5 ARP[38]

The ARP function in the eRouter MUST be compliant with the following RFCs:

- An Ethernet Address Resolution Protocol [RFC 826].
- Requirements for IP Version 4 Routers [RFC 1812], section 3.3.2.
- Requirements for Internet Hosts [RFC 1122], section 2.3.2.

The ARP function in the eRouter is limited to IPv4 network addresses (pln= 4) and Ethernet hardware addresses (hln=6). When the eRouter needs to forward an IPv4 packet to a given IP address on either the Operator-Facing IP Interface or the Customer-Facing IP Interface, it consults a table of IPv4 network addresses that each map to Ethernet addresses. If the corresponding IPv4 network address is found in the table, its corresponding Ethernet address MUST be used as the Ethernet destination address of the packet. If the corresponding IPv4 network address is not found, the eRouter MUST start the ARP protocol in hopes that it will learn the IPv4 network address to Ethernet address association. The eRouter MUST use its own MAC address, as described in Section 9.3, as the source MAC address and source hardware address of all ARP packets.

The eRouter dynamically creates ARP translations based on receiving ARP requests and/or replies for any of its IPv4 network addresses.

ARP entries maintained by the eRouter need careful examination before being aged. Both voice and video present humanly noticeable negative affects when ARP entries are removed in the middle of a session. [RFC 1122] suggests several different ways to age ARP entries in section 2.3.2.1. The eRouter SHOULD use option 2 – "Unicast polling", which allows for the ARP entry to stay fresh and in the ARP table as long as possible. This option is well-suited for routers that expect to have fairly small ARP tables and want long-term uninterrupted connectivity.

9.6 IPv4 Multicast

The eRouter learns IP multicast group membership information received on the Customer-Facing Interfaces and proxies it on the Operator-Facing Interface towards the next upstream multicast router. The eRouter forwards IPv4 multicast packets downstream based on the information learned at each Customer-Facing Interface.

[38] revised per eRouter-N-13.1127-4 on 3/24/14 by PO.

The eRouter proxies IGMP information upstream actively by implementing mutually-independent IGMPv3 router functionality on Customer-Facing Interfaces, and IGMPv3 group member functionality on the Operator-Facing Interface. On each IP interface, and independently of other IP interfaces, the eRouter generates, terminates, and processes IGMP messages according to IGMPv3 requirements. For example, the version of IGMP used on the cable network or the local area network will be defined locally at each network. The eRouter may send IGMPv2 reports on the Operator-Facing Interface while generating IGMPv3 queries on Customer-Facing Interfaces.[39]

The following elements define the eRouter IPv4 multicast behavior (also shown in Figure 9-2):

- An IGMPv3 Group Member that implements the group member part of IGMPv3[RFC 3376] on the Operator-Facing Interface.
- An IGMPv3 Router that implements the router portion of IGMPv3 [RFC 3376] on each Customer-Facing Interface.
- A subscription database per Customer-Facing Interface with multicast reception state of connected CPEs.
- An IPv4 Group Membership Database that merges subscription information from all the Customer-Facing Interfaces.

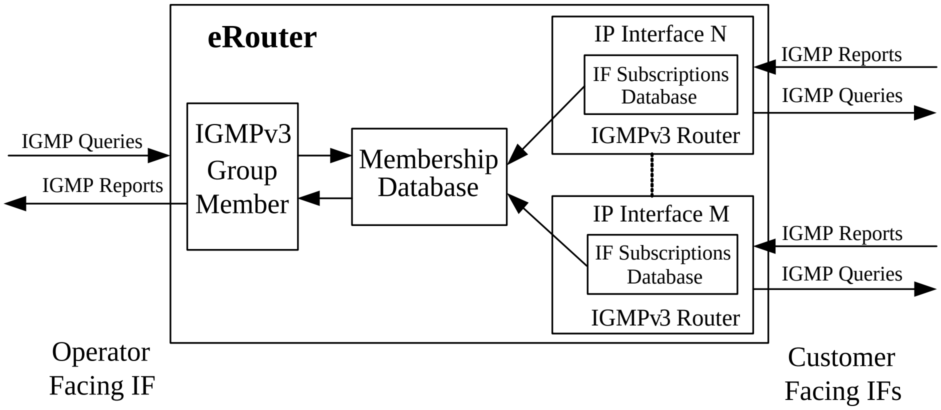


Figure 9-2 - eRouter IPv4 Multicast Forwarding Block Diagram

Central to the operation of the IGMPv3 Router(s) and IGMPv3 Group Member is the IPv4 Group Membership Database, through which the IGMPv3 Router(s) and IGMPv3 Group Member indirectly relate. This database condenses multicast reception state collected by the IGMPv3 Router(s) from connected CPEs. This information is used by the IGMPv3 Group Member on the Operator-Facing Interface as its own multicast reception interface state.

9.6.1 IGMP Proxying

The eRouter maintains the multicast reception state of CPEs on each Customer-Facing Interface in the interface's multicast subscription database. The eRouter obtains multicast reception state information of CPEs through the implementation of an IGMPv3 Router on each Customer-Facing Interface. Multicast reception state arrives at the eRouter in the form of IGMP Report messages transmitted by CPEs. The eRouter MUST implement the router portion of IGMPv3 [RFC 3376] on each Customer-Facing Interface. The eRouter MUST maintain, for each Customer-Facing Interface, the IPv4 multicast reception state of connected CPEs.

In the event of multiple queriers on one subnet, IGMPv3 elects a single querier-based on the querier IP address. However, the querier election rules defined for IGMPv3 do not apply to the eRouter. The eRouter MUST always act as an IGMP querier on its Customer-Facing Interfaces.

On the Operator-Facing Interface, the eRouter MUST implement the group member portion of IGMPv3 [RFC 3376]. The eRouter MUST merge the multicast reception state of connected CPEs into an IPv4 group membership database as described in Section 9.6.1.1. The eRouter MUST use the IPv4 group membership database

[39] Sentence modified by eRouter-N-06.0352-1 by kb 2/2/07.

as multicast reception interface state per [RFC 3376], section 3.2, on the Operator-Facing Interface. Thus, when the composition of the group membership database changes, the eRouter reports the change with an unsolicited report sent on the Operator-Facing Interface. When queried by an upstream multicast router, the eRouter also responds with information from the group membership database.

The eRouter MUST NOT perform the router portion of IGMPv3 on the Operator-Facing Interface.

9.6.1.1 IPv4 Group Membership Database

The eRouter's Membership Database is formed by merging the multicast reception state records of Customer-Facing Interfaces. In compliance with [RFC 3376], the eRouter keeps per Customer-Facing Interface and per multicast address joined one record of the form:

(multicast address, group timer, filter-mode, (source records))

With source records of the form:

(source address, source timer)

The eRouter keeps an IPv4 Group Membership Database with records of the form:

(multicast-address, filter-mode, source-list)

The eRouter uses the IPv4 Group Membership Database records as the interface state for the IGMPv3 Group Member implementation on the Operator-Facing Interface. Each record of the IPv4 Group Membership Database is the result of merging all subscriptions for that record's multicast-address on Customer-Facing Interfaces. For each IPv4 multicast group joined on any Customer-Facing Interface, the eRouter MUST abide by the following process to merge all customer interface records for the group, into one Group Membership Database record:

- First, the eRouter pre-processes all customer interface group records by:

  Converting IGMPv1 and IGMPv2 records into IGMPv3 records.

  Removing group and source timers from IGMPv3 and converted records.

  Removing every source whose source timer is greater than zero from records with a filter mode value of EXCLUDE.
- Then the eRouter creates an IPv4 Group Membership Database record by merging the pre-processed records, using the merging rules for multiple memberships on a single interface specified in section 3.2 of the IGMPv3 specification [RFC 3376].

9.6.2 IPv4 Multicast Forwarding[40]

The forwarding of IPv4 multicast packets received on any interface onto a Customer-Facing Interface is determined by the known multicast reception state of the CPEs connected to the Customer-Facing Interface. The eRouter MUST replicate an IPv4 multicast session on a Customer-Facing Interface, if at least one CPE device connected to the interface has joined the session. The eRouter MUST NOT replicate an IPv4 multicast session on a Customer-Facing Interface, if no CPE device connected to the interface has joined the session.

The eRouter MUST NOT forward IPv4 multicast packets received on any interface, i.e., any Customer-Facing or the Operator-Facing Interface, back to the same interface.

The eRouter MUST NOT forward IGMP messages received on any IP interface onto another IP interface.

The eRouter MUST forward IPv4 Local Scope multicast packets (239.255.0.0/16) to all Customer-Facing Interfaces within the same Customer-Facing IP Interface except the Customer-Facing Interface from which they were received.

[40] Section revised by eRouter-N-12.1085-2 on 3/5/13 by PO, and per eRouter-N-13.1102-3 on 5/24/13 by JB.

Except for IGMP packets and IPv4 administratively scoped (239.0.0.0/8) packets, the eRouter MUST forward all IPv4 multicast traffic received on Customer-Facing Interfaces onto the Operator-Facing Interface. Operator control of multicast traffic forwarding onto the cable network, if desired, can be done through the implementation of filters at the eCM.

9.6.3 IPv4 Multicast Forwarding Example

The eRouter in this example has two Customer-Facing Interfaces: CFIA and CFIB, connected to one LAN segment each. On CFIA, there are two CPEs connected: CPE1 and CPE2. CPE1 is IGMPv2 capable and will attempt to join group 224.0.100.1. CPE2 is IGMPv3 capable and will attempt to join group 224.128.100.1 from all sources. On CFIB, there is one CPE connected, CPE3, which is IGMPv3 capable and that will attempt to join group 224.128.100.1, except from source 198.200.200.200.

The router upstream of the eRouter (e.g., the CMTS) supports and is configured to operate in IGMPv3 mode, and thus the eRouter works in IGMPv3 mode on the Operator-Facing Interface.[41]

The setup is shown in Figure 9-3:

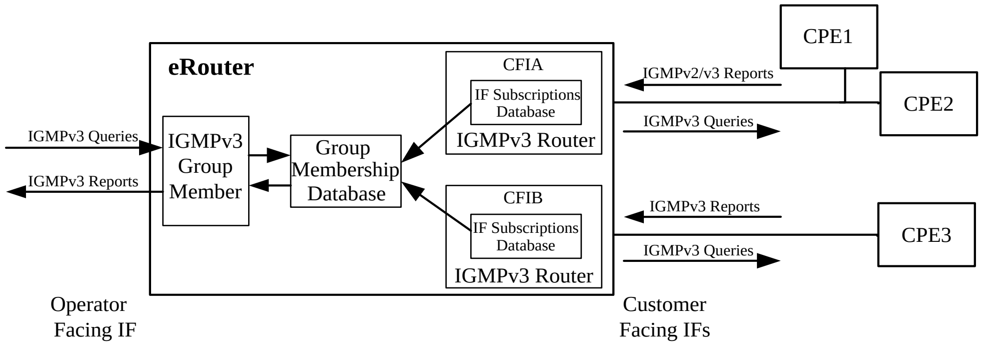


Figure 9-3 - IPv4 Multicast Forwarding Example[42]

The CPEs send reports as follows:[43]

Report From	Report Version	Multicast Address	Record Type	Source Address
CPE1	IGMPv2	224.0.100.1	N/A	N/A
CPE2	IGMPv3	224.128.100.1	EXCLUDE	Null
CPE3	IGMPv3	224.128.100.1	EXCLUDE	198.200.200.200

Because CPE1 sends an IGMPv2 report for group 224.0.100.1, CFIA operates in IGMPv2 compatibility mode for this group. On the other hand, CFIA and CFIB operate in IGMPv3 mode for group 224.128.100.1, because they receive IGMPv3 reports for this group from CPE2 and CPE3, respectively. The eRouter multicast reception state at each Customer-Facing Interface is the following:[44]

Interface	Multicast Address	Group Timer	Filter-Mode	Source Address	Source Timer
CFIA	224.0.100.1	A	EXCLUDE	Null	0
CFIA	224.128.100.1	B	EXCLUDE	Null	0
CFIB	224.128.100.1	C	EXCLUDE	198.200.200.200	0

[41] Paragraph added per eRouter-N-06.0352-1 by kb 2/2/07.
[42] Figure revised per eRouter-N-06.0352-1 by kb 2/2/07.
[43] This paragraph and table below modified per eRouter-N-06.0352-1 by kb 2/2/07.
[44] This paragraph and table below modified per eRouter-N-06.0352-1 by kb 2/2/07.

The interface state at the eRouter's Operator-Facing Interface, stored in the Group Membership Database, is the following:

Multicast Address	Filter-Mode	Source Address
224.0.100.1	EXCLUDE	Null
224.128.100.1	EXCLUDE	Null

The eRouter uses the information in the table above as multicast reception state at the Operator-Facing Interface. For example, in response to an IGMPv3 general query, the eRouter sends an IGMPv3 report for the two records shown.

Assuming that the CMTS is transmitting downstream four multicast streams, the eRouter forwards them as follows:[45]

Stream #	Multicast Address	Source Address	eRouter forwards on interfaces	
			CFIA	CFIB
1	224.0.200.2	198.100.100.100	NO	NO
2	224.0.100.1	198.100.100.100	YES	NO
3	224.128.100.1	198.100.100.100	YES	YES
4	224.128.100.1	198.200.200.200	YES	NO

9.7 Dual-Stack Lite Operation[46]

Even as operators migrate customers from IPv4 to IPv6 addressing, a significant percentage of Internet resources and content will remain accessible only through IPv4. As a consequence of the slow transition to IPv6 on the part of content providers, operators require mechanisms to allow customers to continue to access content and resources using IPv4 even after the last IPv4 allocations have been fully depleted. This necessitates multiplexing several customers behind a single IPv4 address. One technology that satisfies operator requirements for IPv4 address extension is Dual-Stack Lite.

Dual-Stack Lite enables an operator to share IPv4 addresses among multiple customers by combining two well-known technologies: IP in IP (IPv4-in-IPv6) tunneling and NAT. More specifically, Dual-Stack Lite encapsulates IPv4 traffic inside an IPv6 tunnel and sends it to an operator NAT device.

When Dual-Stack Lite is enabled, the eRouter acquires an IPv6 address on its Operator-Facing Interface and learns the address of the operator NAT device via DHCPv6. It encapsulates IPv4 traffic inside IPv6 sourced from its Operator-Facing Interface and destined for the operator NAT device.

To facilitate IPv4 extension over an IPv6 network, the eRouter MAY support Dual-Stack Lite.

If the eRouter supports Dual-Stack Lite, it MUST support Dual-Stack Lite B4 functionality as specified in Section 5 of [RFC 6333] with the exception of Section 5.3.

Requirements in Section 5.3 of [RFC 6333] are replaced by the following requirements:

The provisioning of DS Lite MUST be according to [RFC 6334], and request option code 64 (OPTION_AFTR_NAME) for the AFTR tunnel FQDN endpoint name.

9.7.1 Packet Fragmentation[47]

Packet fragmentation is necessary when an IPv4 packet enters the tunnel and the original packet size exceeds the tunnel MTU (which is 1460 Bytes). The original IPv4 packet is handled as follows.

[45] Table below modified per eRouter-N-06.0352-1 by kb 2/2/07.

[46] Section added per eRouter-N-09.0877-2 on 5/11/10 by JB. Revised per eRouter-N-11.1015-2 on 11/4/11 by JB; revised per eRouter-N-14.1150-2 on 7/17/14 by PO.

[47] Revised per eRouter-N-14.1233-1 on 2/13/15 by JB.

1. If in the original IPv4 packet header, the DF (Don't Fragment) flag is SET, the eRouter MUST discard the packet. It MUST return an ICMP message with type = 3 (unreachable), code = 4 (fragmentation needed and Don’t Fragment was set). The next hop MTU field MUST be set to the size of the tunnel MTU.

2. If in the original IPv4 packet header, the DF (Don't Fragment) flag is CLEAR, the eRouter MUST perform fragmentation of any IPv4 packet that exceeds 1460-bytes. The eRouter MAY fragment in one of two ways:

   a. Via [RFC 6333] Section 5.3 where the original IPv4 packet is encapsulated into the IPv6 payload before fragmentation.

   b. Via alternative method where the original IPv4 packet is fragmented first and then each fragment is placed into a separate IPv6 packet.

3. The method of fragmentation (a or b) MUST be configurable.

The eRouter MUST support TCP MSS clamping for IPv4 packets and MUST overwrite the TCP MSS with a value of 1420 Bytes.

10 IPV6 DATA FORWARDING

The normative requirements of this section are mandatory for an eRouter that implements the 'IPv6 Protocol Enabled' Mode and/or the 'Dual IP Protocol Enabled' mode, as defined in Section 6.

Assumptions[48]

- There is only a single Operator-Facing IP Interface on the eRouter.
- There is typically a single Customer-Facing IP Interface on the eRouter.
- The Operator-Facing IP Interface is Ethernet encapsulated.
- The Customer-Facing IP Interface is Ethernet encapsulated.
- The eRouter advertises itself as a router (using ND) on all Customer-Facing Interfaces so clients and routers learn about the eRouter. The eRouter does not send Router Advertisements on its Operator-Facing Interface as they would be discarded by the eCM.
- All the eRouters are on separate links and therefore will not see each other's RAs.

10.1 Overview

The IPv6 eRouter is responsible for implementing IPv6 routing. This includes looking up the IPv6 Destination address to decide which of the eRouter interfaces to send the packet.

The ND protocol is required on the eRouter. Like ARP in IPv4, it provides a mechanism for converting IPv6 network addresses to Ethernet MAC addresses on both the Customer-Facing IP interfaces and the Operator-Facing IP Interface. It also provides a mechanism for the eRouter to advertise its presence, host configuration parameters, routes, and on-link preferences.

Figure 10-1 shows a block diagram of the IPv6 eRouter with an IPv6 Router block and an ND block. The IPv6 functionality, however, does not have the clean separation indicated by these blocks. The IPv6 Routing and Neighbor Discovery blocks are closely intertwined and, therefore, are discussed together under the same subsection.

The IPv6 eRouter uses a local IPv6 routing table to forward packets. The eRouter creates the IPv6 routing table upon initialization of the IPv6 portion of the eRouter and adds entries according to the receipt of Router Advertisement messages containing on-link prefixes and routes.

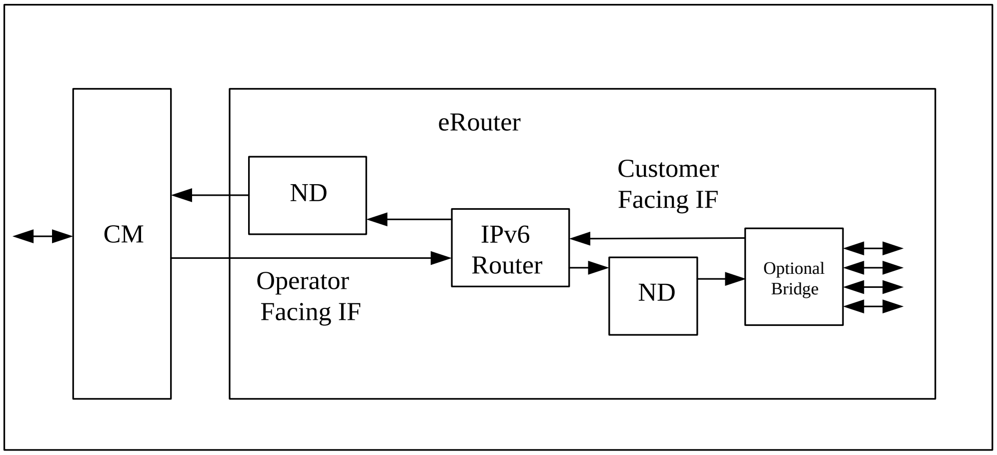


Figure 10-1 - eRouter IPv6 Forwarding Block Diagram

[48] eRouter-N-13.1088-2 modified this section on 3/5/13 by PO.

10.2 System Description[49]

Except when noted, the ND function in the eRouter MUST comply with the Neighbor Discovery for IPv6 [RFC 4861]. Per [RFC 4861], ND is used "to determine the link-layer addresses for neighbors known to reside on attached links and to quickly purge cached values that become invalid."

Several sections of [RFC 4861] do not apply to the eRouter. These sections include:

section 6.2.7 - RA Consistency

section 6.2.8 - Link-local Address Change

section 7.2.8 - Proxy Neighbor Advertisements

section 8 - Redirect Function

section 11 - Security Considerations

section 12 - Renumbering Considerations

The eRouter MUST support the following ND messages per [RFC 4861]: Router Solicitation, Router Advertisement, Neighbor Solicitation, and Neighbor Advertisement.

The eRouter receives a packet and checks the destination address of the packet. If the destination IPv6 address matches the address assigned to the eRouter's IP interface, the eRouter forwards the packet to its local IP stack for processing. If the destination IPv6 address does not match the eRouter's address, the eRouter determines the next-hop address of the destination in order to forward the packet. The next-hop can be a router, or the destination itself. The next-hop is determined by comparing the destination IPv6 address to the prefixes assigned to the IP interfaces on which the eRouter is communicating. If the destination IPv6 address matches a sub-net prefix, the destination is considered directly connected or "on-link", and the next-hop to use for ND purposes is the destination IPv6 address. If the address of the packet does not match, the destination is considered remote or "not on-link", and the next-hop to use for ND purposes is the address of the intermediate router. If there is no intermediate router, the eRouter MUST immediately drop the packet.

The typical scenario for packets routed to the Operator-Facing IP Interface is that the next-hop router will be the eRouter's default router address, learned via Router Advertisement [RFC 3315], from the CMTS. Discovering other routers, aside from the CMTS (or routing delegate if the CMTS is a bridge), on the Operator-Facing IP Interface is vendor-specific. Discovery of other directly-connected devices on the Operator-Facing IP Interface is also vendor-specific. The typical scenario for packets routed back out the Customer-Facing IP Interface is that the next-hop is a local host on a different subnet than that of the source, but directly connected to the eRouter. If the eRouter cannot determine the next-hop of the IPv6 destination address, then it MUST immediately drop the packet.

Once a next-hop is determined, the eRouter's Neighbor Cache is consulted for the link-layer address of the next-hop address. If necessary, address resolution is performed. Address resolution is accomplished by multicasting a Neighbor Solicitation that prompts the addressed neighbor to return its link-layer address in a Neighbor Advertisement. The neighbor cache entry is then updated with this link-layer address, and the eRouter then forwards the packet to the link-layer address contained in this cache entry. If an error occurs at any point in the process, the eRouter discards the packet. Regardless of whether the packet was received from the Customer-Facing IP Interface or the Operator IP Interface, the eRouter MUST generate an appropriate ICMP error message, as described in [RFC 4884], to identify the reason for dropping an IPv6 datagram, except in the follow cases:

- The drop is due to congestion.
- The packet is itself an ICMPv6 error message.
- The packet is destined for an IPv6 multicast address (except if the packet is the "Packet Too Big Message" or the "Parameter Problem Message", as explained in [RFC 4884], section 2.4, paragraph (e)).
- The packet is destined for a link-layer multicast address.

[49] Revised per eRouter-N-09.0877-2 on 5/11/10 and per eRouter N-11.1015-2 on 11/4/11 by JB, revised per eRouter-N-13.1127-4 on 3/24/14 by PO, revised per eRouter-14.1134-2 on 3/25/14 by PO.

- The source IPv6 address of the packet does not uniquely identify a single node, as explained in detail in [RFC 4884], section 2.4, paragraph (e).
- The eRouter MUST process and/or generate the following ICMPv6 messages when appropriate:

1	Destination Unreachable	[RFC 4443]
3	Time Exceeded	[RFC 4443]
129	Echo Reply	[RFC 4443]
130	Multicast Listener Query	[RFC 3810]
131	Multicast Listener Report	[RFC 3810]
132	Multicast Listener Done	[RFC 3810]
133	Router Solicitation	[RFC 4861]
134	Router Advertisement	[RFC 4861]
135	Neighbor Solicitation	[RFC 4861]
136	Neighbor Advertisement	[RFC 4861]
143	Version 2 Multicast Listener Report	[RFC 3810]

NOTE: It is considered inappropriate for the eRouter to generate ICMPv6 Destination Unreachable messages on the operator-facing interface.[50]

The IPv6 CE router MUST implement ICMPv6 according to [RFC 4443].

The eRouter is responsible for decrementing the Hop Limit field in the IPv6 packet that it is going to forward. If the eRouter receives an IPv6 packet with a Hop Limit of zero, or the eRouter decrements an IPv6 packet's Hop Limit to zero, it MUST discard that packet and send an ICMPv6 Time Exceeded message with Code 0 to the source of that IPv6 packet.

The eRouter is also responsible for reinserting the Ethernet header of IPv6 packets. The eRouter has at least one MAC address for its Operator-Facing IP Interface and one MAC address for its Customer-Facing IP Interface that are shared for IPv4 and IPv6 (see Section 8.3). The eRouter MUST use the MAC address assigned to its Operator-Facing IP Interface as the source MAC address for all IPv6 packets that it sends out its Operator-Facing IP Interface. The eRouter MUST use the MAC address assigned to the Customer-Facing IP Interface as the source MAC address for all IPv6 packets that it sends out its Customer-Facing IP Interfaces. Per [RFC 4861], the eRouter uses the MAC address of the next-hop address learned via Neighbor Discovery as the destination MAC address for the IPv6 packet.

The eRouter MUST forward link-local multicast packets received on either interface only to the eRouter's IP stack. The eRouter MUST NOT forward link-local multicast packets received on either interface to any interface other than the eRouter's IP stack.

By default, an eRouter MUST NOT initiate any IPv4 or IPv6 dynamic routing protocols on its Operator-facing interface.

10.3 IPv6 Multicast

The eRouter learns IP multicast group membership information received on the Customer-Facing Interfaces and proxies it on the Operator-Facing Interface towards the next upstream multicast router. The eRouter forwards IPv6 multicast packets downstream based upon the information learned at each Customer-Facing Interface.

The eRouter proxies MLD information upstream actively by implementing mutually-independent MLDv2 router functionality on Customer-Facing Interfaces and MLDv2 multicast listener functionality on the Operator-Facing Interface. On each IP interface, and independently of other IP interfaces, the eRouter generates, terminates, and processes MLD messages according to MLDv2 requirements. For example, the version of MLD used on the cable network or the local area network will be defined locally at each network. The eRouter may send MLDv1 reports on the Operator-Facing Interface while generating MLDv2 queries on Customer-Facing Interfaces.[51]

[50] Revised per eRouter-N-11.0991-1 on 5/6/11 by JB.

[51] Sentence modified per eRouter-N-06.0352-1 by kb 2/2/07.

The following elements define the eRouter IPv6 multicast behavior (also shown in Figure 10-2):

- An MLDv2 Multicast Listener that implements the multicast listener part of MLDv2 [RFC 3810] on the Operator-Facing Interface;
- An MLDv2 Router that implements the router part of MLDv2 [RFC 3810] on each Customer-Facing Interface;
- A Subscription Database per Customer-Facing Interface with multicast reception state of connected CPEs;
- An IPv6 Group Membership Database that merges subscription information from all the Customer-Facing Interfaces.

These logical sub-elements are shown in Figure 10-2.

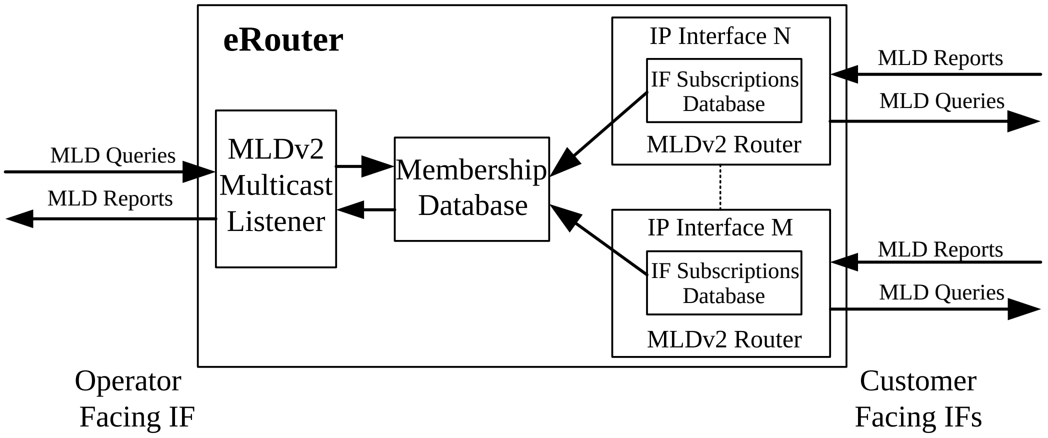


Figure 10-2 - eRouter IPv6 Multicast Forwarding Block Diagram

10.3.1 MLD Proxying

The eRouter maintains the multicast reception state of CPEs on each Customer-Facing Interface in the interface's multicast subscription database. The eRouter obtains CPE's multicast reception state information through the implementation of an MLDv2 Router on each Customer-Facing interface. Multicast reception state arrives at the eRouter in the form of MLD Report messages transmitted by CPEs. The eRouter MUST implement the router portion of the MLDv2 protocol, [RFC 3810], on each Customer-Facing Interface. The eRouter MUST maintain, for each Customer-Facing Interface, the IPv6 multicast reception state of connected CPEs.

In the event of multiple queriers on one subnet, MLDv2 elects a single querier based on the querier IP address. However, the querier election rules defined for MLDv2 do not apply to the eRouter. The eRouter MUST always act as an MLD querier on its Customer-Facing Interfaces.

On the Operator-Facing Interface, the eRouter MUST implement the multicast listener portion of the MLDv2 protocol, [RFC 3810]. The eRouter MUST merge the multicast reception state of connected CPEs into an IPv6 group membership database, as described in Section 10.3.2, IPv6 Group Membership Database. The eRouter MUST use the membership database as multicast reception interface state per [RFC 3810], section 4.2, for the Operator-Facing Interface. Thus, when the composition of the IPv6 multicast membership database changes, the eRouter reports the change with an unsolicited report sent on the Operator-Facing Interface. When queried by an upstream multicast router, the eRouter also responds with information from the membership database.

The eRouter MUST NOT perform the router portion of MLDv2 on the Operator-Facing Interface.

10.3.2 IPv6 Group Membership Database

The eRouter's Membership Database is formed by merging the multicast reception state records of Customer-Facing Interfaces. In compliance with [RFC 3810], the eRouter keeps per Customer-Facing Interface and per multicast address joined one record of the form:

(multicast address, group timer, filter-mode, (source records))

With source records of the form:

(source address, source timer)

The eRouter keeps an IPv6 Group Membership Database with records of the form:

(multicast-address, filter-mode, source-list)

The eRouter uses the IPv6 Group Membership Database records as interface state for the MLDv2 Multicast Listener implementation on the Operator-Facing Interface. Each record of the IPv6 Group Membership Database is the result of merging all subscriptions for that record's IPv6 multicast-address on Customer-Facing Interfaces. For each IPv6 multicast group joined on any Customer-Facing Interface, the eRouter MUST abide by the following process to merge all customer interface records for the group into one Group Membership Database record:

- First, the eRouter pre-processes all customer interface group records by:

  Converting MLDv1 records into MLDv2 records.

  Removing group and source timers from MLDv2 and converted records.

  Removing every source whose source timer is greater than zero from records with a filter mode value of EXCLUDE.
- Then the eRouter creates an IPv6 Group Membership Database record by merging the pre-processed records, using the merging rules for multiple memberships on a single interface specified in section 4.2 of the MLDv2 specification [RFC 3810].

10.3.3 IPv6 Multicast Forwarding[52]

The forwarding of IPv6 multicast packets received on any interface onto a Customer-Facing Interface is determined by the known multicast reception state of the CPEs connected to the Customer-Facing Interface. The eRouter MUST replicate an IPv6 multicast session on a Customer-Facing Interface if at least one CPE device connected to the interface has joined the session. The eRouter MUST NOT replicate an IPv6 multicast session on a Customer-Facing Interface if no CPE device connected to the interface has joined the session.

The eRouter MUST NOT forward IPv6 multicast packets received on any interface, i.e., any Customer-Facing or the Operator-Facing Interface, back to the same interface.

In compliance with IPv6 link-scope packet forwarding rules, the eRouter MUST NOT forward MLD messages received on an IP interface onto another IP interface. Also, the eRouter MUST NOT forward link-scoped IPv6 multicast packets received on an IP interface onto another IP interface.

The eRouter MUST forward site-scoped IPv6 multicast packets to all Customer-Facing Interfaces within the same Customer-Facing IP Interface except the Customer-Facing Interface from which they were received.

The eRouter MUST forward all non-link-scoped and non-site-scoped (e.g., not addressed to FF02::/16 or FF05::/16) IPv6 multicast traffic received on Customer-Facing Interfaces onto the Operator-Facing Interface. Operator control of multicast traffic forwarding onto the cable network, if desired, can be done through the implementation of filters at the eCM.

10.3.4 IPv6 Multicast Forwarding Example

The eRouter in this example has two Customer-Facing Interfaces: CFIA and CFIB, connected to one LAN segment each. On CFIA, there are two CPEs connected: CPE1 and CPE2. CPE1 is MLDv1-capable and will attempt to join group FF1E::100. CPE2 is MLDv2-capable and will attempt to join group FF1E::128 from all sources. On CFIB, there is one CPE connected, CPE3, which is MLDv2 capable and that will attempt to join group FF1E::128, except from source 3FFE:2900::200.

The router upstream of the eRouter (e.g., the CMTS) supports and is configured to operate in MLDv2 mode, and thus the eRouter works in MLDv2 mode on the Operator-Facing Interface.

[52] Section modified per eRouter-N- 12.1085-2 by PO 3/5/13 and per eRouter-N-13.1102-3 on 5/24/13 by JB.

The setup is shown in Figure 10-3:

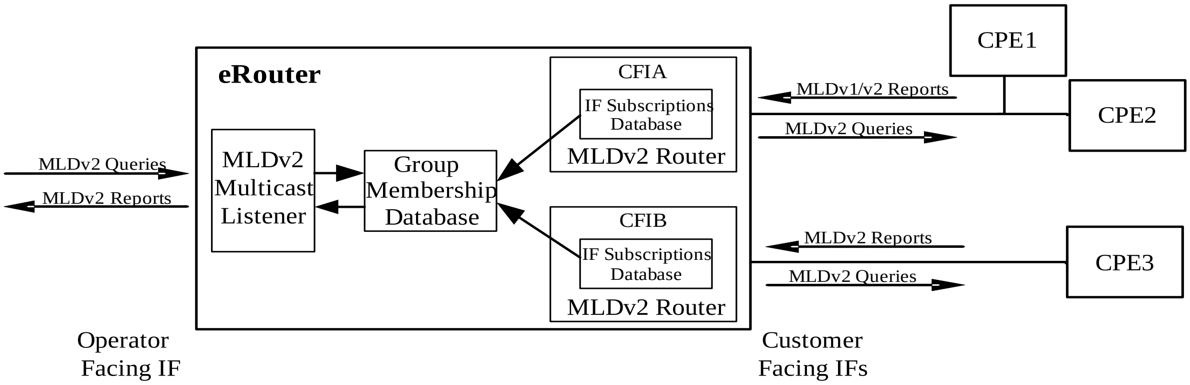


Figure 10-3 - IPv6 Multicast Forwarding Example[53]

The CPEs send reports as follows:[54]

Report From	Report Version	Multicast Address	Record Type	Source Address
CPE1	MLDv1	FF1E::100	N/A	N/A
CPE2	MLDv2	FF1E::128	EXCLUDE	Null
CPE3	MLDv2	FF1E::128	EXCLUDE	3FFE:2900::200

Because CPE1 sends an MLDv1 report for group FF1E::100, CFIA operates in MLDv1 compatibility mode for this group. On the other hand, CFIA and CFIB operate in MLDv2 mode for group FF1E::128, because they receive MLDv2 reports for this group from CPE2 and CPE3, respectively. The eRouter multicast reception state at each Customer-Facing Interface is the following:[55]

Interface	Multicast Address	Group Timer	Filter-Mode	Source Address	Source Timer
CFIA	FF1E::100	A	EXCLUDE	Null	0
CFIA	FF1E::128	B	EXCLUDE	Null	0
CFIB	FF1E::128	C	EXCLUDE	3FFE:2900::200	0

The eRouter merges the multicast reception state of connected CPEs shown above into the Group Membership Database as follows:

Multicast Address	Filter-Mode	Source Address
FF1E::100	EXCLUDE	Null
FF1E::128	EXCLUDE	Null

The eRouter uses the information in the Group Membership Database as multicast reception state at the Operator-Facing Interface. For example, in response to an MLDv2 general query, the eRouter sends an MLDv2 report for the two records shown.

[53] Figure modified per eRouter-N-06.0352-1 by kb 2/2/07.
[54] This paragraph and table below modified per eRouter-N-06.0352-1 by kb 2/2/07.
[55] This paragraph and table below modified per eRouter-N-06.0352-1 by kb 2/2/07.

Assuming that the CMTS is transmitting four multicast streams downstream, the eRouter forwards them as follows:[56]

Stream #	Multicast Address	Source Address	eRouter forwards on interfaces	
			CFIA	CFIB
1	FF1E::200	3FFE:2900::100	NO	NO
2	FF1E::100	3FFE:2900::100	YES	NO
3	FF1E::128	3FFE:2900::100	YES	YES
4	FF1E::128	3FFE:2900::200	YES	NO

[56] Table below modified per eRouter-N-06.0352-1 by kb 2/2/07.

11 QUALITY OF SERVICE[57]

QoS on the eRouter is optional. The eRouter SHOULD support Layer 2 and Layer 3 QoS, as defined in this section. The QoS functionality described herein allows the operator to selectively provide a level of differentiation among the various data streams destined for CPE behind the eRouter. Typical applications could include Internet Protocol Television (IPTV) services and other enhanced data services, though it is anticipated that overall packet counts will still be dominated by largely undifferentiated best-effort data traffic.

If the eRouter supports QoS, the eRouter MUST prioritize the forwarding of IP packets based on the values marked in the IPv4 ToS byte or IPv6 Traffic Class field. This is because Layer 2 (e.g., 802.1 p/Q Ethernet) headers will be removed as the packets traverse the eRouter.

11.1 Downstream Quality of Service Operation[58]

This section deals with the requirements regarding traffic going to CPEs, through the eRouter, from the Cable network.

If the eRouter supports QoS, the eRouter MUST provide two or more priority queues on each Customer-Facing Interface for traffic going to CPEs. The eRouter MAY provide a configuration mechanism to map ToS/Traffic Class field priority values to the high and low priority queues. As a default setting, the eRouter might use the most significant bit of the ToS/Traffic Class field to determine priority to queue mappings.

11.2 Upstream Quality of Service Operation[59]

This section deals with traffic coming from the CPEs attached to the eRouter to the cable network.

For the purposes of applying QoS to upstream traffic sourced from CPE devices, the interface between the eRouter and the embedded CM is considered to be of infinite bandwidth per [eDOCSIS], and thus no congestion, control, priority, nor reservation of bandwidth resources should be expected to occur on this interface. Thus, the eRouter does not need to provide any queues in the upstream direction. The eRouter MAY provide a configuration mechanism to determine whether the eRouter allows CPE devices to pass QoS-tagged packets with the IP ToS/Traffic Class field intact, or whether the eRouter resets the IP ToS/Traffic Class field to 0. The eRouter MAY use the IP ToS/Traffic Class field to populate Layer 2 QoS headers to ensure upstream QoS treatment. Although other implementations are possible, one such implementation is to directly map the three most significant bits of the IP ToS/Traffic Class field into the 802.1Q priority field.

In the case where multiple Customer-Facing Interfaces are implemented, the eRouter may support additional QoS mechanisms to prioritize upstream traffic based on ingress interface.

[57] Section modified by eRouter-N-13.1127-4 on 3/24/14 by PO.
[58] Section modified by eRouter-N-13.1127-4 on 3/24/14 by PO.
[59] Section modified by eRouter-N-13.1127-4 on 3/24/14 by PO. Revised per eRouter-N-14.1233-1 on 2/13/15 by JB

12 eRouter Management[60]

The eRouter allows the implementation of different management interfaces as described in this section. Management interfaces in this specification refer to the protocols, data models, and semantic representation of the data exchange to perform the conventional management functions in the device.

The eRouter MUST support either SNMP [RFC 3412] or TR-069 [TR-069a4] from the Operator-Facing management interface.

The eRouter is not required to support both management interfaces simultaneously for a given system boot instance.

User management from the Customer-Facing interface is vendor specific. Remote management of the eRouter (from the Operator-Facing interface) by the customer is outside of the scope of this specification.

Other specifications referring to the eRouter specification might add requirements to the eRouter management interface for additional functionality.

12.1 eRouter SNMP Management Interface Requirements

The eRouter SNMP Management Interface requirements are listed in Annex A and Annex B. Annex A lists the management objects requirements for the eRouter to support. Annex B, sections B.1, B.4.5, and B.4.6 provide the provisioning elements to secure the SNMP access control by SNMP entities.

If SNMP is supported from the Operator-Facing Interface, the eRouter MAY support SNMP [RFC 3412] from the Customer-Facing Interface.

12.2 eRouter TR-069 Management Interface Requirements

The eRouter TR-069 Management Interface requirements are listed in Annex D.

If TR-069 is supported from the Operator-Facing Interface, the eRouter MAY support [RFC 3412] for Customer-Facing Interface management.

12.2.1 ACS Discovery

The eRouter performs initial ACS discovery via the mechanisms in the following sections.

12.2.1.1 eRouter TR-069 Management Server Configuration File TLV Encapsulation

The eRouter MUST support the TR-069 Management Server Configuration File TLV Encapsulation as defined in Annex B.4.3.2 for ACS selection.

12.2.1.2 TR-069 Management Server URL DHCP Option

The eRouter MAY support the DHCPv4 option CL_V4OPTION_TR-069_MANAGEMENT_SERVER_URL, and DHCPv6 option CL_OPTION_TR-069_MANAGEMENT_SERVER_URL [CANN DHCP] for the ACS selection as part of the eRouter DHCP ACK and/or REPLY messages. The eRouter MUST follow the DHCP requirements in [TR-069a4] for the initial ACS discovery with the exception of using the CableLabs-defined DHCP options.

12.2.2 ACS Selection

If the TR-069 Management Server URL is present in only one of TR-069 Management Server Configuration File TLV Encapsulation or TR-069 Management Server URL DHCP Option, the eRouter MUST use the present URL as the initial ACS URL. If the TR-069 Management Server URL is present in both TR-069 Management Server Configuration File TLV Encapsulation and TR-069 Management Server URL DHCP Option, the eRouter MUST use the former as the ACS URL. If the TR-069 Management Server URL is present in neither the CM configuration file nor the DHCP Offer/Response, the eRouter MUST NOT communicate with any ACS.

[60] Section added per eRouter N-11.1014-3 on 10/27/11 by JB.

12.2.3 Dynamic ACS Updates

After the initial discovery, the ACS URL can be changed by updating the Device.ManagementServer.URL attribute value. The eRouter MUST ignore the ACS URL if it is present in DHCP renew/rebind messages.

12.2.4 TR-069 CWMP Control and Credentials[61]

The TR-069 Device.ManagementServer object defines controls for CWMP operations and credentials for authentication of connection requests between the CPE and ACS. All TR-069 Device.ManagementServer objects can be configured by the ACS via [TR-069a4] procedures.

In addition, the parameter Device.ManagementServer.URL can be delivered via DHCP or Configuration File TLV, as specified in Section 12.2.2.

For security reasons, the TR-069 Device.ManagementServer object credential attributes (Username, Password, ConnectionRequestUsername and ConnectionRequestPassword) are also configurable via the TR-069 Management Server Configuration File TLV Encapsulation (see Annex B.4.3).

To prevent dead-lock situations that would require user interventions, the Device.ManagementServer.EnableCWMP is also configurable via the TR-069 Management Server Configuration File TLV Encapsulation (see Annex B.4.3.1).

[61] Section modified by eRouter-N-13.1127-4 on 3/24/14 by PO.

13 SECURITY [62]

It is considered a best practice to filter obviously malicious traffic (e.g., spoofed packets, "Martian" addresses, etc.). Thus, the eRouter ought to support basic stateless egress and ingress filters. The eRouter is also expected to offer mechanisms to filter traffic entering the customer network; however, the method by which vendors implement configurable packet filtering is beyond the scope of this document.

The eRouter MUST enable a stateful firewall by default. In particular, the eRouter SHOULD support functionality sufficient for implementing the set of recommendations in [RFC 6092], section 4. The eRouter MUST support ingress filtering in accordance with BCP 38 [RFC 2827].

[RFC 6092] contains 50 "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service." Not all of these recommendations are applicable to MSO networks. Of the applicable recommendations, not all are needed immediately. In order to ensure that vendors are able to implement "simple security" support in eRouter devices, Annex F categorizes the recommendations into five requirement categories:

- Critical - Critical to network connectivity. Include in initial release.
- Important - Failure to implement could open subscribers to infosec attack.
- BCP - Security best practice / nice to have but not critical.
- Other - MSOs have indicated ambivalence to this category of recommendations.
- Conflict - Recommendation conflicts with MSO needs and requires modification or should not be implemented.

[62] Added per eRouter-N-11.1015-2 on 11/8/11 by JB, revised per 12.1081-2 on 1/4/13 by PO.

14 eRouter Tunnel Management and Configuration [63]

14.1 GRE Requirements

Some of the applications envisioned for the eRouter rely upon the tunneling of traffic between the customer's service location and the Operator's network core. For example, a community Wi-Fi application that utilizes one or more SSIDs to provide public WiFi could be configured to tunnel its traffic to a central concentrator within the Operator's network core. While multiple tunneling protocols and techniques exist, Generic Route Encapsulation (GRE) tunneling see [RFC 2784] and [RFC 2890] has become the prevalent method for conveying traffic to the Operator's core. In order to support the management and configuration of these GRE tunnels, this specification defines both SNMP based MIBs and TR-181 data model elements. The SNMP MIB defined here originated from the [TR-181] data model profiles for GRE tunnels.

An eRouter that implements GRE over IPv4 SHOULD support [RFC 2784].

An eRouter that implements GRE over IPv6 SHOULD support [RFC 2890].

An eRouter that supports GRE tunneling MUST support

- the TR-181 GRE data model elements found in [TR-181]
- the CLAB-GRE-MIB defined in Annex A.4.

When the eRouter is provisioned via the GRE tunneling MIB Annex A.4, it MUST permit the index of the WiFi SSID to be set to the index numbers defined in Annex A.1.

When the eRouter is provisioned via the [TR-181] GRE tunneling profile, it MUST permit the index of the WiFi SSID to be set to the index numbers defined in Annex A.2.

[63] Section adder per eRouter-N-15.1235-3 on 2/23/15 by PO.

Annex A SNMP MIB Objects supported by the eRouter[64]

This Annex defines the SNMP MIBs that the eRouter is required to implement.

The eRouter MUST support the following MIB objects:

- ifTable [RFC 2863]
- ipNetToPhysicalTable [RFC 4293];
- vacmAccessTable [RFC 3415];
- vacmSecurityToGroupTable [RFC 3415];
- vacmViewTreeFamilyTable [RFC 3415];
- vacmAccessReadViewName [RFC 3415];
- vacmAccessWriteViewName [RFC 3415];
- snmpCommunityTable [RFC 3584];
- snmpTargetAddrTable [RFC 3413]
- snmpTargetAddrTAddress [RFC 3413];
- snmpTargetAddrTMask [RFC 3584];
- snmpTargetAddrExtTable [RFC 3584];
- esafeErouterInitModeControl [eDOCSIS].

Additional information for the configuration and use of the above MIB objects is defined in Annex B.

A.1 eRouter Interface Numbering[65]

The eRouter MUST use in its MIB tables, when appropriate, an ifIndex number of '1' for the Operator-Facing Interface and an ifIndex number of '2' for the first Customer-Facing Interface. The eRouter MUST use an ifIndex number in accordance with Table A-1 for any additional Customer-Facing Interfaces.

Table A-1 - eRouter Interface Numbering

Interface	Type
1	Primary CPE interface (eRouter Operator-Facing Interface), when eRouter is enabled
2 - 4	Reserved
5 - 15	Ethernet interfaces
16 - 31	Reserved
32 - 39	USB interfaces
40 – 47	MoCA interfaces
48 - 199	Reserved
200 – 299	Customer-Facing IP interfaces
300 – 399	Operator-Facing IP interfaces
400 – 499	GRE tunnel interfaces
1xxyy	WiFi and SSID interfaces (where xx corresponds to the WiFi radio interface (0 – 99), and yy corresponds to the SSID logical interface for WiFi radio xx with yy in the range 1 – 99)
500 - 599	Additional eRouter CPE interfaces
600 - 699	eRouter internal interfaces (optional)

[64] Modified per eRouter-N-13.1099-1 on 3/20/13 by PO.Revised per eRouter-N-14.1217-3 on 2/12/15 by JB.
[65] Section modified per eRouter-N-14-121703, table updated per eRouter-N-15.234-2 on 2/23/15 by PO.

eRouter devices that include one or more WiFi radios MUST follow the interface numbering and naming conventions specified in section 6.2.1 of [WI-FI MGMT] .

A.2 eRouter ifTable Requirements[66]

The eRouter MUST implement the row entry specified in Table A-2 for the ifTable as specified in [RFC 2863].

Table A-2 - eRouter ifTable Row Entries

ifTable Object	Ethernet	USB	MoCA	CFI IP	OFI IP	GRE	WiFi	SSID
Ifindex	5 - 15	32- 39	40 - 47	200 - 299	300 - 399	400 - 499	Per Annex A.1	Per Annex A.1
Ifdescr	Unspecified	Unspecified	Unspecified	Unspecified	Unspecified	Unspecified	Wifi Radio Interface	Wifi SSID Sub-Interface
Iftype	Ethernetcsmacd	Usb	Moca	Ipforward	Ipforward	Tunnel	Ieee80211(71)	Ieee80211(71)
Ifmtu	0	0	0	0	0	0	0	0
Ifspeed	0	0	0	0	0	0	0	0
Ifphysaddress	Ethernet Physical Address	USB Physical Address	Moca Physical Address	CFI IP MAC Address	OFI IP MAC Address	MAC Associated With Tunnel Endpoint	Empty String	SSID Physical Address
Ifadminstatus	Per [RFC 2863]	Per [RFC 2863]	Per [RFC 2863]	Per [RFC 2863] Implemented As Read-Only	Per [RFC 2863] Implemented As Read-Only	Per [RFC 2863] Implemented As Read-Only	Per [RFC 2863]	Per [RFC 2863]
Ifoperstatus	Per [RFC 2863]	Per [RFC 2863]	Per [RFC 2863]	Per [RFC 2863]	Per [RFC 2863]	Per [RFC 2863]	Per [RFC 2863]	Per [RFC 2863]
Iflastchange	Unspecified	Unspecified	Unspecified	Unspecified	Unspecified	Unspecified	Unspecified	Unspecified
Ifinoctets	0	0	0	0	0	0	0	0
Ifinnucastpkts	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated
Ifindiscards	0	0	0	0	0	0	0	0
Ifinerrors	0	0	0	0	0	0	0	0
Ifunknownprotos	0	0	0	0	0	0	0	0
Ifoutoctets	0	0	0	0	0	0	0	0
Ifoutucastpkts	0	0	0	0	0	0	0	0
Ifoutnucastpkts	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated
Ifoutdiscards	0	0	0	0	0	0	0	0
Ifouterrors	0	0	0	0	0	0	0	0
Ifoutqlen	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated
Ifspecific	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated	Deprecated

[66] Section added per eRouter-N-14-121703, table updated per eRouter-N-15.234-2 on 2/23/15 by PO.

Additionally, for all interfaces supported the eRouter MUST use the values contained in Table A-3for the referenced objects in the ifTable. This includes modifications to the MAX-ACCESS for specific objects, which differs from [RFC 2863] in some cases.

Table A-3 - eRouter ifTable Row Entries for Supported Interfaces

ifTable Object	Ethernet	USB	MoCA	CFI IP	OFI IP	GRE	WiFi	SSID
IfIndex	5 - 15	30 - 39	40 - 47	200 - 299	300 - 399	400 - 499	Per Annex A.1	Per Annex A.1
ifDescr	unspecified	unspecified	unspecified	unspecified	unspecified	unspecified	WiFi radio interface	WiFi SSID sub-interface
IfType	ethernetCsmacd	usb	moca	ipForward	ipForward	tunnel	ieee80211(71)	ieee80211(71)
IfMtu	0	0	0	0	0	0	0	0
IfSpeed	0	0	0	0	0	0	0	0
ifPhysAddress	Ethernet physical address	USB physical address	MoCA physical address	CFI IP MAC address	OFI IP MAC address	MAC associated with tunnel endpoint	Empty string	SSID physical address
IfAdminStatus	Per [RFC 2863]	Per [RFC 2863]	Per [RFC 2863]	Per [RFC 2863] implemented as read-only	Per [RFC 2863] implemented as read-only	Per [RFC 2863] implemented as read-only	Per [RFC 2863]	Per [RFC 2863]
ifOperStatus	Per [RFC 2863]	Per [RFC 2863]	Per[RFC 2863]	Per [RFC 2863]	Per [RFC 2863]	Per [RFC 2863]	Per [RFC 2863]	Per [RFC 2863]
IfLastChange	unspecified	unspecified	unspecified	unspecified	unspecified	unspecified	unspecified	unspecified
ifInOctets	0	0	0	0	0	0	0	0
IfInNUCastPkts	deprecated	deprecated	deprecated	deprecated	deprecated	deprecated	deprecated	deprecated
IfInDiscards	0	0	0	0	0	0	0	0
IfInErrors	0	0	0	0	0	0	0	0
IfUnknownProtos	0	0	0	0	0	0	0	0
ifOutOctets	0	0	0	0	0	0	0	0
ifOutUCastPkts	0	0	0	0	0	0	0	0
IfOutNUCastPkts	deprecated	deprecated	deprecated	deprecated	deprecated	deprecated	deprecated	deprecated
IfOutDiscards	0	0	0	0	0	0	0	0
IfOutErrors	0	0	0	0	0	0	0	0
IfOutQlen	deprecated	deprecated	deprecated	deprecated	deprecated	deprecated	deprecated	deprecated
IfSpecific	deprecated	deprecated	deprecated	deprecated	deprecated	deprecated	deprecated	deprecated

A.3 eRouter ipNetToPhysicalTable Requirements

The eRouter MUST implement the row entry specified inTable A-4 for the ipNetToPhysicalTable as specified in [RFC 4293] .

Table A-4 - eRouter ipNetToPhysicalTable Row Entries

ipNetToPhysicalTable ([RFC 4293])	eRouter device
ipNetToPhysicalIfIndex	1
ipNetToPhysicalNetAddressType	ipv4(1) or ipv6(2)
ipNetToPhysicalNetAddress	eRouter IP Address
ipNetToPhysicalPhysAddress	eRouter MAC Address
ipNetToPhysicalLastUpdated	<refer to [RFC 4293]>
ipNetToPhysicalType	static(4)
ipNetToPhysicalState	<refer to [RFC 4293]>
ipNetToPhysicalRowStatus	‘active’

A.4 CLAB-GRE-MIB[67]

An eRouter that implements GRE tunneling MUST support the following MIB.

```
CLAB-GRE-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY,
    OBJECT-TYPE,
    Counter64,
    Unsigned32,
    Integer32
        FROM SNMPv2-SMI

    MODULE-COMPLIANCE,
    OBJECT-GROUP
        FROM SNMPv2-CONF

    RowStatus,
    TruthValue
        FROM SNMPv2-TC

    SnmpAdminString
        FROM SNMP-FRAMEWORK-MIB

    clabCommonMibs
        FROM CLAB-DEF-MIB;

clabGREMib MODULE-IDENTITY
    LAST-UPDATED "201502040000Z"
    ORGANIZATION "Cable Television Laboratories, Inc."
    CONTACT-INFO
            "Broadband Network Services
             Cable Television Laboratories, Inc.
             858 Coal Creek Circle,
             Louisville, CO 80027, USA
             Phone: +1 303-661-9100
             Email: mibs@cablelabs.com"
    DESCRIPTION
        "This MIB module contains the management objects for
```

[67] MIB added per eRouter-N-15.1235-3 on 2/23/15 by PO.

```
         multiple GRE tunnels in residential gateway devices."
    REVISION      "201502040000Z"
    DESCRIPTION
        "Initial version of this MIB module."
    ::= { clabCommonMibs 3 }

--
-- Textual Conventions
--

--
-- Object Definitions
--

clabGREObjects OBJECT IDENTIFIER ::= { clabGREMib 1 }

--
-- MIB Objects
--

clabGRETunnelNumberOfEntries OBJECT-TYPE
    SYNTAX      Unsigned32 (0..256)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number of configured entries in the Tunnel table."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.TunnelNumberOfEntries"
    ::= { clabGREObjects 1 }

clabGREFilterNumberOfEntries OBJECT-TYPE
    SYNTAX      Unsigned32 (0..256)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number of configured entries in the Filter table."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.FilterNumberOfEntries"
    ::= { clabGREObjects 2 }

clabGRETunnelTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF ClabGRETunnelEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table allows configuration and monitoring of GRE tunnels."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069, Device.GRE.Tunnel"
    ::= { clabGREObjects 3 }

clabGRETunnelEntry OBJECT-TYPE
    SYNTAX      ClabGRETunnelEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table allows configuration and monitoring of GRE tunnels."
    INDEX { clabGRETunnelIndex }
    ::= { clabGRETunnelTable 1 }

ClabGRETunnelEntry ::= SEQUENCE {
    clabGRETunnelIndex                           Unsigned32,
    clabGRETunnelRowStatus                        RowStatus,
    clabGRETunnelEnable                           TruthValue,
    clabGRETunnelStatus                           INTEGER,
    clabGRETunnelAlias                            SnmpAdminString,
    clabGRETunnelRemoteEndpoints                  OCTET STRING,
    clabGRETunnelKeepAlivePolicy                  INTEGER,
    clabGRETunnelKeepAliveTimeout                 Unsigned32,
    clabGRETunnelKeepAliveThreshold               Unsigned32,
    clabGRETunnelKeepAliveCount                   Unsigned32,
    clabGRETunnelKeepAliveInterval                Unsigned32,
    clabGRETunnelKeepAliveFailureInterval         Unsigned32,
    clabGRETunnelKeepAliveRecoverInterval         Unsigned32,
    clabGRETunnelDeliveryHeaderProtocol           INTEGER,
    clabGRETunnelDefaultDscpMark                  Unsigned32,
    clabGRETunnelConnectedRemoteEndpoint          SnmpAdminString,
    clabGRETunnelInterfaceNumberOfEntries         Unsigned32,
    clabGRETunnelTcpMssClamping                   Unsigned32,
    clabGRETunnelConcentratorServiceName          SnmpAdminString,
    clabGRETunnnelRemoteEndpointConnectivityState OCTET STRING
}

clabGRETunnelIndex OBJECT-TYPE
    SYNTAX      Unsigned32 (400..499)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Table index. These ifIndex values are the GRE tunnel index number."
    ::= { clabGRETunnelEntry 1 }

clabGRETunnelRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The RowStatus interlock for the creation and deletion of a table
         entry."
    ::= { clabGRETunnelEntry 2 }

clabGRETunnelEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Enables or disable this GRE tunnel."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069, Device.GRE.Tunnel.{i}.Enable"
    ::= { clabGRETunnelEntry 3 }

clabGRETunnelStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                    disabled(1),
                    enabled(2),
                    error(3),
                    errorMisconfigured(4),
                    retryWait(5)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The status of this tunnel. The errorMisconfigured value indicates
         that a necessary configuration value is undefined or invalid. The
         error value MAY be used by the CPE to indicate a locally defined
         error condition. retryWait indicates that all concentrators have
         been detected as failed and a FailureInterval is in progress before
         retrying again."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069, Device.GRE.Tunnel.{i}.Status"
    DEFVAL      { disabled }
    ::= { clabGRETunnelEntry 4 }

clabGRETunnelAlias OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "A non-volatile handle used to reference this instance. Alias
         provides a mechanism for an ACS to label this instance for future
         reference."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069, Device.GRE.Tunnel.{i}.Alias"
    ::= { clabGRETunnelEntry 5 }

clabGRETunnelRemoteEndpoints OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE (0..1040))
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Comma-separated list (up to 4 items) of strings (maximum item length
         256). The Fully Qualified Domain Name (FQDN) or IP address of the
         remote GRE tunnel endpoint. If TunnelConcentratorServiceName is set,
         this field is ignored. If more than one name/address is supplied,
         they will be tried in turn, i.e. they are in decreasing order of
         precedence. IPv4 or IPv6 addresses are accepted."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.RemoteEndpoints"
    ::= { clabGRETunnelEntry 6 }

clabGRETunnelKeepAlivePolicy OBJECT-TYPE
    SYNTAX      INTEGER {
                    icmp(1),
                    none(2)
    }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The mechanism used to keep the tunnel from timing out."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.KeepAlivePolicy"
    DEFVAL { none }
    ::= { clabGRETunnelEntry 7 }

clabGRETunnelKeepAliveTimeout OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Time to wait for a response to a keep alive message."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.KeepAliveTimeout"
    DEFVAL { 10 }
    ::= { clabGRETunnelEntry 8 }
```

```
clabGRETunnelKeepAliveThreshold OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "errors"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Number of KeepAlive messages unacknowledged by the
         ConnectedRemoteEndpoint before the Tunnel is perceived failed."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.KeepAliveThreshold"
    DEFVAL { 3 }
    ::= { clabGRETunnelEntry 9 }

clabGRETunnelKeepAliveCount OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "packets"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Number of keep-alive messages sent in a burst at regular intervals."
    DEFVAL { 3 }
    ::= { clabGRETunnelEntry 10 }

clabGRETunnelKeepAliveInterval OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Interval in seconds between keep-alive message bursts."
    DEFVAL { 60 }
    ::= { clabGRETunnelEntry 11 }

clabGRETunnelKeepAliveFailureInterval OBJECT-TYPE
    SYNTAX      Unsigned32 (0..65535)
    UNITS       "seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Time (in seconds) to wait after all available GRE concentrators fail
         to respond, before retrying first GRE concentrator address."
    DEFVAL { 300 }
    ::= { clabGRETunnelEntry 12 }

clabGRETunnelKeepAliveRecoverInterval OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Time (in seconds) to remain on a secondary GRE concentrator, with
         clients connected, before retrying primary GRE concentrator. Zero
         value means no limit. Setting to a small non-zero value will cause
         an immediate switch from a secondary GRE concentrator back to the
         primary."
    DEFVAL { 43200 }
    ::= { clabGRETunnelEntry 13 }

clabGRETunnelDeliveryHeaderProtocol OBJECT-TYPE
    SYNTAX      INTEGER {
                    ipv4(1),
                    ipv6(2)
    }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "If ConcentratorServiceName or RemoteEndpoint FQDNs are provided,
         TunnelDeliveryHeaderProtocol determines whether IPv4 or IPv6
         addresses are requested from DNS."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.DeliveryHeaderProtocol"
    ::= { clabGRETunnelEntry 14 }

clabGRETunnelDefaultDscpMark OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "If the specified entries in the Filter table do not match a packet,
         then the value of this parameter is applied to the delivery header of
         the packet."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.DefaultDSCPMark"
    ::= { clabGRETunnelEntry 15 }

clabGRETunnelConnectedRemoteEndpoint OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The Fully Qualified Domain Name (FQDN) name or IP address of the
         currently connected remote GRE tunnel endpoint."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.ConnectedRemoteEndpoint"
    ::= { clabGRETunnelEntry 16 }

clabGRETunnelInterfaceNumberOfEntries OBJECT-TYPE
    SYNTAX      Unsigned32 (0..256)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number of configured entries in the TunnelInterface table for
         this tunnel."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.InterfaceNumberOfEntries"
    ::= { clabGRETunnelEntry 17 }

clabGRETunnelTcpMssClamping OBJECT-TYPE
    SYNTAX      Unsigned32 (0..1420)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Specifies whether TCP MSS clamping is enabled on the tunnel.
         0 disables clamping, 1 clamps the MSS depending on the
         interface MTU. A value > 1 will be used as clamping size.
              See Reference: RFC 4459"
    DEFVAL { 0 }
    ::= { clabGRETunnelEntry 18 }

clabGRETunnelConcentratorServiceName OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "FQDN of GRE tunnel concentrator/gw service.
         If this is set, then a DNS query of type SRV will be used for
         discovering the FQDN of remote endpoints of GRE tunnel."
    ::= { clabGRETunnelEntry 19 }

clabGRETunnnelRemoteEndpointConnectivityState OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE (0..60))
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Comma-separated list (up to 4 items) of strings. Each item
         corresponds to one item in the RemoteEndpoints list, and contains
         one of the following strings: 'Reachable' indicates that the
         corresponding remote endpoint is responding to any configured
         KeepAlive messages. 'Unreachable' indicates that the remote
         endpoint has failed to adequately respond to the most recent
         KeepAlive attempt. 'NotInUse' indicates that the remote endpoint
         has not been used. 'NoKeepAlives' indicates that the KeepAlive
         policy is set to 'none'."
    ::= { clabGRETunnelEntry 20 }

clabGRETunnelStatsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF ClabGRETunnelStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table provides statistical information of GRE tunnels.
         This table extends clabGRETunnelTable. The index used is
         imported from the the clabGRETunnelTable. If a row is deleted
         from clabGRETunnelTable, then the extension row will also be deleted."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Stats"
    ::= { clabGREObjects 4 }

clabGRETunnelStatsEntry OBJECT-TYPE
    SYNTAX      ClabGRETunnelStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Each table entry provides statistical information on a GRE tunnel."
    AUGMENTS { clabGRETunnelEntry }
    ::= { clabGRETunnelStatsTable 1 }

ClabGRETunnelStatsEntry ::= SEQUENCE {
    clabGRETunnelStatsKeepAliveSent            Counter64,
    clabGRETunnelStatsKeepAliveReceived        Counter64,
    clabGRETunnelStatsBytesSent                Counter64,
    clabGRETunnelStatsBytesReceived            Counter64,
    clabGRETunnelStatsPacketsSent              Counter64,
    clabGRETunnelStatsPacketsReceived          Counter64,
    clabGRETunnelStatsErrorsSent               Counter64,
    clabGRETunnelStatsErrorsReceived           Counter64
}

clabGRETunnelStatsKeepAliveSent OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number keepalive packets sent."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Stats.KeepAliveSent"
    ::= { clabGRETunnelStatsEntry 1 }

clabGRETunnelStatsKeepAliveReceived OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number keepalive packets received."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Stats.KeepAliveReceived"
    ::= { clabGRETunnelStatsEntry 2 }

clabGRETunnelStatsBytesSent OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "bytes"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number bytes sent."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Stats.BytesSent"
    ::= { clabGRETunnelStatsEntry 3 }

clabGRETunnelStatsBytesReceived OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "bytes"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number bytes received."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Stats.BytesReceived"
    ::= { clabGRETunnelStatsEntry 4 }

clabGRETunnelStatsPacketsSent OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number packets sent."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Stats.PacketsSent"
    ::= { clabGRETunnelStatsEntry 5 }

clabGRETunnelStatsPacketsReceived OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number packets received."
    REFERENCE
```

```
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Stats.PacketsReceived"
    ::= { clabGRETunnelStatsEntry 6 }

clabGRETunnelStatsErrorsSent OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number packets received that were discarded.
         Because source network interface is down."
    ::= { clabGRETunnelStatsEntry 7 }

clabGRETunnelStatsErrorsReceived OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number packets received that were in error.
         Because tunnel packets are not correct type.
         Eg. Received IP packet instead of L2 packets."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Stats.ErrorsReceived"
    ::= { clabGRETunnelStatsEntry 8 }

clabGRETunnelInterfaceTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF ClabGRETunnelInterfaceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains information specific to network source interface
         whose traffic is mapped into a GRE tunnel."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface"
    ::= { clabGREObjects 5 }

clabGRETunnelInterfaceEntry OBJECT-TYPE
    SYNTAX      ClabGRETunnelInterfaceEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "List of individual source interface entries mapped into GRE tunnels."
    INDEX {
        clabGRETunnelIndex,
        clabGRETunnelInterfaceIndex
    }
    ::= { clabGRETunnelInterfaceTable 1 }

ClabGRETunnelInterfaceEntry ::= SEQUENCE {
    clabGRETunnelInterfaceIndex                         Unsigned32,
    clabGRETunnelInterfaceRowStatus                     RowStatus,
    clabGRETunnelInterfaceEnable                        TruthValue,
    clabGRETunnelInterfaceStatus                        INTEGER,
    clabGRETunnelInterfaceAlias                         SnmpAdminString,
    clabGRETunnelInterfaceName                          SnmpAdminString,
    clabGRETunnelInterfaceLastChange                    Unsigned32,
    clabGRETunnelInterfaceLowerLayers                   SnmpAdminString,
    clabGRETunnelInterfaceProtocolIdOverride            Unsigned32,
    clabGRETunnelInterfaceUseChecksum                   TruthValue,
    clabGRETunnelInterfaceKeyIdentifierGenerationPolicy  INTEGER,
    clabGRETunnelInterfaceKeyIdentifier                  Unsigned32,
    clabGRETunnelInterfaceUseSequenceNumber              TruthValue
}

clabGRETunnelInterfaceIndex    OBJECT-TYPE
    SYNTAX      Unsigned32 (1..256)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The key for a unique instance of this object."
    ::= {clabGRETunnelInterfaceEntry 1 }

clabGRETunnelInterfaceRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The RowStatus interlock for the creation and deletion of a table
         entry."
    ::= { clabGRETunnelInterfaceEntry 2 }

clabGRETunnelInterfaceEnable OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Enable or disable mapping traffic from network interface into GRE
         tunnel."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface.{i}.Enable"
    ::= { clabGRETunnelInterfaceEntry 3 }

clabGRETunnelInterfaceStatus OBJECT-TYPE
    SYNTAX      INTEGER {
                    down(1),
                    up(2),
                    unknown(3),
                    dormant(4),
                    notPresent(5),
                    lowerLayerDown(6),
                    error(7)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The status of this TunnelInterface. When Enable is false then Status
         SHOULD normally be Down (or NotPresent or Error if there is a
         fault condition on the interface).

         When Enable is changed to true then Status SHOULD change to Up
         if and only if the interface is able to transmit and receive
         network traffic; it SHOULD change to Dormant if and only if the
         interface is operable but is waiting for external actions before
         it can transmit and receive network traffic (and subsequently change
         to Up if still operable when the expected actions have completed);
         it SHOULD change to LowerLayerDown if and only if the interface is
         prevented from entering the Up state because one or more of the
         interfaces beneath it is down; it SHOULD remain in the Error state
         if there is an error or other fault condition detected on the
         interface; it SHOULD remain in the NotPresent state if the interface
         has missing (typically hardware) components; it SHOULD change to
         Unknown if the state of the interface can not be determined for some
         reason."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface.{i}.Status"
    DEFVAL { down }
    ::= { clabGRETunnelInterfaceEntry 4 }

clabGRETunnelInterfaceAlias OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "A non-volatile handle used to reference this instance. Alias
         provides a mechanism for an ACS to label this instance for future
         reference."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface.{i}.Alias"
    ::= { clabGRETunnelInterfaceEntry 5 }

clabGRETunnelInterfaceName OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "GRE tunnel interface name (The textual name of the tunnel interface
         as assigned by the CPE)"
    ::= { clabGRETunnelInterfaceEntry 6 }

clabGRETunnelInterfaceLastChange OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS       "seconds"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The accumulated time in seconds since the tunnel interface entered
         its current operational state."
    ::= { clabGRETunnelInterfaceEntry 7 }

clabGRETunnelInterfaceLowerLayers OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Comma-separated list of source ifIndex values whose traffic will be
         passed through the GRE tunnel associated with this TunnelInterface.
         Inferface numbers are defined according to Annex A of the eRouter
         specification."

    ::= { clabGRETunnelInterfaceEntry 8 }

clabGRETunnelInterfaceProtocolIdOverride OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The protocol ID assigned to this TunnelInterface. The Protocol ID
         is the EtherType value used to identify the encapsulated payload.
         When set to 0 the CPE determines the Protocol Id."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface.{i}.ProtocolIdOverride"
    ::= { clabGRETunnelInterfaceEntry 9 }

clabGRETunnelInterfaceUseChecksum OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Enable or disable optional checksum in outgoing GRE packets."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface.{i}.UseChecksum"
    DEFVAL { false }
    ::= { clabGRETunnelInterfaceEntry 10 }

clabGRETunnelInterfaceKeyIdentifierGenerationPolicy OBJECT-TYPE
    SYNTAX      INTEGER {
                    disabled(1),
                    provisioned(2),
                    generated(3)
    }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Control over optional Key field in GRE header of outgoing packets.
         Disabled means key is not present, provisioned uses the value
         provided in the key field, generated means CPE will generate key."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface.{i}.KeyIdentifierGenerationPolicy"
    DEFVAL { disabled }
    ::= { clabGRETunnelInterfaceEntry 11 }

clabGRETunnelInterfaceKeyIdentifier OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "When KeyIdentifierGenerationPolicy is provisioned, this is the
         value for the Key header."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface.{i}.KeyIdentifier"
    ::= { clabGRETunnelInterfaceEntry 12 }

clabGRETunnelInterfaceUseSequenceNumber OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Enable or disable optional sequence number field in outgoing GRE
         packets."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface.{i}.UseSequenceNumber"
    DEFVAL { false }
    ::= { clabGRETunnelInterfaceEntry 13 }

clabGRETunnelInterfaceStatsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF ClabGRETunnelInterfaceStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table provides statistical information of GRE tunnel interfaces.
```

```
         This table extends clabGRETunnelInterfaceTable. The index used is
         imported from the the clabGRETunnelInterfaceTable. If a row is
         deleted from clabGRETunnelInterfaceTable, then the extension row
         will also be deleted."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface.{i}.Stats"
    ::= { clabGREObjects 6 }

clabGRETunnelInterfaceStatsEntry OBJECT-TYPE
    SYNTAX      ClabGRETunnelInterfaceStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Each table entry provides statistical information on a GRE tunnel."
    AUGMENTS { clabGRETunnelInterfaceEntry }
    ::= { clabGRETunnelInterfaceStatsTable 1 }

ClabGRETunnelInterfaceStatsEntry ::= SEQUENCE {
    clabGRETunnelInterfaceStatsBytesSent                        Counter64,
    clabGRETunnelInterfaceStatsBytesReceived                    Counter64,
    clabGRETunnelInterfaceStatsPacketsSent                      Counter64,
    clabGRETunnelInterfaceStatsPacketsReceived                  Counter64,
    clabGRETunnelInterfaceStatsErrorsSent                       Counter64,
    clabGRETunnelInterfaceStatsErrorsReceived                   Counter64,
    clabGRETunnelInterfaceStatsDiscardChecksumReceived          Counter64,
    clabGRETunnelInterfaceStatsDiscardSequenceNumberReceived    Counter64
}

clabGRETunnelInterfaceStatsBytesSent OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "bytes"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number bytes sent."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface.{i}.Stats.BytesSent"
    ::= { clabGRETunnelInterfaceStatsEntry 1 }

clabGRETunnelInterfaceStatsBytesReceived OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "bytes"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number bytes received."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface.{i}.Stats.BytesReceived"
    ::= { clabGRETunnelInterfaceStatsEntry 2 }

clabGRETunnelInterfaceStatsPacketsSent OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number packets sent."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface.{i}.Stats.PacketsSent"
    ::= { clabGRETunnelInterfaceStatsEntry 3 }

clabGRETunnelInterfaceStatsPacketsReceived OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number packets received."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface.{i}.Stats.PacketsReceived"
    ::= { clabGRETunnelInterfaceStatsEntry 4 }

clabGRETunnelInterfaceStatsErrorsSent OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number packets received that were discarded.
         Because source network interface is down."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface.{i}.Stats.ErrorsSent"
    ::= { clabGRETunnelInterfaceStatsEntry 5 }

clabGRETunnelInterfaceStatsErrorsReceived OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number packets received that were in error.
         Because tunnel packets are not correct type.
         Eg. Received IP packet instead of L2 packets."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface.{i}.Stats.ErrorsReceived"
    ::= { clabGRETunnelInterfaceStatsEntry 6 }

clabGRETunnelInterfaceStatsDiscardChecksumReceived OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total number of packets received by the tunnel interface that
         are discarded because of Checksum errors."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface.{i}.Stats.DiscardChecksumReceived"
    ::= { clabGRETunnelInterfaceStatsEntry 7 }

clabGRETunnelInterfaceStatsDiscardSequenceNumberReceived OBJECT-TYPE
    SYNTAX      Counter64
    UNITS       "packets"
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total number of packets received by the tunnel interface that
         are discarded because of Sequence Number errors."
    REFERENCE
```

```
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Tunnel.{i}.Interface.{i}.Stats
         .DiscardSequenceNumberReceived"
    ::= { clabGRETunnelInterfaceStatsEntry 8 }

clabGREFilterTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF ClabGREFilterEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains settings to allow packets passing through
         any tunnel, a particular tunnel, or a particular tunnel interface
         to have defined header modifications made.

         Entries in this table are applied against the packets that are to be
         encapsulated within the GRE Tunnel based on the following conditions.

         Filter table entries are in priority order, with the lowest Order
         values being the highest priority. Only the first enabled matching
         entry is applied to each packet.

         * If the value of FilterAllInterfaces is true, then this entry
           applies to all Tunnel and TunnelInterface instances.
         * Otherwise, this filter applies to the Tunnel or TunnelInterface
           referenced by FilterInterface.
         * For enabled table entries, if FilterInterface is not valid,
           then the table entry is inoperable and the CPE MUST set Status to
           errorMisconfigured.

         For each packet, enabled entries are checked from lowest to highest
         Order, and the first table entry that matches according to the
         AllInterfaces, Interface, VlanIdCheck, and VlanIdExclude criteria
         is applied to the packet. All other Filter entries are ignored."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069, Device.GRE.Filter"
    ::= { clabGREObjects 7 }

clabGREFilterEntry OBJECT-TYPE
    SYNTAX      ClabGREFilterEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "List of individual filter entries."
    INDEX { clabGREFilterIndex }
    ::= { clabGREFilterTable 1 }

ClabGREFilterEntry ::= SEQUENCE {
    clabGREFilterIndex              Unsigned32,
    clabGREFilterRowStatus          RowStatus,
    clabGREFilterEnable             TruthValue,
    clabGREFilterStatus             INTEGER,
    clabGREFilterOrder              Unsigned32,
    clabGREFilterAlias              SnmpAdminString,
    clabGREFilterInterface          SnmpAdminString,
    clabGREFilterAllInterfaces      TruthValue,
    clabGREFilterVlanIdCheck        Integer32,
    clabGREFilterVlanIdExclude      TruthValue,
    clabGREFilterDscpMarkPolicy     Integer32,
    clabGREFilterVlanTag            Unsigned32
}

clabGREFilterIndex    OBJECT-TYPE
    SYNTAX      Unsigned32 (1..256)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The key for a unique instance of this object."
    ::= { clabGREFilterEntry 1 }

clabGREFilterRowStatus OBJECT-TYPE
    SYNTAX      RowStatus
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The RowStatus interlock for the creation and deletion of a table
         entry."
    ::= { clabGREFilterEntry 2 }

clabGREFilterEnable    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Enable or disable this filter to modify packets."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069, Device.GRE.Filter.{i}.Enable"
    DEFVAL { false }
    ::= { clabGREFilterEntry 3 }

clabGREFilterStatus    OBJECT-TYPE
    SYNTAX      INTEGER {
                    disabled(1),
                    enabled(2),
                    errorMisconfigured(3),
                    error(4)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The status of this filter. The errorMisconfigured value indicates
         that a necessary configuration value is undefined or invalid. The
         error value MAY be used by the CPE to indicate a locally defined
         error condition."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069, Device.GRE.Filter.{i}.Status"
    DEFVAL { disabled }
    ::= { clabGREFilterEntry 4 }

clabGREFilterOrder OBJECT-TYPE
    SYNTAX      Unsigned32 (1..256)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Position of the Filter entry in the order of precedence. A value
         of 1 indicates the first entry considered (highest precedence).
         For each packet, the highest ordered entry that matches the filter
         criteria is applied. All lower order entries are ignored.

         When this value is modified, if the value matches that of an
         existing entry, the Order value for the existing entry and all
         lower Order entries is incremented (lowered in precedence) to
         ensure uniqueness of this value. A deletion causes Order values to
         be compacted. When a value is changed, incrementing occurs before
         compaction.

         The value of Order on creation of a Filter table entry MUST be one
```

```
          greater than the largest current value (initially assigned the
          lowest precedence)."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069, Device.GRE.Filter.{i}.Order"
    ::= { clabGREFilterEntry 5 }

clabGREFilterAlias OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "A non-volatile handle used to reference this instance. Alias
         provides a mechanism for an ACS to label this instance for future
         reference."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069, Device.GRE.Filter.{i}.Alias"
    ::= { clabGREFilterEntry 6 }

clabGREFilterInterface      OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Filter criterion. This specifies the path name of a row in the
         Tunnel or TunnelInterface table to which the filter is applied."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Filter.{i}.Interface"
    ::= { clabGREFilterEntry 7 }

clabGREFilterAllInterfaces  OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Filter criterion. If false, this filter only applies to the Tunnel
         or TunnelInterface referred to by FilterInterface. If true, this
         filter applies to all TunnelInterfaces for all Tunnels."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Filter.{i}.AllInterfaces"
    ::= { clabGREFilterEntry 8 }

clabGREFilterVlanIdCheck    OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Filter criterion. Current outer Ethernet VLAN ID as defined in
         802.1Q. A value of -1 indicates this criterion is not used for
         filtering."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Filter.{i}.VLANIDCheck"
    DEFVAL      { -1 }
    ::= { clabGREFilterEntry 9 }

clabGREFilterVlanIdExclude    OBJECT-TYPE
    SYNTAX      TruthValue
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "If false, the filter includes only those packets that match the
```

```
         VlanIdCheck entry, if specified. If true, the filter includes all
         packets except those that match the VlanIdCheck entry, if specified."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Filter.{i}.VLANIDExclude"
    DEFVAL { false }
    ::= { clabGREFilterEntry 10 }

clabGREFilterDscpMarkPolicy OBJECT-TYPE
    SYNTAX      Integer32 (-2..63)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The method is used to set the 6 bits of the DSCP field in the
         IPv4 header or traffic class in IPv6 header of the GRE packet.
         A value of -1 indicates copy from the upstream packet. A value
         of -2 indicates automatic marking of DSCP."
    REFERENCE
        "TR-181 Device:2.8 Data Model for TR-069,
         Device.GRE.Filter.{i}.DSCPMarkPolicy"
    DEFVAL { 0 }
    ::= { clabGREFilterEntry 11 }

clabGREFilterVlanTag OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Add VLAN tag consists of 16bit of Tag Protocol Identifier (TPID)
         and 16 bit of Tag Control Identifier (TCI).
         Value of 0 will disable adding VLAN tag.
         Tag Protocol Identifier (TPID) should set to 0x8100."
    DEFVAL { 0 }
    ::= { clabGREFilterEntry 12 }

--
-- Conformance definitions
--
clabGREMibConformance  OBJECT IDENTIFIER ::= { clabGREMib 2 }

--
-- Compliance Statements
--
clabGREMibCompliances  OBJECT IDENTIFIER ::= { clabGREMibConformance 1 }
clabGREMibGroups       OBJECT IDENTIFIER ::= { clabGREMibConformance 2 }

clabGRECompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "The compliance statement for entities which implement the GRE MIB."
    MODULE -- this module
    MANDATORY-GROUPS { clabGREGroup }
    ::= { clabGREMibCompliances 1 }

clabGREGroup OBJECT-GROUP
    OBJECTS {
        clabGRETunnelNumberOfEntries,
        clabGREFilterNumberOfEntries,
        clabGRETunnelRowStatus,
        clabGRETunnelEnable,
        clabGRETunnelStatus,
        clabGRETunnelAlias,
        clabGRETunnelRemoteEndpoints,
        clabGRETunnelKeepAlivePolicy,
        clabGRETunnelKeepAliveTimeout,
        clabGRETunnelKeepAliveThreshold,
        clabGRETunnelKeepAliveCount,
        clabGRETunnelKeepAliveInterval,
        clabGRETunnelKeepAliveFailureInterval,
        clabGRETunnelKeepAliveRecoverInterval,
        clabGRETunnelDeliveryHeaderProtocol,
        clabGRETunnelDefaultDscpMark,
        clabGRETunnelConnectedRemoteEndpoint,
        clabGRETunnelInterfaceNumberOfEntries,
        clabGRETunnelTcpMssClamping,
        clabGRETunnelConcentratorServiceName,
        clabGRETunnnelRemoteEndpointConnectivityState,
        clabGRETunnelStatsKeepAliveSent,
        clabGRETunnelStatsKeepAliveReceived,
        clabGRETunnelStatsBytesSent,
        clabGRETunnelStatsBytesReceived,
        clabGRETunnelStatsPacketsSent,
        clabGRETunnelStatsPacketsReceived,
        clabGRETunnelStatsErrorsSent,
        clabGRETunnelStatsErrorsReceived,
        clabGRETunnelInterfaceRowStatus,
        clabGRETunnelInterfaceStatus,
        clabGRETunnelInterfaceAlias,
        clabGRETunnelInterfaceName,
        clabGRETunnelInterfaceLastChange,
        clabGRETunnelInterfaceLowerLayers,
        clabGRETunnelInterfaceEnable,
        clabGRETunnelInterfaceProtocolIdOverride,
        clabGRETunnelInterfaceUseChecksum,
        clabGRETunnelInterfaceKeyIdentifierGenerationPolicy,
        clabGRETunnelInterfaceKeyIdentifier,
        clabGRETunnelInterfaceUseSequenceNumber,
        clabGRETunnelInterfaceStatsBytesSent,
        clabGRETunnelInterfaceStatsBytesReceived,
        clabGRETunnelInterfaceStatsPacketsSent,
        clabGRETunnelInterfaceStatsPacketsReceived,
        clabGRETunnelInterfaceStatsErrorsSent,
        clabGRETunnelInterfaceStatsErrorsReceived,
        clabGRETunnelInterfaceStatsDiscardChecksumReceived,
        clabGRETunnelInterfaceStatsDiscardSequenceNumberReceived,
        clabGREFilterRowStatus,
        clabGREFilterEnable,
        clabGREFilterStatus,
        clabGREFilterOrder,
        clabGREFilterAlias,
        clabGREFilterInterface,
        clabGREFilterAllInterfaces,
        clabGREFilterVlanIdCheck,
        clabGREFilterVlanIdExclude,
        clabGREFilterVlanTag,
        clabGREFilterDscpMarkPolicy
    }
    STATUS current
    DESCRIPTION
        "A collection of objects providing instrumentation and control over
         GRE tunnels."
    ::= { clabGREMibGroups 1 }

END
```

Annex B Configuration of eRouter Operational Parameters

This Annex defines the configuration TLVs used by the eRouter and describes how the configuration parameters are transferred from the eCM to the eRouter.

B.1 eRouter SNMP Configuration

This Annex subsection defines the configuration of SNMP access to the eRouter.

B.1.1 eRouter SNMP Modes of Operation

The eRouter MUST support SNMPv1, SNMPv2c, in SNMP-coexistence mode as defined in [RFC 3584]. The eRouter MAY support SNMPv3 as defined in [OSSIv3.0].

B.1.2 eRouter SNMP Access Control Configuration

The eRouter uses the View-based Access Control Model (VACM) for configuration of SNMPv1v2c co-existence as defined in [RFC 3584].

B.1.2.1 View-based Access Control Model (VACM) Profile

This section addresses the default VACM profile for the eRouter.

The eRouter MUST support a pre-installed entry in the vacmViewTreeFamilyTable [RFC 3415] as follows:

Table B-1 - vacmViewTreeFamilyTable

Column Name (* = Part of Index)	Column Value
* vacmViewTreeFamilyViewName	eRouterManagerView
* vacmViewTreeFamilySubtree	<1.3.6.1>
vacmViewTreeFamilyMask	Zero-length String
vacmViewTreeFamilyType	'included'
vacmViewTreeFamilyStorageType	volatile (2) or nonvolatile (3)
vacmViewTreeFamilyStatus	active (1)

The eRouter MAY also support additional views to be configured by the operator during the provisioning process, as defined in the SNMPv1v2c Access View Name encoding Annex B.4.5.4 and the SNMPv3 Access View Configuration encoding.

B.1.3 SNMPv1v2c Coexistence Configuration

This section specifies eRouter handling of the SNMPv1v2c Coexistence Configuration encodings as defined in Annex B.4.3.1 when included in the eRouter configuration information. The SNMPv1v2c Coexistence Configuration encoding is used to configure SNMPv3 framework tables for SNMPv1 and v2c access.

The eRouter uses the SNMPv1v2c Coexistence Configuration encodings to create entries in the following tables:

- snmpCommunityTable;
- snmpTargetAddrTable;
- vacmSecurityToGroupTable;
- vacmAccessTable;
- snmpTargetAddrExtTable.

B.1.3.1 Mapping SNMPv1v2c Coexistence Configuration

This section describes the mapping of SNMPv1v2c Coexistence Configuration into SNMPv3 entries.

Table B-2 provides a Variable Name as a short-hand reference to be used in the SNMPv3 tables defined in subsections below for each of the SNMPv1v2c Coexistence Configuration encodings. The table also defines the mapping between each of the SNMPv1v2c Coexistence Configuration encodings and the associated SNMP MIB objects.

Table B-2 - SNMPv1v2c Coexistence Configuration Mapping

Encodings	Variable Name	Associated MIB Object
SNMPv1v2c Community Name	CommunityName	snmpCommunityName [RFC 3584]
SNMPv1v2c Transport Address Access		
SNMPv1v2c Transport Address	TAddress	snmpTargetAddrTAddress [RFC 3413]
SNMPv1v2c Transport Address Mask	TMask	snmpTargetAddrTMask [RFC 3584]
SNMPv1v2c Access View Type	AccessViewType	
SNMPv1v2c Access View Name (optional, see Section B.4.5.4)	AccessViewName or eRouterManagerView	vacmAccessReadViewName and vacmAccessWriteViewName [RFC 3415]

The eRouter is not required to verify the consistency across tables.

Table B-3 through Table B-7 describe the eRouter procedures to populate the SNMPv3 framework tables to conform to the "SNMP Management Framework Message Processing and Access Control Subsystems" [RFC 3412].

When configuring entries in these SNMPv3 tables:

- The ReadViewName and WriteViewName may correspond to default entries as defined in Annex B.1.3.1 or entries created using SNMPv3 Access View Configuration (see Annex B.4.6).
- Multiple columnar objects can be configured with indexes containing the string "@eRouterRouterconfig". If these tables are configured through other mechanisms, network operators should not use values beginning with "@eRouterconfig", to avoid conflicts.

B.1.3.1.1 snmpCommunityTable

The snmpCommunityTable is defined in the "SNMP Community MIB Module" section of [RFC 3584].

The eRouter MUST create one row in snmpCommunityTable for each SNMPv1v2c Coexistence Configuration TLV as follows:

- The eRouter sets the value of snmpCommunityIndex to "@eRouterconfig_n" where 'n' is a sequential number starting at 0 for each TLV processed (e.g., "@eRouterconfig_0", "@eRouterconfig_1", etc.)
- The eRouter creates space separated tags in snmpCommunityTransportTag for each SNMPv1v2c Community Name sub-TLV of the SNMPv1v2c Coexistence Configuration encoding.

Table B-3 - snmpCommunityTable

Column Name (* = Part of Index)	Column Value
* snmpCommunityIndex	"@eRouterconfig_n" where n is 0..m-1 and m is the number of SNMPv1v2c Community Name TLVs
snmpCommunityName	<CommunityName>
snmpCommunitySecurityName	"@eRouterconfig_n"
snmpCommunityContextEngineID	<the engineID populated by the SNMP>
snmpCommunityContextName	<Zero-length OCTET STRING>
snmpCommunityTransportTag	"@eRouterconfigTag_n" where n is 0..m-1 and m is the number of SNMPv1v2c Coexistence Configuration TLVs
snmpCommunityStorageType	volatile (2)
snmpCommunityStatus	active (1)

B.1.3.1.2 snmpTargetAddrTable

The snmpTargetAddrTable is defined in the "Definitions" section of [RFC 3413].

The eRouter MUST create one row in snmpTargetAddrTable for each SNMPv1v2c Transport Address Access sub-TLV of the SNMPv1v2c Coexistence Configuration encoding.

Table B-4 - snmpTargetAddrTable

Column Name (* = Part of Index)	**Column Value**
* snmpTargetAddrName	"@eRouterconfigTag_n_i" where 'n' is 0..m-1 and 'm' is the number of SNMPv1v2c Coexistence Configuration TLVs. Where 'I' is 0..p-1 and p is the number of SNMPv1v2c Transport Address Access sub-TLV within the SNMPv1v2c Coexistence Configuration TLV n
snmpTargetAddrTDomain	IPv4: snmpUDPDomain [RFC 3417] IPv6: transportDomainUdpIpv6 [RFC 3419]
snmpTargetAddrTAddress (IP Address and UDP Port)	IPv4: SnmpUDPAddress [RFC 3417] OCTET STRING (6) Octets 1-4: <*TAddress*> Octets 5-6: <*TAddress*> IPv6: TransportAddressIPv6 [RFC 3419] OCTET STRING (18) Octets 1-16: <*TAddress*> Octets 17-18: <*TAddress*>
snmpTargetAddrTimeout	Default from MIB
snmpTargetAddrRetryCount	Default from MIB
snmpTargetAddrTagList	"@eRouterconfigTag_n" where n is 0..m-1 and m is the number of SNMPv1v2c Coexistence Configuration TLVs
snmpTargetAddrParams	'00'h (null character)
snmpTargetAddrStorageType	volatile (2)
snmpTargetAddrRowStatus	active (1)

B.1.3.1.3 snmpTargetAddrExtTable

The snmpTargetAddrExtTable is defined in the "SNMP Community MIB Module" section of [RFC 3584].

The eRouter MUST create one row in snmpTargetAddrExtTable for each SNMPv1v2c Transport Address Access sub-TLV of the SNMPv1v2c Coexistence Configuration encoding.

Table B-5 - snmpTargetAddrExtTable

Column Name (* = Part of Index)	**Column Value**
* snmpTargetAddrName	"@eRouterconfigTag_n_i" where 'n' is 0..m-1 and 'm' is the number of SNMPv1v2c Coexistence Configuration TLVs. Where 'i' is 0..p-1 and p is the number of SNMPv1v2c Transport Address Access sub-TLV within the SNMPv1v2c Coexistence Configuration TLV n
snmpTargetAddrTMask	<Zero-length OCTET STRING> when <*TMask*> is not provided in the i-th SNMPv1v2c Transport Address Access sub-TLV IPv4: SnmpUDPAddress [RFC 3417] OCTET STRING (6) Octets 1-4: <*TMask*> Octets 5-6: <UDP Port> IPv6: TransportAddressIPv6 [RFC 3419] OCTET STRING (18) Octets 1-16: <*TMask*> Octets 17-18: <UDP Port>
snmpTargetAddrMMS	Maximum Message Size

B.1.3.1.4 vacmSecurityToGroupTable

The vacmSecurityToGroupTable is defined in the "Definitions" section of [RFC 3415].

The eRouter MUST create two rows in vacmSecurityGroupTable for each SNMPv1v2c Coexistence Configuration TLV as follows:

- The eRouter sets the value of vacmSecurityName to "@eRouterconfig_n" where 'n' is a sequential number starting at 0 for each SNMPv1v2c Coexistence Configuration TLV processed (e.g., "@eRouterconfig_0", "@eRouterconfig_1", etc.);
- The eRouter sets the value of vacmGroupName to "@eRouterconfigV1_n" for the first row and "@eRouterconfigV2_n" for the second row where 'n' is a sequential number starting at 0 for each SNMPv1v2c Coexistence Configuration TLV processed (e.g., "@eRouterconfigV1_0", "@eRouterconfigV1_1", etc.).

Table B-6 - vacmSecurityToGroupTable

Column Name **(* = Part of Index)**	**First Row Column Value**	**Second Row Column Value**
* vacmSecurityModel	SNMPV1 (1)	SNMPV2c (2)
* vacmSecurityName	"@eRouterconfig_n"	"@eRouterconfig_n"
vacmGroupName	"@eRouterconfigV1_n"	"@eRouterconfigV2_n"
vacmSecurityToGroupStorageType	volatile (2)	volatile (2)
vacmSecurityToGroupStatus	active (1)	active (1)

B.1.3.1.5 vacmAccessTable

The vacmAccessTable is defined in the "Definitions" section of [RFC 3415].

The eRouter MUST create two rows in vacmAccessTable for each SNMPv1v2c Coexistence Configuration encoding as follows:

- The eRouter sets the value of vacmGroupName to "@eRouterconfigV1_n" for the first row and "@eRouterconfigV2_n" for the second row where 'n' is a sequential number starting at 0 for each SNMPv1v2c Coexistence Configuration encoding processed (e.g., "@eRouterconfigV1_0", "@eRouterconfigV1_1", etc.);

  In case the eRouter does not support the SNMPv3 Access View Name encoding in Annex B.4, the eRouter MUST use the default view defined in Annex B.1.2.1 and ignore the Sub-TLV SNMPv1v2c Access View Name.

Table B-7 - vacmAccessTable

Column Name **(* = Part of Index)**	**Column Value**	**Column Value**
* vacmGroupName	"@eRouterconfigV1_n"	"@eRouterconfigV2_n"
* vacmAccessContextPrefix	<zero-length string>	<zero-length string>
* vacmAccessSecurityModel	SNMPV1 (1)	SNMPV2c (2)
* vacmAccessSecurityLevel	noAuthNoPriv (1)	noAuthNoPriv (1)
vacmAccessContextMatch	exact (1)	exact (1)
vacmAccessReadViewName	Set < AccessViewName> or eRouterManagerView	Set < AccessViewName> or eRouterManagerView
vacmAccessWriteViewName	When <*AccessViewType*> == '2' Set < AccessViewName> or eRouterManagerView When <*AccessViewType*> != '2' Set <Zero-length OCTET STRING>	When <*AccessViewType*> == '2' Set < AccessViewName or eRouterManagerView When <*AccessViewType*> != '2' Set <Zero-length OCTET STRING>
vacmAccessNotifyViewName	<Zero-length OCTET STRING>	<Zero-length OCTET STRING>
vacmAccessStorageType	volatile (2)	volatile (2)
vacmAccessStatus	active (1)	active (1)

B.1.3.2 *Mapping SNMPv3 Access View Configuration*

If SNMPv3 is supported by the eRouter, the SNMPv3 Access View Configuration encoding is used to configure the vacmViewTreeFamilyTable.

Table B-8 provides a Variable Name as a short-hand reference to be used in the SNMPv3 tables defined in the subsections below for each of the SNMPv3 Access View Configuration encodings. The table also defines the mapping between each of the SNMPv3 Coexistence Configuration encodings and the associated SNMP MIB objects.

Table B-8 - SNMPv3 Access View Configuration Encoding

Encodings	Variable Name	Associated MIB Object [RFC 3415]
SNMPv3 Access View Name	AccessViewName	vacmViewTreeFamilyViewName
SNMPv3 Access View Subtree	AccessViewSubTree	vacmViewTreeFamilySubtree
SNMPv3 Access View Mask	AccessViewMask	vacmViewTreeFamilyMask
SNMPv3 Access View Type	AccessViewType	vacmViewTreeFamilyType

The eRouter is not required to verify the consistency across tables.

Table B-9 describes the eRouter procedures to populate the vacmViewTreeFamilyTable to conform to the "SNMP Management Framework Message Processing and Access Control Subsystems" [RFC 3412].

When configuring entries in these SNMPv3 tables:

- One entry is created for each SNMPv3 Access View Configuration encoding. Some Access Views may have a number of included/excluded OID branches. Only Access View Name will be common for all these OID branches. To support such type of Access View, multiple SNMPv3 Access View Configuration encodings need to be defined.

B.1.3.2.1 *vacmViewTreeFamilyTable*

The vacmViewTreeFamilyTable is defined in the "Definitions" section of [RFC 3415].

If the SNMPv3 Access View Configuration encoding is supported by the eRouter, then the eRouter MUST:

- Create one row in vacmViewTreeFamilyTable for each SNMPv3 Access View Configuration TLV;
- Reject the configuration if two or more SNMPv3 Access View Configuration encodings have identical index components (*AccessViewName* and *AccessViewSubTree*);
- Set the object vacmViewTreeFamilySubtree to 1.3.6 when no sub-TLV SNMPv3 Access View Subtree is defined;
- Set the object vacmViewTreeFamilyMask to the default zero-length string when no sub-TLV SNMPv3 Access View Mask is defined;
- Set the object vacmViewTreeFamilyType to the default value 1 (included) when no sub-TLV SNMPv3 Access View Type is defined.

Table B-9 - vacmViewTreeFamilyTable

Column Name (* = Part of Index)	Column Value
* vacmViewTreeFamilyViewName	<AccessViewName>
* vacmViewTreeFamilySubtree	<AccessViewSubTree>
vacmViewTreeFamilyMask	<AccessViewMask>
vacmViewTreeFamilyType	<AccessViewType>
vacmViewTreeFamilyStorageType	volatile (2)
vacmViewTreeFamilyStatus	active (1)

B.2 SNMP Configuration of eRouter[68]

The esafeErouterInitModeControl object is defined in the "eSAFE MIB Definition" section of [eDOCSIS].

This object provides a means of changing the IP Protocol Enabled Mode of the DOCSIS eRouter. The eRouter only evaluates this object when it is modified via an SNMP SET initiated from an SNMP management station after the eRouter is initialized. The eRouter MUST ignore the esafeErouterInitModeControl whenever it is included in TLV202.11 in the CM configuration file.

The value of this object MUST persist across cable modem reinitialization. The eRouter MUST NOT require a reset when the eRouter Initialization mode is changed via this object from 'IPv4 Protocol Enabled' mode to 'Dual IP Protocol Enabled' mode. The eRouter MUST NOT require a reset when the eRouter Initialization mode is changed via this object from 'IPv6 Protocol Enabled' mode to 'Dual IP Protocol Enabled' mode.

The esafeErouterInitModeControl object MUST be accessible via the eCM SNMP agent through the eCM management address.

The possible values for this object are listed in Table B-10.

Table B-10 - esafeErouterInitModeControl

Value	Description
ipDisabled(1)	When this object is set to ipDisabled(1), the eRouter MUST switch to Disabled Mode.
ipv4Only(2)	When this object is set to ipv4Only(2), the eRouter MUST switch to IPv4 Protocol Enabled Mode.
ipv6Only(3)	When this object is set to ipv6Only(3), the eRouter MUST switch to IPv6 Protocol Enabled Mode.
ipv4AndIpv6(4)	When this object is set to ipv4AndIpv6(4), the eRouter MUST switch to Dual IP Protocol Enabled Mode.
honoreRouterInitMode(5)	When this object is set to honoreRouterInitMode(5), the eRouter MUST honor the eRouter Initialization Mode Encoding encapsulated in the eCM Config File under TLV 202.

B.3 eCM Proxy mechanism for configuration of eRouter[69]

The eRouter configuration encodings are encapsulated in the 'eCM Config File Encapsulation' encoding defined in [eDOCSIS]. The eCM receives the configuration file and parses its contents. The encodings in the eCM configuration file encapsulated in Type 202 are for exclusive use of the eRouter, and these TLVs are transferred from the eCM to the eRouter in a vendor specific manner. This TLV may appear multiple times. If this TLV setting appears multiple times, all sub-TLVs MUST be considered by the eRouter to be part of a single configuration. In other words, the sub-TLVs from the first instance of this configuration setting would comprise the first entries; the second instance would comprise the next. After the eCM successfully completes registration, the eRouter uses these encapsulated TLVs for initialization.

The eRouter initializes per the 'eRouter Operation Mode' encoding, encapsulated under the TLV 202 in the eCMs configuration file. During the eRouter initialization process, the eCM reports the eRouter state with the Flow Step information and status in the esafeProvisioningStatusTable [eDOCSIS].

The eCM configuration download process includes certain security aspects; e.g., EAE and secure download which provide for confidentiality and authenticity of the information contained in the CM configuration file as defined in [MULPIv3.0] and [SECv3.0].

[68] Added per eRouter-N-13.1091-4 on 3/11/13 by PO, updated by eRouter-N-13.1127-4 on 3/24/14 by PO.
[69] Revised per eRouter-N-12.1034-1 on 3/5/12 by JB.

B.4 eRouter Configuration Encodings[70]

This section defines the encodings required for eRouter configuration and how those are processed by the eRouter. All of the TLVs listed here are sub-TLVs of Type 202.

B.4.1 eRouter TLV Processing[71]

The eRouter MUST disregard encodings that are not defined in this section.

The following subsections provide definitions of Configuration Encodings that are valid for eRouter use. .The eRouter MUST ignore invalid eRouter Configuration Encodings. When the eRouter Configuration Encodings are ignored, the eRouter MUST follow the behavior described in Annex C.

The eRouter MUST ignore the eRouter Configuration Encoding if that encoding results in an entry in the SNMP table that cannot be created because of a conflict with an existing entry.

B.4.2 eRouter Initialization Mode Encoding[72]

This encoding defines the eRouter initialization mode (Section 6) configured by the Operator.

A valid eRouter Initialization Mode Encoding contains exactly one instance of this TLV.

Type	Length	Value
1	1	0: Disabled 1: IPv4 Protocol Enabled 2: IPv6 Protocol Enabled 3: Dual IP Protocol Enabled 4-255: Invalid Default: 3 (Dual IP Protocol Enabled)

The eRouter will use 'Dual IP Protocol Enabled' mode by default per Section 6, as recommended by [RFC 6540].

B.4.3 TR-069 Management Server[73]

This encoding specifies some aspects of TR-069 Device.ManagementServer object to be used by the cable provisioning system. Whenever a TLV or sub-TLV is absent, default values from [TR-069a4] and [TR-181i2a8] apply.

Type	Length	Value
2	N	Composite

B.4.3.1 EnableCWMP

This encoding specifies the Device.ManagementServer.EnableCWMP parameter from [TR-181i2a8].

A valid eRouter Initialization Mode Encoding contains at most one instance of this TLV.

Type	Length	Value
2.1	1	0: false 1: true

B.4.3.2 URL

This encoding specifies the Device.ManagementServer.URL parameter from [TR-181i2a8].

[70] Revised per eRouter N-11.1014-3 on 10/27/11 by JB, updated by eRouter-N-13.1127-4 on 3/24/14 by PO.
[71] Revised per eRouter-N-13.1127-4 on 3/24/14 by PO.
[72] Revised per eRouter-N-13.1091-4 on 3/11/13 by PO.
[73] Revised per eRouter N-12.1034-1 on 3/5/12 by JB.

A valid eRouter Initialization Mode Encoding contains at most one instance of this TLV.

Type	Length	Value
2.2	n	String

B.4.3.3 Username

This encoding specifies the Device.ManagementServer.Username parameter from [TR-181i2a8].

A valid eRouter Initialization Mode Encoding contains at most one instance of this TLV.

Type	Length	Value
2.3	n	String

B.4.3.4 Password

This encoding specifies the Device.ManagementServer.Password parameter from [TR-181i2a8].

A valid eRouter Initialization Mode Encoding contains at most one instance of this TLV.

Type	Length	Value
2.4	n	String

B.4.3.5 ConnectionRequestUsername

This encoding specifies the Device.ManagementServer.ConnectionRequestUsername parameter from [TR-181i2a8].

A valid eRouter Initialization Mode Encoding contains at most one instance of this TLV.

Type	Length	Value
2.5	n	String

B.4.3.6 ConnectionRequestPassword

This encoding specifies the Device.ManagementServer.ConnectionRequestPassword parameter from [TR-181i2a8].

A valid eRouter Initialization Mode Encoding contains at most one instance of this TLV.

Type	Length	Value
2.6	n	String

B.4.3.7 ACS Override[74]

If enabled, the CPE MUST accept the ACS URL from the CM configuration file, even if the ACS has overwritten the values.

If disabled, the CPE accepts the CM configuration file values only if the ACS has not overwritten the ACS URL.

Type	Length	Value
2.7	N	0: disabled 1: enabled

[74] Revised per eRouter N-12.1034-1 on 3/5/12 by JB.

B.4.4 eRouter Initialization Mode Override[75]

The eRouter Initialization Mode Override encoding provides a means of overriding the eRouter Initialization Mode encoding on an eRouter'configured to be 'Disabled'. This encoding applies only when eRouter functionality is 'Disabled', such as when the eRouter is manually disabled by the subscriber, service technician, or installer. In all other cases, this override encoding is ignored.

The default value of this TLV encoding (when omitted) is zero (0).

Type	Length	Value
3	1	1 = Ignore eRouter Initialization Mode TLV and keep the eRouter Disabled 0 = Follow eRouter Initialization Mode TLV Default: 0

B.4.5 SNMPv1v2c Coexistence Configuration

This encoding specifies the SNMPv1v2c Coexistence Access Control configuration for the eRouter. This encoding creates entries in the SNMPv3 framework tables as specified in Annex B.1.3.1 above.

A valid SNMPv1v2c Coexistence Configuration (Type 53) encoding contains the SNMPv1v2c Community Name and one or more instance(s) of SNMPv1v2c Transport Address Access. A valid SNMPv1v2c Coexistence Configuration (Type 53) encoding may also contain the SNMPv1v2c Access View Type and the SNMPv1v2c Access View Name.

The eRouter does not make persistent entries in the SNMP framework table.

The eRouter MUST support a minimum of five (5) SNMPv1v2c Coexistence Configuration encodings.

Type	Length	Value
53	N	Composite

B.4.5.1 SNMPv1v2c Community Name

This sub-TLV specifies the Community Name (community string) used in SNMP requests to the eRouter.

Type	Length	Value
53.1	1..32	Text

B.4.5.2 SNMPv1v2c Transport Address Access

This sub-TLV specifies the Transport Address and Transport Address Mask pair used by the eRouter to grant access to the SNMP entity querying the eRouter.

Type	Length	Value
53.2	N	Variable

A valid SNMPv1v2c Transport Address Access encoding contains one instance of SNMPv1v2c Transport Address and may contain one instance of SNMPv1v2c Transport Address Mask.

The eRouter accepts one or more instances of sub-TLV 53.2 SNMPv1v2c Transport Address Access within a TLV 53.

B.4.5.2.1 SNMPv1v2c Transport Address

This sub-TLV specifies the Transport Address to use in conjunction with the Transport Address Mask used by the eRouter to grant access to the SNMP entity querying the eRouter.

[75] Revised per eRouter-N-13.1091-4 on 3/11/13 by PO, updated by eRouter-N-13.1127-4 on 3/24/14 by PO.

Type	Length	Value
53.2.1	6 or 18	Transport Address

Transport addresses are 6 or 18 bytes in length for IPv4 and IPv6 type addresses respectively.

B.4.5.2.2 SNMPv1v2c Transport Address Mask

This sub-TLV specifies the Transport Address Mask to use in conjunction with the Transport Address used by the eRouter to grant access to the SNMP entity querying the eRouter. This sub-TLV is optional.

Type	Length	Value
53.2.2	6 or 18	Transport Address Mask

Transport addresses are 6 or 18 bytes in length for IPv4 and IPv6 type addresses respectively.

B.4.5.3 SNMPv1v2c Access View Type

The SNMPv1v2c Access View Type encoding specifies the type of access to grant to the community name specified in the SNMPv1v2c Community Name encoding. This TLV is optional. If this TLV is not present, the eRouter MUST set the value of the SNMPv1v2c Access View Type to Read-Only.

Type	Length	Value
53.3	1	1: Read-only 2: Read-write

B.4.5.4 SNMPv1v2c Access View Name

This sub-TLV specifies the name of the view that provides the access indicated in the SNMPv1v2c Access View Type. This sub-TLV is optional.

Type	Length	Value
53.4	1.32	String

B.4.6 SNMPv3 Access View Configuration

This encoding specifies the SNMPv3 Simplified Access View configuration of the eRouter. This TLV creates entries in SNMPv3 tables.

The eRouter supports SNMPv3 Access View Configuration encoding only if the eRouter supports SNMPv3.

A valid SNMPv3 Access View Configuration encoding contains one instance of SNMPv3 Access View Name. The eRouter does not make persistent entries in the SNMP framework table.

The eRouter MUST reject the eRouter Configuration Encoding if an eRouter created entry in an SNMP table is rejected due reaching the limit in the number of entries supported for that table.

Type	Length	Value
54	N	Composite

B.4.6.1 SNMPv3 Access View Name

This encoding specifies the administrative name of the view defined by the SNMPv3 Access View Configuration.

Type	Length	Value
54.1	1..32	Text

B.4.6.2 SNMPv3 Access View Subtree

This encoding specifies an ASN.1 formatted object identifier (OID) that represents the filter sub-tree included in the SNMPv3 Access View Configuration encoding.

A valid SNMPv3 Access View Subtree encoding starts with the ASN.1 Universal type 6 (OID) byte, followed by the ASN.1 length field, and then followed by the ASN.1 encoded object identifier components. For example, the sub-tree 1.3.6 is encoded as 0x06 0x03 0x01 0x03 0x06.

If this encoding is not included under the SNMPv3 Access View Name encoding, the eRouter MUST use the default OID sub-tree of 1.3.6.

Type	Length	Value
54.2	N	OID

B.4.6.3 SNMPv3 Access View Mask

This sub-TLV specifies the bit mask to apply to the Access View Subtree of the Access View TLV.

Type	Length	Value
54.3	0..16	Bits

This sub-TLV is optional. If this sub-TLV is not present, the eRouter MUST assign a zero-length string to SNMPv3 Access View Mask.

B.4.6.4 SNMPv3 Access View Type

This sub-TLV specifies the inclusion or exclusion of the sub-tree indicated by SNMPv3 Access View Subtree. The value of 1 indicates that the sub-tree of SNMPv3 Access View SubTree is included in the Access View. The value of 2 indicates that the sub-tree of SNMPv3 Access View Sub Tree is excluded from the Access View.

Type	Length	Value
54.4	1	1: included 2: excluded

This sub-TLV is optional. If this sub-TLV is not present, the eRouter MUST assign the value 'included' to SNMPv3 Access View Type.

B.4.7 Vendor Specific Information

The Vendor Specific Information encoding is used to extend the capabilities of the eRouter specification, through the use of vendor-specific features. A valid Vendor Specific Information encoding contains only one Vendor ID field (see Annex B.4.7.1) to indicate that the settings apply to a specific vendor device.

The eRouter MUST ignore a Vendor Specific Information encoding that includes a Vendor ID different to the one of the eRouter.

Type	Length	Value
43	N	Variable

B.4.7.1 Vendor ID Encoding

The Vendor ID encoding contains the vendor identification specified by the three-byte vendor-specific Organization Unique Identifier of the eRouter's MAC addresses.

The Vendor ID 0xFFFFFF is reserved.

Type	Length	Value
43.8	3	OUI

B.4.8 SNMP MIB Object[76]

If the eRouter relies upon SNMP to configure and manage the device, it MUST support the ability to SET SNMP MIB objects defined in this specification via the CM's DOCSIS configuration file.

Type	Length	Value
11	N	variable binding

The value is an SNMP VarBind as defined in [RFC 1157]. The VarBind is encoded in ASN.1 Basic Encoding Rules, just as it would be if part of an SNMP SET request.

The eRouter treats this encoding as if it were part of an SNMP SET request with the following caveats:

- The request is treated as fully authorized (it cannot refuse the request for lack of privilege).
- SNMP Write-Control provisions do not apply.
- No SNMP response is generated by the eRouter.

This encoding may be repeated with different VarBinds to "Set" a number of MIB objects. All such Sets are treated by the eRouter as if simultaneous.

Each VarBind is limited to 255 bytes.

B.4.9 Topology Mode Encoding[77]

This encoding defines the eRouter Topology Mode used for subdividing an Operator-delegated IPv6 prefix (Section 8.5).

A valid eRouter Topology Mode Encoding contains exactly one instance of this TLV.

Type	Length	Value
42	1	1: Favor Depth 2: Favor Width

If this encoding is absent, the eRouter SHOULD set the Topology Mode as follows unless administratively reconfigured:

- If the eRouter has fewer than 8 Customer-Facing Interfaces, set the Topology Mode to "Favor Depth".
- If the eRouter has 8 or more Customer-Facing Interfaces, set the Topology Mode to "Favor Width".

Customer-Facing Interfaces (physical ports) include RJ-45 Ethernet ports, 802.11 radios, MoCA ports, and USB ports that are capable of supporting network interconnections. However, Customer-Facing Interfaces do not include SSIDs, VLANs, or other logical interfaces for the purposes of setting the Topology Mode.

B.4.10 Router Advertisement (RA) Transmission Interval[78]

This encoding specifies the eRouter's Router Advertisement (RA) transmission period. The eRouter MUST support the Router Advertisement (RA) Transmission Interval encoding.

[76] Section added per eRouter-N-12.1080-4 on 1/4/13 by PO. Revised per eRouter-N-14.1233-1 and eRouter-N-15.1234-2 on 2/13/15 by JB.

[77] Section added per eRouter-N-13.1090-5 3/7/13, PO.

[78] Section added per eRouter-N-13.1101-3 on 5/24/13 by JB.

Type	Length	Value
10	N	Integer between 3-1800

The value is the number of seconds, between 3 and 1800, to which the eRouter's RA transmission period is set. .If this encoding is absent, the eRouter MUST set its RA transmission period to the default of 30 seconds.

B.5 SNMP Soft Reset[79]

The esafeErouterSoftReset object is defined in the "eSAFE MIB Definition" section of [eDOCSIS].

The Soft Reset object provides a mechanism to Soft Reset the eRouter. A "soft" reset differs from a "hard" reset in that a Soft Reset reinitializes the software layer of the eRouter eSAFE, and leaves the embedded CM and any other embedded eSAFE applications unaffected. The function of the Soft Reset control object is to clear the operational state information of the eRouter (e.g., ARP tables, NAT translation table bindings, Neighbor Discovery caches, etc.), force Operator-Facing Interface IP provisioning to be restarted, and trigger CPE IP provisioning.

The eRouter only evaluates the Soft Reset object when it is modified via an SNMP SET initiated from an SNMP management station after the eRouter has been fully initialized. The eRouter MUST ignore the esafeErouterSoftReset whenever it is included in TLV202.11 encodings within the eCM configuration file.

The esafeErouterSoftReset object MUST be accessible via the eCM SNMP agent through the eCM management address.

Setting esafeErouterSoftReset to true(1) causes the eRouter to perform a Soft Reset, without resetting the eCM or any other eSAFEs. An SNMP GET/GETNEXT (poll) of this object always returns a value of false(2).

When esafeErouterSoftReset is set to true(1), the eRouter MUST perform a Soft Reset in the following order:

1. Retain all current running configuration information. This information includes any TLV202 CM configuration file TLV entries previously learned during eCM provisioning and any configuration information that would normally be saved across any form of reset.
2. Immediately notify CPEs on the Customer-Facing Interfaces of impending reset:
   a. Send IPv4 IPv6 DHCP RECONFIGURE (type 6, Rebind) on all Customer-Facing Interfaces per [RFC 6644]. The eRouter ignores all DHCP messages on Customer-Facing Interfaces after sending the RECONFIGURE until it has completed the reset operation and has successfully completed IPv6 provisioning on its Operator-Facing Interface.
   b. Send Router Advertisements (RAs) with the current valid and preferred prefix liftetimes, router lifetime, and the M, A, and O bits all set to zero (0) on all IPv6-enabled Customer-Facing Interfaces. The eRouter continues sending RAs with these lifetime values and provisioning bits set to 0 until the reset operation has completed and its Operator-Facing Interface has successfully completed IPv6 provisioning.
3. Disable all Customer Facing Interfaces. This action will shut down the physical link state of all Customer Facing Interfaces.
4. Release Operator-Facing Interface provisioning information as mandated in Section 7.4 and Section 8.6.
5. Reset the eRouter, this includes clearing all operational state information.
6. Enable all Customer Facing Interfaces a minimum of 100 ms after completing step 3. This action will start up the physical link state of all Customer Facing Interfaces in order to aid in the triggering of CPE re-provisioning operations.
7. Perform eRouter Initialization, as described in Section 6, using the running configuration information retained in step one (1) of this process.

[79] Section added per eRouter-N-13.1108-2 on 7/16/13 by PO, updated by eRouter-N-13.1127-4 on 3/24/14 by PO.

In the event of a conflict between the eRouter's retained configuration and configuration information obtained during the provisioning of the Operator-Facing Interface, the eRouter MUST prefer new information received during provisioning, which will take precedence over the retained information.

This Soft Reset capability is very useful in situations in which the customer has experienced problems with interface addressing or system faults that can typically be resolved efficiently by remotely resetting the eRouter. Because additional eSAFEs may be implemented in the same device, targeting only the eRouter prevents video streams or voice calls from being terminated as would occur when the entire device is rebooted.

B.6 Provisioning and Operational Event Messages[80]

This list of Event Messages will facilitate resolution of issues and is focused toward the visiting technician's use.

Device	Error Code	Event ID	Severity	Intf	Event Message Text	Variables	Message Counter	Time Stamp
eRouter	H10.1	72001001	Informational		OFI - DHCPv4 Provisioning Complete	NA	NA	<time>
eRouter	H10.2	72001002	Informational		OFI - DHCPv6 Provisioning Complete	NA	NA	<time>
eRouter	H10.3	72001003	Critical		OFI - DHCPv4 Provisioning - X Retries attempted; Last attempt at <time>	NA	<count>	<time>
eRouter	H10.4	72001004	Critical		OFI - DHCPv6 Provisioning - X Retries attempted; Last attempt at <time>	NA	<count>	<time>
eRouter	H10.5	72001005	Critical		OFI - ICMPv6 No RA message received in response to RS	NA	<count>	<time>
eRouter	H10.6	72001006	Critical		OFI - ICMPv6 RA not properly configured for DHCPv6 (M = 0)	NA	NA	<time>
eRouter	H10.7	72001007	Critical		OFI - ICMPv6 Link Local DAD issue - Duplicate IP address detected	NA	NA	<time>
eRouter	H10.8	72001008	Critical		OFI - DHCPv4 No Offer / Ack message received	NA	<count>	<time>
eRouter	H10.9	72001009	Critical		OFI - DHCPv6 No Advertise / Reply message received	NA	<count>	<time>
eRouter	H10.10	72001010	Critical		OFI - DHCPv4 Missing Required DHCP option	<option #>	NA	<time>
eRouter	H10.11	72001011	Critical		OFI - DHCPv6 Missing Required DHCP option	<option #>	NA	<time>
eRouter	H10.12	72001012	Critical		OFI - DHCPv4 Bad value in required DHCP option	<option #>	NA	<time>
eRouter	H10.13	72001013	Critical		OFI - DHCPv6 Bad value in required DHCP option	<option #>	NA	<time>
eRouter	H10.14	72001014	Critical		OFI - DHCPv6 failed - No Address Available	NA	NA	<time>
eRouter	H10.15	72001015	Critical		OFI - DHCPv6 failed - No Prefix Available	NA	NA	<time>
eRouter	H10.16	72001016	Critical		OFI - DHCPv6 GUA DAD issue - Duplicate IP address detected	NA	NA	<time>
eRouter	H10.17	72001017	Critical		OFI - DHCPv6 Failure to renew Address	NA	<count>	<time>

[80] Section added per eRouter-N-13.1108-2 on 7/16/13 by PO, updated by eRouter-N-13.1127-4 on 3/24/14 by PO.

Device	Error Code	Event ID	Severity	Intf	Event Message Text	Variables	Message Counter	Time Stamp
eRouter	H10.18	72001018	Critical		OFI - DHCPv6 Failure to renew Prefix	NA	<count>	<time>
eRouter	H10.19	72001019	Critical		OFI - DHCPv4 Failure to renew lease	NA	<count>	<time>
eRouter	H10.20	72001020	Informational		OFI - DHCPv4 IP address released	NA	NA	<time>
eRouter	H10.21	72001021	Informational		OFI - DHCPv6 IP address released	NA	NA	<time>
eRouter	H20.1	72002001	Critical		CFI - LAN Provisioning No Prefix available for eRouter interface(s)	<interface #>	NA	<time>
eRouter	H20.2	72002002	Critical		CFI - LAN Provisioning DHCPv6-PD No Prefix available - Subdelegation	NA	Client ID (duid)	<time>
eRouter	H20.3	72002003	Informational		CFI - LAN Provisioning DHCPv6-PD PD allocation mode set to width	NA	NA	<time>
eRouter	H20.4	72002004	Informational		CFI - LAN Provisioning DHCPv6-PD PD allocation mode set to depth	NA	NA	<time>
eRouter	H20.5	72002005	Informational		CFI - LAN Provisioning DHCPv6-PD PD hint of length 'X' received	X = [Null], Range [48 - 64]	Client ID (duid)	<time>
eRouter	H20.6	72002006	Alert		CFI - LAN Provisioning DHCPv6-PD Allocated prefix size X less than requested prefix Y - Subdelegation	X, Y	Client ID (duid)	<time>
eRouter	H20.7	72002007	Critical		CFI - LAN Provisioning DHCPv6 No client addresses available	NA	Client ID (duid)	<time>
eRouter	H20.8	72002008	Critical		CFI - Link Local DAD - Duplicate IP address detected	NA	Client ID (duid)	<time>
eRouter	H20.9	72002009	Critical		CFI - GUA DAD - Duplicate IP address detected	NA	Client ID (duid)	<time>
eRouter	H20.10	72002010	Critical		CFI - DHCPv6 Reconfigure Failure	NA	Client ID (duid)	<time>
eRouter	H20.11	72002011	Critical		CFI - LAN Provisioning DHCPv4 No Address available	NA	NA	<time>
eRouter	H20.12	72002012	Informational		CFI - LAN Provisioning DHCPv4 IP Address Conflict	NA	CHADDR	<time>
eRouter	H30.1	72003001	Informational		eRouter is administratively disabled	NA	NA	<time>
eRouter	H30.2	72003002	Informational		eRouter is enabled as IPv4 Only	NA	NA	<time>
eRouter	H30.3	72003003	Informational		eRouter is enabled as IPv6 Only	NA	NA	<time>
eRouter	H30.4	72003004	Informational		eRouter is enabled as Dual Stack	NA	NA	<time>

Annex C eRouter Initialization Mode Control Interactions[81]

The table in this annex defines the interactions between the methods available for configuring the eRouter Initialization Mode, and the expected operational mode.

The table includes interactions between the following methods of configuring eRouter Initialization Mode:

- eSafeErouterInitModeControl MIB object
- eRouter Initialization Mode Encoding (TLV 202.1)
- eRouter Initialization Mode Override Encoding (TLV 202.3)
- previous initialization mode value stored in NVRAM, referred to in this Annex as the “previous persistent value”. This is the value previously supplied via the TLV 202.1 encoding.

The value of the previous persistent mode value stored in NVRAM is condensed into two possible values.

1. 'Disabled'
2. 'Not Disabled'

There is no difference in behavior for any of the modes that fall into the 'Not Disabled' condensed value. 'Not Disabled' covers IPv4 only, IPv6 only, IPv4 and IPv6 (dual).

C.1 Assumptions

1. The table represents eRouter initialization behavior after a reset, regardless of whether that reset was “hard” or “soft”, or upon initial bootup of a factory fresh device.
2. It is assumed that whenever the esafeErouterInitModeControl object is set to any value other than honorErouterInitMode(5), the value represented in the previous persistent value column is the current value of eSafeErouterInitModeControl. The value of this object is required to be persistent across resets, and would take precedence over any previously stored persistent TLV 202.1 value.
3. Factory fresh device cases are represented by a previous persistent value of None.

Table C-1 - eRouter Initialization Behavior Based upon Mode Control Interactions

	eSafeErouter-InitModeControl	previous persistent value	TLV 202.3	TLV 202.1	Resulting Operational Mode
1	honorErouterInitMode(5)	None	not present	not present	Dual[1]
2	honorErouterInitMode(5)	None	0 or not present	0 (Disabled)	Disabled[1]
3	honorErouterInitMode(5)	None	0 or not present	1 (IPv4)	IPv4 Only[1]
4	honorErouterInitMode(5)	None	0 or not present	2 (IPv6)	IPv6 Only[1]
5	honorErouterInitMode(5)	None	0 or not present	3 (Dual)	Dual[1]
6	honorErouterInitMode(5)	None	1	0 (Disabled)	Disabled[1]
7	honorErouterInitMode(5)	None	1	1 (IPv4)	IPv4 Only[1]
8	honorErouterInitMode(5)	None	1	2 (IPv6)	IPv6 Only[1]
9	honorErouterInitMode(5)	None	1	3 (Dual)	Dual[1]
10	honorErouterInitMode(5)	Disabled	0 or not present	0 (Disabled)	Disabled
11	honorErouterInitMode(5)	Disabled	0 or not present	1 (IPv4)	IPv4 Only
12	honorErouterInitMode(5)	Disabled	0 or not present	2 (IPv6)	IPv6 Only
13	honorErouterInitMode(5)	Disabled	0 or not present	3 (Dual)	Dual
14	honorErouterInitMode(5)	Disabled	0 or not present	not present	Disabled
15	honorErouterInitMode(5)	Disabled	1	0 (Disabled)	Disabled
16	honorErouterInitMode(5)	Disabled	1	1 (IPv4)	Disabled

[81] Annex added per eRouter-N-14.1132-1 on 3/25/14 by PO.

	eSafeErouter-InitModeControl	previous persistent value	TLV 202.3	TLV 202.1	Resulting Operational Mode
17	honorErouterInitMode(5)	Disabled	1	2 (IPv6)	Disabled
18	honorErouterInitMode(5)	Disabled	1	3 (Dual)	Disabled
19	honorErouterInitMode(5)	Disabled	1	not present	Disabled
20	honorErouterInitMode(5)	Not Disabled	0 or not present	0 (Disabled)	Disabled
21	honorErouterInitMode(5)	Not Disabled	0 or not present	1 (IPv4)	IPv4 Only
22	honorErouterInitMode(5)	Not Disabled	0 or not present	2 (IPv6)	IPv6 Only
23	honorErouterInitMode(5)	Not Disabled	0 or not present	3 (Dual)	Dual
24	honorErouterInitMode(5)	Not Disabled	0 or not present	not present	Previous 'Not Disabled' Mode
25	honorErouterInitMode(5)	Not Disabled	1	0 (Disabled)	Disabled
26	honorErouterInitMode(5)	Not Disabled	1	1 (IPv4)	IPv4 Only
27	honorErouterInitMode(5)	Not Disabled	1	2 (IPv6)	IPv6 Only
28	honorErouterInitMode(5)	Not Disabled	1	3 (Dual)	Dual
29	honorErouterInitMode(5)	Not Disabled	1	not present	Previous 'Not Disabled' Mode
30	ipDisabled(1)	Disabled	0 or not present	0 (Disabled)	Disabled
31	ipDisabled(1)	Disabled	0 or not present	1 (IPv4)	Disabled
32	ipDisabled(1)	Disabled	0 or not present	2 (IPv6)	Disabled
33	ipDisabled(1)	Disabled	0 or not present	3 (Dual)	Disabled
34	ipDisabled(1)	Disabled	0 or not present	not present	Disabled
35	ipDisabled(1)	Disabled	1	0 (Disabled)	Disabled
36	ipDisabled(1)	Disabled	1	1 (IPv4)	Disabled
37	ipDisabled(1)	Disabled	1	2 (IPv6)	Disabled
38	ipDisabled(1)	Disabled	1	3 (Dual)	Disabled
39	ipv4Only(2)	IPv4Only	0 or not present	0 (Disabled)	IPv4Only
40	ipv4Only(2)	IPv4Only	0 or not present	1 (IPv4)	IPv4Only
41	ipv4Only(2)	IPv4Only	0 or not present	2 (IPv6)	IPv4Only
42	ipv4Only(2)	IPv4Only	0 or not present	3 (Dual)	IPv4Only
43	ipv4Only(2)	IPv4Only	0 or not present	not present	IPv4Only
44	ipv4Only(2)	IPv4Only	1	0 (Disabled)	IPv4Only
45	ipv4Only(2)	IPv4Only	1	1 (IPv4)	IPv4Only
46	ipv4Only(2)	IPv4Only	1	2 (IPv6)	IPv4Only
47	ipv4Only(2)	IPv4Only	1	3 (Dual)	IPv4Only
48	ipv4Only(2)	IPv4Only	1	not present	IPv4Only
49	ipv6Only(3)	IPv6Only	0 or not present	0 (Disabled)	IPv6Only
50	ipv6Only(3)	IPv6Only	0 or not present	1 (IPv4)	IPv6Only
51	ipv6Only(3)	IPv6Only	0 or not present	2 (IPv6)	IPv6Only
52	ipv6Only(3)	IPv6Only	0 or not present	3 (Dual)	IPv6Only
53	ipv6Only(3)	IPv6Only	0 or not present	not present	IPv6Only
54	ipv6Only(3)	IPv6Only	1	0 (Disabled)	IPv6Only
55	ipv6Only(3)	IPv6Only	1	1 (IPv4)	IPv6Only
56	ipv6Only(3)	IPv6Only	1	2 (IPv6)	IPv6Only
57	ipv6Only(3)	IPv6Only	1	3 (Dual)	IPv6Only
58	ipv6Only(3)	IPv6Only	1	not present	IPv6Only
59	ipv4AndIpv6(4)	Dual	0 or not present	0 (Disabled)	Dual
60	ipv4AndIpv6(4)	Dual	0 or not present	1 (IPv4)	Dual

	eSafeErouter-InitModeControl	previous persistent value	TLV 202.3	TLV 202.1	Resulting Operational Mode
61	ipv4AndIpv6(4)	Dual	0 or not present	2 (IPv6)	Dual
62	ipv4AndIpv6(4)	Dual	0 or not present	3 (Dual)	Dual
63	ipv4AndIpv6(4)	Dual	0 or not present	not present	Dual
64	ipv4AndIpv6(4)	Dual	1	0 (Disabled)	Dual
65	ipv4AndIpv6(4)	Dual	1	1 (IPv4)	Dual
66	ipv4AndIpv6(4)	Dual	1	2 (IPv6)	Dual
67	ipv4AndIpv6(4)	Dual	1	3 (Dual)	Dual
68	ipv4AndIpv6(4)	Dual	1	not present	Dual

[1] This entry in the table represents the case of a device booting up for the first time after being taken out of the box, or a device being booted up after a factory reset.

C.2 Invalid Cases

Invalid cases cannot exist. They are included in this Annex for the purpose of completeness. If eSafeErouterInitModeControl was set to anything but honorErouterInitMode(5), it means the device has been previously initialized, as eSafeErouterInitModeControl can only be set via SNMP *after* provisioning is complete. This means there has to be a previous persistent value. If no Initialization Mode Encoding was present in the Cable Modem configuration file, the default would be 'Dual', so the previous persistent value would always be Dual for these cases.

Table C-2 - Invalid Cases

eSafeErouter-InitModeControl	previous persistent value	TLV 202.3	TLV 202.1	Resulting Operational Mode
ipDisabled(1)	None	1, 0, or not present	Any value or not present	N/A
ipv4Only(2)	None	1, 0, or not present	Any value or not present	N/A
Ipv6Only(3)	None	1, 0, or not present	Any value or not present	N/A
ipv4Only(2)	None	1, 0, or not present	Any value or not present	N/A

Annex D TR-069 Managed Objects Requirements[82]

The eRouter MUST support the objects associated with the Profiles and Components listed below. See [TR-106a5] for information about Components and Profiles in TR-069.

D.1 Profiles from [TR-181i2a8]

Table D-1 - TR-181 Profiles for eRouter

Profile	Requirement	Notes
Download:1	MAY	
DownloadTCP:1	MAY	
Upload:1	MAY	
UploadTCP:1	MAY	
UDPEcho:1	MAY	
UDPEchoPlus:1	MAY	
SupportedDataModel:1	MUST	
SupportedDataModel:2	MUST	
MemoryStatus:1	MAY	
ProcessStatus:1	MAY	
TempStatus:1	MAY	
TempStatusAdv:1	MAY	
TempStatusAdv:2	MAY	
User:1	MUST	
UPnPDev:1	MUST	Support Data model, other specs to detail UPNP functional requirements
UPnPDiscBasic:1	MUST	Support Data model, other specs to detail UPNP functional requirements
UPnPDiscAdv:1	MAY	
UPnPDiscAdv:2	MAY	
SelfTestDiag:1	MAY	
NSLookupDiag:1	MAY	
SimpleFirewall:1	MUST	
AdvancedFirewall:1	MUST	
Baseline:3	MUST	
DNSRelay:1	MAY	
Routing:1	MUST	
Routing:2	MUST	
IPv6Routing:1	MUST	

[82] Annex added per eRouter N-11.1014-3 on 10/27/11 by JB. Revised per eRouter-N-14.1179-2 on 212/15 by JB.

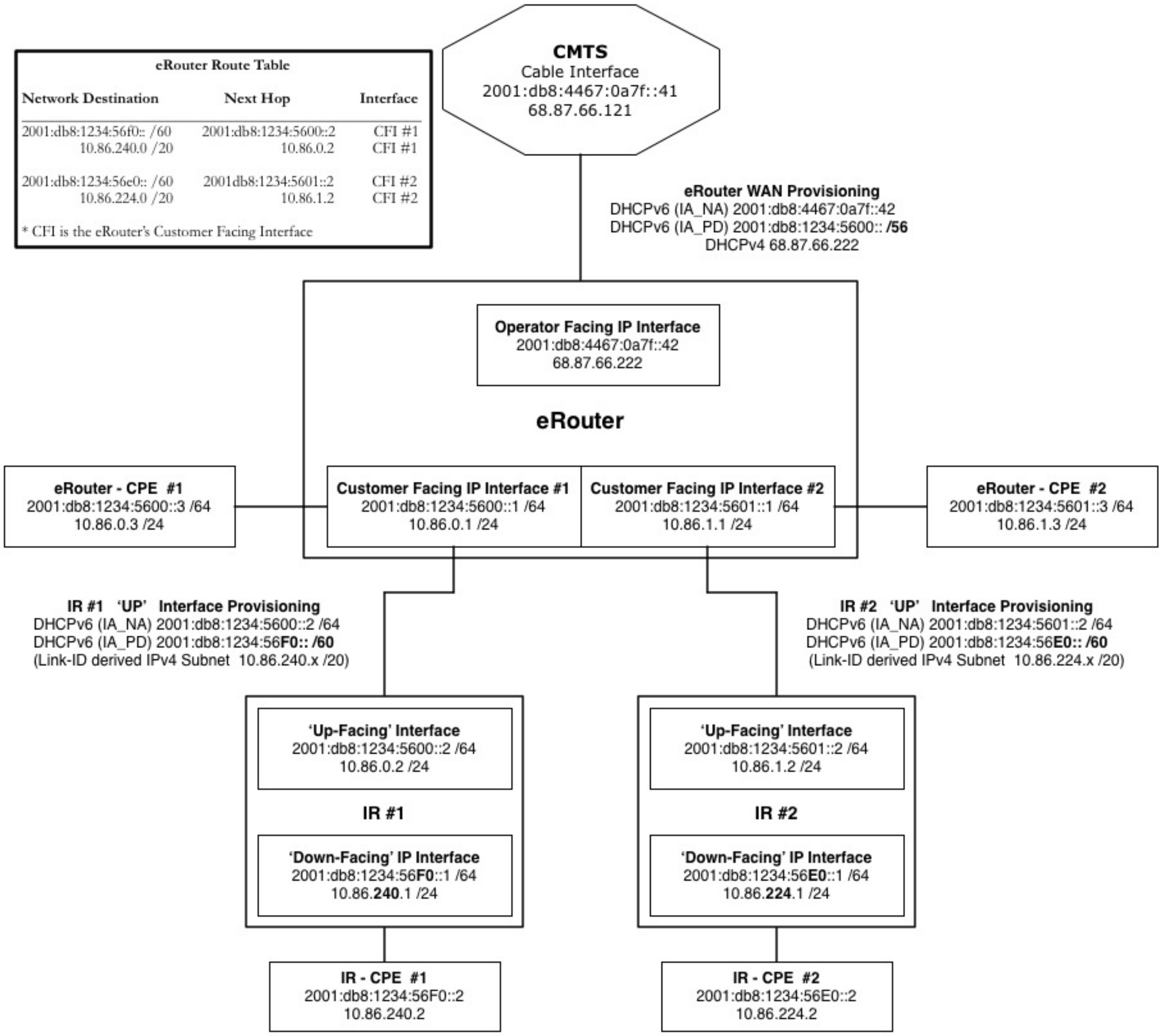
Profile	Requirement	Notes
IPInterface:2	MUST	
IPv6Interface:1	MUST	
VLANTermination:1	MUST	
EthernetLink:1	MUST	
Bridge:1	MUST	
VLANBridge:1	MUST	
BridgeFilter:1	MUST	
BridgeFilter:2	MUST	
EthernetInterface:1	MUST	
HPNA:1	MUST if interface supported	
HPNADiagnostics:1	MUST if interface supported	
HPNAQoS:1	MUST if interface supported	
HomePlug:1	MUST if interface supported	
MoCA:1	MUST if interface supported	
WiFiRadio:1	Per [WI-FI MGMT]	
WiFiSSID:1	Per [WI-FI MGMT]	
WiFiAccessPoint:1	Per [WI-FI MGMT]	
WiFiEndPoint:1	Per [WI-FI MGMT]	
USBInterface:1	MUST if interface supported	
USBPort:1	MUST if interface supported	
NAT:1	MUST	
QoS:2	MAY	
QoSDynamicFlow:1	MAY	
QoSStats:1	MAY	
NeighborDiscovery:1	MUST	
RouterAdvertisement:1	MUST	
IPv6rd:1	MAY	
DSLite:1	MAY	
Hosts:2	MUST	
GatewayInfo:1	MUST	
DeviceAssociation:1	MUST	

Profile	Requirement	Notes
UDPConnReq:1	MAY	
CaptivePortal:1	MAY	
Time:1	MAY	
IEEE8021xAuthentication:1	Per [WI-FI MGMT]	
IPPing:1	MAY	
TraceRoute:1	MAY	
DHCPv4Client:1	MUST	
DHCPv4Server:1	MUST	
DHCPv4CondServing:1	MAY	
DHCPv4Relay:1	MUST NOT	
DHCPv4ServerClientInfo:1	MUST	
DHCPv6Client:1	MUST	
DHCPv6ClientServerIdentity:1	MUST	
DHCPv6Server:1	MUST	
DHCPv6ServerAdv:1	MAY	
DHCPv6ServerClientInfo:1	MUST	
Processors:1	MAY	
VendorLogFiles:1	MAY	
DUStateChngComplPolicy:1	MAY	
SM_ExecEnvs:1	MAY	
SM_DeployAndExecUnits:1	MAY	
SM_Baseline:1	MAY	
Location:1 Profile	MAY	
FaultMgmtSupportedAlarms:1 Profile	MAY	
FaultMgmtActive:1 Profile	MAY	
FaultMgmtHistory:1	MAY	
FaultMgmtExpedited:1 Profile	MAY	
FaultMgmtQueued:1	MAY	
DNS_SD:1 Profile	MUST	
XMPPBasic:1 Profile	MAY	
XMPPReconnect:1 Profile	MAY	
UDPEchoDiag:1	MAY	
ServerSelectionDiag:1	MAY	
InformParameters:1	MAY	

Profile	Requirement	Notes
GRE Basic:1	MUST	
CRE Adv:1	MUST	
PCP	MAY	MUST if DSLite is implemented

Annex E Example: Routing with Link ID (Normative) [83]

This Annex provides example IP addressing and routing using Link ID as described throughout this document. The intention is to provide a reference example to aid in the proper application of Link ID for consistent multi-router packet forwarding without a routing protocol. In this example, an eRouter is provisioned with a /56 IPv6 prefix and (4) Customer Facing IP Interfaces:



eRouter Route Table

Network Destination	Next Hop	Interface
2001:db8:1234:56f0:: /60	2001:db8:1234:5600::2	CFI #1
10.86.240.0 /20	10.86.0.2	CFI #1
2001:db8:1234:56e0:: /60	2001db8:1234:5601::2	CFI #2
10.86.224.0 /20	10.86.1.2	CFI #2

* CFI is the eRouter's Customer Facing Interface

CMTS
Cable Interface
2001:db8:4467:0a7f::41
68.87.66.121
eRouter WAN Provisioning
DHCPv6 (IA_NA) 2001:db8:4467:0a7f::42
DHCPv6 (IA_PD) 2001:db8:1234:5600:: /56
DHCPv4 68.87.66.222
Operator Facing IP Interface
2001:db8:4467:0a7f::42
68.87.66.222
eRouter
eRouter - CPE #1
2001:db8:1234:5600::3 /64
10.86.0.3 /24
Customer Facing IP Interface #1
2001:db8:1234:5600::1 /64
10.86.0.1 /24
Customer Facing IP Interface #2
2001:db8:1234:5601::1 /64
10.86.1.1 /24
eRouter - CPE #2
2001:db8:1234:5601::3 /64
10.86.1.3 /24
IR #1 'UP' Interface Provisioning
DHCPv6 (IA_NA) 2001:db8:1234:5600::2 /64
DHCPv6 (IA_PD) 2001:db8:1234:56F0:: /60
(Link-ID derived IPv4 Subnet 10.86.240.x /20)
IR #2 'UP' Interface Provisioning
DHCPv6 (IA_NA) 2001:db8:1234:5601::2 /64
DHCPv6 (IA_PD) 2001:db8:1234:56E0:: /60
(Link-ID derived IPv4 Subnet 10.86.224.x /20)
'Up-Facing' Interface
2001:db8:1234:5600::2 /64
10.86.0.2 /24
IR #1
'Down-Facing' IP Interface
2001:db8:1234:56F0::1 /64
10.86.240.1 /24
'Up-Facing' Interface
2001:db8:1234:5601::2 /64
10.86.1.2 /24
IR #2
'Down-Facing' IP Interface
2001:db8:1234:56E0::1 /64
10.86.224.1 /24
IR - CPE #1
2001:db8:1234:56F0::2
10.86.240.2
IR - CPE #2
2001:db8:1234:56E0::2
10.86.224.2

Figure E-1 - Example of Link-ID with Prefix Delegation – Topology Mode Favors Width

[83] Revised per eRouter-N-14.1155-3 on 2/10/15 by JB.

Annex F Section Categorizing [RFC 6092] Simple Security Recommendations[84] (Normative)

This section categorizes the recommendations from [RFC 6092] into recommendations for eRouter devices. While the RFC provides a good foundation for the development of a stateful inspection packet filtering firewall, it is not without omission and not all of its recommendations conform with best practices for cable networks. Additionally, the cable industry has developed several security mechanisms that supersede those provided in the recommendations. Where conflicts or recommendations other than those supplied by the RFC occur, they are called out explicitly.

F.1 Summary of Simple Security Requirements

This section provides a quick reference to the [RFC 6092] recommendations required by eRouter.

Critical - see Table F–1:
REC-3, REC-4, REC-5, REC-7, REC-10, REC-12, REC-14, REC-16, REC-18, REC-19, REC-21, REC-22, REC-23, REC-24, REC-25, REC-31, REC-32, REC-35, REC-36, REC-37, MSO-REC.

Important - see Table F–2:
REC-1, REC-2, REC-6, REC-8, REC-9, REC-11, REC-17, REC-33, REC-47.

BCP - see Table F–3:
REC-15, REC-20, REC-26, REC-27, REC-28, REC-29, REC-30, REC-38, REC-40, REC-41, REC-42, REC-43, REC-44, REC-45, REC-46, REC-48.

Other see - Table F–4:
REC-13, REC-49, REC-50.

Conflict - see Table F–5:
REC-34, REC-39.

F.2 Critical Recommendations

The following [RFC 6092] recommendations are critical to network connectivity and are to be included in all eRouter devices. All requirements in this section should be deemed mandatory as noted. These recommendations are in compliance with MSO security requirements for the eRouter as the highest priority for development and testing.

Table F–1 - Critical Recommendations

REC #	RFC 6092 Recommendation Text	Comments
REC-3	Packets bearing source and/or destination addresses forbidden to appear in the outer headers of packets transmitted over the public Internet MUST NOT be forwarded. In particular, site-local addresses are deprecated by [RFC 3879], and [RFC 5156] explicitly forbids the use of addresses with IPv4-Mapped, IPv4-Compatible, Documentation and ORCHID prefixes.	This would be the equivalent of an IPv6 bogon / martians list. Due to the CPU / memory resources of the devices and the fact that once deployed, it won't likely be changed, this would not include unallocated IPv6 space like it might on the backbone.
REC-4	Packets bearing deprecated extension headers prior to their first upper-layer-protocol header SHOULD NOT be forwarded or transmitted on any interface. In particular, all packets with routing extension header type 0 [RFC 2460] preceding the first upper-layer-protocol header MUST NOT be forwarded. (See [RFC 5095] for additional background.)	

[84] Added per eRouter N-12.1081-2 on 1/4/13 by PO. Changed to ANNEX per eRouter-N-14.1233-1 on 2/13/15 by JB.

REC #	RFC 6092 Recommendation Text	Comments
REC-5	Outbound packets MUST NOT be forwarded if the source address in their outer IPv6 header does not have a unicast prefix assigned for use by globally reachable nodes on the interior network.	uRPF like behavior
REC-7	By DEFAULT, packets with unique local source and/or destination addresses [RFC 4193] SHOULD NOT be forwarded to or from the exterior network.	Unique local addresses (ULA) can be forwarded between LAN interfaces on a customer premises router but as defined, should not exit the WAN interface. It is expected that ISP network will not carry routes for ULA address blocks so traffic will be dropped anyway.
REC-10	IPv6 gateways MUST forward ICMPv6 "Destination Unreachable" and "Packet Too Big" messages containing IP headers that match generic upper-layer transport state records.	If not, an MTU size mismatch can prevent connectivity, causing IPv6 sessions to fail. Conversely, if there is no state table entry, drop the packets.
REC-12	Filter state records for generic upper-layer transport protocols MUST NOT be deleted or recycled until an idle timer not less than two minutes has expired without having forwarded a packet matching the state in some configurable amount of time. By DEFAULT, the idle timer for such state records is five minutes.	If the timers are less than 2-5 minutes, many VPN tunnels break because the keep alive timer is often set to 360 seconds
REC-14	A state record for a UDP flow where both source and destination ports are outside the well-known port range (ports 0-1023) MUST NOT expire in less than two minutes of idle time. The value of the UDP state record idle timer MAY be configurable. The DEFAULT is five minutes.	See REC-12 except this applies to low ports instead of high ports.
REC-16	A state record for a UDP flow MUST be refreshed when a packet is forwarded from the interior to the exterior, and it MAY be refreshed when a packet is forwarded in the reverse direction.	
REC-18	If a gateway forwards a UDP flow, it MUST also forward ICMPv6 "Destination Unreachable" and "Packet Too Big" messages containing UDP headers that match the flow state record.	Avoiding breaking path MTU discovery.
REC-19	Receipt of any sort of ICMPv6 message MUST NOT terminate the state record for a UDP flow.	If not supported, this could be employed in a DOS/DDOS attack against a CPE device by causing UDP sessions to close simply by receiving unsolicited ICMP reply messages.
REC-21	In their DEFAULT operating mode, IPv6 gateways MUST NOT prohibit the forwarding of packets, to and from legitimate node addresses, with destination extension headers of type "Authentication Header (AH)" [RFC 4302] in their outer IP extension header chain.	This requirement applies only to IPv6 packets. IPv4 IPsec AH packets should continue to be blocked from the Internet to internal hosts by default.

REC #	RFC 6092 Recommendation Text	Comments
REC-22	In their DEFAULT operating mode, IPv6 gateways MUST NOT prohibit the forwarding of packets, to and from legitimate node addresses, with an upper-layer protocol of type "Encapsulating Security Payload (ESP)" [RFC 4303] in their outer IP extension header chain.	This requirement applies only to IPv6 packets. Hosts sufficient to support IPv6 should support rejecting unrequested AH/ESP packets by any hosts within the LAN/WAN. IPv4 IPsec AH packets should continue to be blocked from the Internet to internal hosts by DEFAULT.
REC-23	If a gateway forwards an ESP flow, it MUST also forward (in the reverse direction) ICMPv6 "Destination Unreachable" and "Packet Too Big" messages containing ESP headers that match the flow state record.	
REC-24	In their DEFAULT operating mode, IPv6 gateways MUST NOT prohibit the forwarding of any UDP packets, to and from legitimate node addresses, with a destination port of 500, i.e., the port reserved by IANA for the Internet Key Exchange (IKE) Protocol [RFC 5996].	Blocking will likely break common L3 VPN (IPsec) connectivity. The IPsec IKE service listening on UDP/500 will not respond if it does not have a corresponding IPsec policy configured. As a result, leaving UDP/500 open could expose hosts to attack but could not be used in a reflection attack. IPv4 UDP/500 packets should continue to be blocked from the Internet to internal hosts by DEFAULT.
REC-25	In all operating modes, IPv6 gateways SHOULD use filter state records for Encapsulating Security Payload (ESP) [RFC 4303] that are indexable by a 3-tuple comprising the interior node address, the exterior node address, and the ESP protocol identifier. In particular, the IPv4/NAT method of indexing state records also by security parameters index (SPI) SHOULD NOT be used. Likewise, any mechanism that depends on detection of Internet Key Exchange (IKE) [RFC 5996] initiations SHOULD NOT be used.	ESP protocol identifier interactions may preclude more than one tunnel per endpoint.
REC-31	All valid sequences of TCP packets (defined in [RFC 793] MUST be forwarded for outbound flows and explicitly permitted inbound flows. In particular, both the normal TCP 3-way handshake mode of operation and the simultaneous-open mode of operation MUST be supported.	
REC-32	The TCP window invariant MUST NOT be enforced on flows for which the filter did not detect whether the window-scale option (see [RFC 1323]) was sent in the 3-way handshake or simultaneous-open.	Fast start support. May be more difficult for vendors, but could be necessary for high-latency connections.
REC-35	If a gateway cannot determine whether the endpoints of a TCP flow are active, then it MAY abandon the state record if it has been idle for some time. In such cases, the value of the "established flow idle-timeout" MUST NOT be less than two hours four minutes, as discussed in [RFC 5382]. The value of the "transitory flow idle-timeout" MUST NOT be less than four minutes. The value of the idle-timeouts MAY be configurable by the network administrator.	

REC #	RFC 6092 Recommendation Text	Comments
REC-36	If a gateway forwards a TCP flow, it MUST also forward ICMPv6 "Destination Unreachable" and "Packet Too Big" messages containing TCP headers that match the flow state record.	Path MTU discovery and accessibility necessary for connectivity.
REC-37	Receipt of any sort of ICMPv6 message MUST NOT terminate the state record for a TCP flow.	This will prevent DoS against router due to unsolicited ICMPv6 messages.
MSO-REC	By default an IGW MUST deny any protocol received on the WAN (operator facing) interface not specifically allowed by configuration with the following exceptions: DHCP, ND, ICMP and established TCP & UDP flows.	This recommendation is not found in [RFC 6092] but support is required in eRouter devices.

F.3 Important Recommendations

Failure to implement these [RFC 6092] recommendations could expose subscribers to infosec attacks. All eRouter implementations should support the list below as security requirements. The requirements below should be developed and tested after all critical requirements (Section F.2) are satisfied.

Table F–2 - Important Recommendations

REC #	RFC 6092 Recommendation Text	Comments
REC-1	Packets bearing in their outer IPv6 headers multicast source addresses MUST NOT be forwarded or transmitted on any interface.	
REC-2	Packets which bear in their outer IPv6 headers multicast destination addresses of equal or narrower scope (see IPv6 Scoped Address Architecture [RFC 4007]) than the configured scope boundary level of the gateway MUST NOT be forwarded in any direction. The DEFAULT scope boundary level SHOULD be organization-local scope, and it SHOULD be configurable by the network administrator.	
REC-6	Inbound packets MUST NOT be forwarded if the source address in their outer IPv6 header has a global unicast prefix assigned for use by globally reachable nodes on the interior network.	Anti-spoofing
REC-8	By DEFAULT, inbound DNS queries received on exterior interfaces MUST NOT be processed by any integrated DNS resolving server.	Prevents DNS reflection attacks. It will also prevent subscribers from hosting a DNS server behind a router by default.
REC-9	Inbound DHCPv6 discovery packets [RFC 3315] received on exterior interfaces MUST NOT be processed by any integrated DHCPv6 server or relay agent.	Prevent recon scans (work around for vast IPv6 address space).

REC #	RFC 6092 Recommendation Text	Comments
REC-11	If application transparency is most important, then a stateful packet filter SHOULD have "endpoint independent filter" behavior for generic upper-layer transport protocols. If a more stringent filtering behavior is most important, then a filter SHOULD have "address dependent filtering" behavior. The filtering behavior MAY be an option configurable by the network administrator, and it MAY be independent of the filtering behavior for other protocols. Filtering behavior SHOULD be endpoint independent by DEFAULT in gateways intended for provisioning without service-provider management.	For example, this would support allowing all http but reduces ability to block access to specific http websites since the solution uses the same port on the external interface. Since most gateways are not managed, that blocking is unlikely unless the device is subscribed to a reputation like service.
REC-17	If application transparency is most important, then a stateful packet filter SHOULD have "endpoint-independent filtering" behavior for UDP. If a more stringent filtering behavior is most important, then a filter SHOULD have "address-dependent filtering" behavior. The filtering behavior MAY be an option configurable by the network administrator, and it MAY be independent of the filtering behavior for TCP and other protocols. Filtering behavior SHOULD be endpoint independent by DEFAULT in gateways intended for provisioning without service-provider management.	Similar to the REC-11 requirement but specific to UDP.
REC-33	If application transparency is most important, then a stateful packet filter SHOULD have "endpoint-independent filtering" behavior for TCP. If a more stringent filtering behavior is most important, then a filter SHOULD have "address-dependent filtering" behavior. The filtering behavior MAY be an option configurable by the network administrator, and it MAY be independent of the filtering behavior for UDP and other protocols. Filtering behavior SHOULD be endpoint independent by DEFAULT in gateways intended for provisioning without service-provider management.	Similar to the REC-11 requirement but specific to TCP.
REC-47	Valid sequences of packets bearing Shim6 payload extension headers in their outer IP extension header chains MUST be forwarded for all outbound and explicitly permitted flows. The content of the Shim6 payload extension header MAY be ignored for the purpose of state tracking.	

F.4 BCP Recommendations

The following [RFC 6092] recommendations are security best practices but are not critical to network communication. They may be supported as security requirements by eRouter devices, but are not deemed mandatory. These requirements should only be developed and tested after all requirements listed as critical (Section F.2) and important (Section F.3) have been implemented.

Table F–3 - BCP Recommendations

REC #	RFC 6092 Recommendation Text	Comments
REC-15	A state record for a UDP flow where one or both of the source and destination ports are in the well-known port range (ports 0-1023) MAY expire after a period of idle time shorter than two minutes to facilitate the operation of the IANA- registered service assigned to the port in question.	This supports SIP, SKINNY, FTP or other parsers typically found in firewalls. By watching the control traffic, they can close a session early.
REC-20	UDP-Lite flows [RFC 3828] SHOULD be handled in the same way as UDP flows, except that the upper-layer transport protocol identifier for UDP-Lite is not the same as UDP; therefore, UDP packets MUST NOT match UDP-Lite state records, and vice versa.	UDP-Lite is an uncommon protocol and further implications may exist.
REC-26	In their DEFAULT operating mode, IPv6 gateways MUST NOT prohibit the forwarding of packets, to and from legitimate node addresses, with destination extension headers of type "Host Identity Protocol (HIP)" [RFC 5201] in their outer IP extension header chain.	Not currently a significant protocol, category approaches experimental.
REC-27	The state records for flows initiated by outbound packets that bear a Home Address destination option [RFC 3775] are distinguished by the addition of the home address of the flow as well as the interior care-of address. IPv6 gateways MUST NOT prohibit the forwarding of any inbound packets bearing type 2 routing headers, which otherwise match a flow state record, and where A) the address in the destination field of the IPv6 header matches the interior care-of address of the flow, and B) the Home Address field in the Type 2 Routing Header matches the home address of the flow.	This will be needed to support IPv6 mobility but its use case in home is not clear at this time.
REC-28	Valid sequences of Mobility Header [RFC 3775] packets MUST be forwarded for all outbound and explicitly permitted inbound Mobility Header flows.	
REC-29	If a gateway forwards a Mobility Header [RFC 3775] flow, then it MUST also forward, in both directions, the IPv4 and IPv6 packets that are encapsulated in IPv6 associated with the tunnel between the home agent and the correspondent node.	
REC-30	If a gateway forwards a Mobility Header [RFC 3775] flow, then it MUST also forward (in the reverse direction) ICMPv6 "Destination Unreachable" and "Packet Too Big" messages containing any headers that match the associated flow state records.	
REC-38	All valid sequences of SCTP packets (defined in [RFC 4960]) MUST be forwarded for outbound associations and explicitly permitted inbound associations. In particular, both the normal SCTP association establishment and the simultaneous- open mode of operation MUST be supported.	If not implemented in first phase, SCTP should be dropped until this feature is implemented. Any unknown / unimplemented protocol MUST be dropped.

REC #	RFC 6092 Recommendation Text	Comments
REC-40	If a gateway cannot determine whether the endpoints of an SCTP association are active, then it MAY abandon the state record if it has been idle for some time. In such cases, the value of the "established association idle-timeout" MUST NOT be less than two hours four minutes. The value of the "transitory association idle-timeout" MUST NOT be less than four minutes. The value of the idle-timeouts MAY be configurable by the network administrator.	
REC-41	If a gateway forwards an SCTP association, it MUST also forward ICMPv6 "Destination Unreachable" and "Packet Too Big" messages containing SCTP headers that match the association state record.	
REC-42	Receipt of any sort of ICMPv6 message MUST NOT terminate the state record for an SCTP association.	
REC-43	All valid sequences of DCCP packets (defined in [RFC 4340]) MUST be forwarded for all flows to exterior servers, and for any flows to interior servers with explicitly permitted service codes.	
REC-44	A gateway MAY abandon a DCCP state record if it has been idle for some time. In such cases, the value of the "open flow idle-timeout" MUST NOT be less than two hours four minutes. The value of the "transitory flow idle- timeout" MUST NOT be less than eight minutes. The value of the idle-timeouts MAY be configurable by the network administrator.	
REC-45	If an Internet gateway forwards a DCCP flow, it MUST also forward ICMPv6 "Destination Unreachable" and "Packet Too Big" messages containing DCCP headers that match the flowstate record.	
REC-46	Receipt of any sort of ICMPv6 message MUST NOT terminate the state record for a DCCP flow.	
REC-48	Internet gateways with IPv6 simple security capabilities SHOULD implement a protocol to permit applications to solicit inbound traffic without advance knowledge of the addresses of exterior nodes with which they expect to communicate.	UPnP like functionality, but the protocol to do this reliably and the need to do this may not exist.

F.5 Other RFC 6092 Recommendations

These [RFC 6092] recommendations are not explicitly requirements for eRouter devices at this time. However, MSO consensus was reached for the incorporation of these requirements into eRouter to supplement and extend what is present in [RFC 6092]. These requirements should only be implemented after all other requirements have been satisfied.

Table F–4 - Other 6092 Recommendations[85]

REC #	RFC 6092 Recommendation Text	Comments
REC-13	Residential IPv6 gateways SHOULD provide a convenient means to update their firmware securely, for the installation of security patches and other manufacturer-recommended changes.	This requirement applies more to home routers owned by subscribers.
REC-49	Internet gateways with IPv6 simple security capabilities MUST provide an easily selected configuration option that permits a "transparent mode" of operation that forwards all unsolicited flows regardless of forwarding direction, i.e., not to use the IPv6 simple security capabilities of the gateway. The transparent mode of operation MAY be the default configuration.	The ability to turn off the firewall will probably be a requested feature but use case is still unclear as to who should be able to do this and when. The firewall MUST be on by default.
REC-50	By DEFAULT, subscriber-managed residential gateways MUST NOT offer management application services to the exterior network.	Common management application services that need to be controlled include http (tcp/80), https (tcp/443), ssh (tcp/22), telnet (tcp/23) & snmp (udp161/162). As a default setting, it is important these be disabled to prevent blocking. Unclear impact to our ability to manage CPE devices like the integrated home gateway router. All externally facing management application services support authentication and require changing of all default credentials. All externally facing management application services also support restricting access to trusted IP blocks via an ACL. ACL(s) block both IPv4 and IPv6 by default unless explicitly allowed.

F.6 RFC 6092 Recommendations In Conflict With MSO Needs

The remaining recommendations from [RFC 6092] have been found to conflict with existing or proposed MSO requirements and should not be included in eRouter devices without explicit MSO approved modifications to render them useful. Such requirements should be interpreted to be "MUST NOT" to avoid such conflicts with MSO security policies.

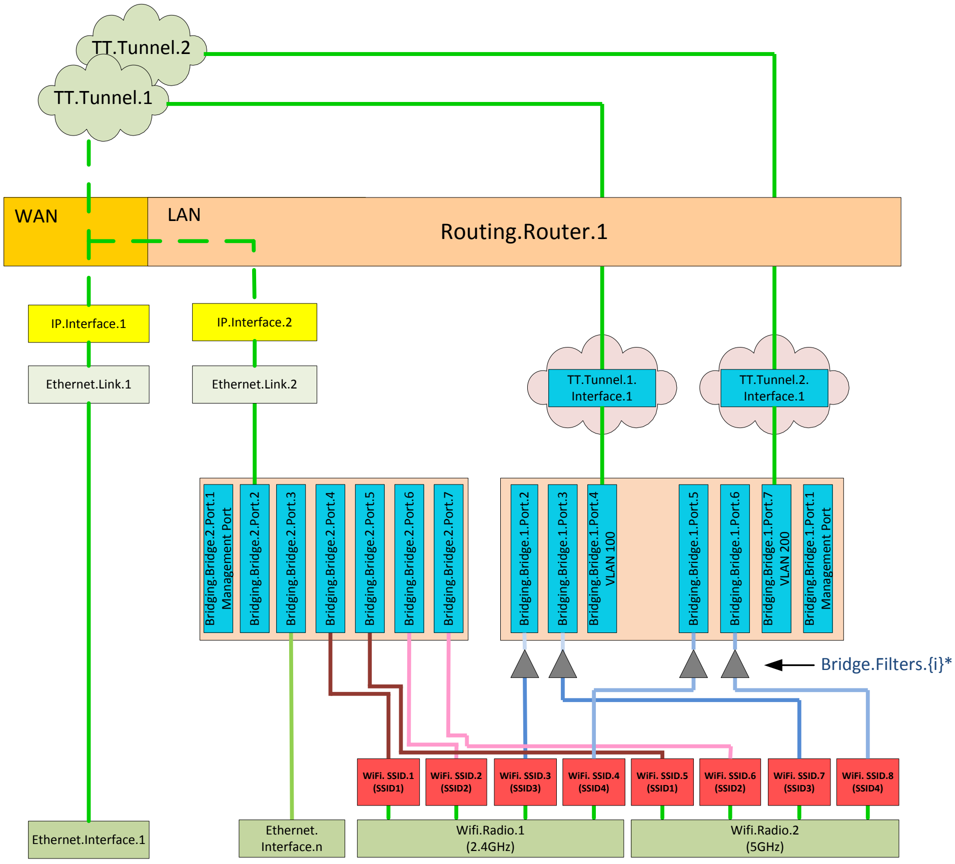
[85] Revised per eRouter-N-14.1233-1 on 2/13/15 by JB.

Table F–5 - RFC 6092 Recommendations In Conflict With MSO Needs

REC #	RFC 6092 Recommendation Text	Comments
REC-34	By DEFAULT, a gateway MUST respond with an ICMPv6 "Destination Unreachable" error code 1 (Communication with destination administratively prohibited), to any unsolicited inbound SYN packet after waiting at least 6 seconds without first forwarding the associated outbound SYN or SYN/ACK from the interior peer.	Preference would be to silently drop unsolicited packets from external sources rather than generate ICMPv6 unreachable due to administratively prohibited packets.
REC-39	By DEFAULT, a gateway MUST respond with an ICMPv6 "Destination Unreachable" error code 1 (Communication with destination administratively prohibited) to any unsolicited inbound INIT packet after waiting at least 6 seconds without first forwarding the associated outbound INIT from the interior peer.	Similar to syn dropping / errors. Prefer to silent drop instead of sending ICMP for DDoS protection and bounce attack protection.

Annex G eRouter GRE Tunneling Architecture[86]

Figure G–1 depicts a GRE forwarding model using one tunnel interface. It is included here to depict the data objects and indexing for mapping the interface, bridge ports and SSID when packets traverse the tunnel. The diagram differentiates between traffic via a private and public SSID. The configuration applies whether the configuration mechanism is SNMP or TR-069. For the purposes of this use case, it is presumed that provisioning of private and public SSIDs on the eRouter has already been completed. Data objects required for provisioning of the eRouter for private and public SSIDs is discussed elsewhere in this specification. It is also presumed that SSIDs 1 and 2 are public and SSIDs 3 and 4 are private.



*Note: Classify and assign VLAN Tags for traffic separation.

Figure G–1 - eRouter GRE Tunneling Architecture

The following table depicts the physical and logical interfaces, bridges and SSIDs that are referenced in Figure G–1 above.

Table G–1 - IF Indices and Row Instances for Data Objects Associated with GRE Tunneling

Row/Instance	Higher Layer Interface	if Index	Lower Layer	if Index
1	IP.Interface.1	300	Ethernet.Link.1	
2	Ethernet.Link.1		Ethernet.Interface.1	1
3	IP.Interface.2	200	Ethernet.Link.2	
4	Ethernet.Link.2		Bridging.Bridge.1.Port.4	

[86] Revised per eRouter-N-14.1236-2 on 2/13/15 by JB.

Row/Instance	Higher Layer Interface	if Index	Lower Layer	if Index
5	Bridging.Bridge.1.Port.1		Bridging.Bridge.1.Port.2 , Bridging.Bridge.1.Port.3, Bridging.Bridge.1.Port.4 , Bridging.Bridge.1.Port.5 , Bridging.Bridge.1.Port.6	
6	Bridging.Bridge.1.Port.2		Ethernet.Interface.2	
7	Bridging.Bridge.1.Port.3		WiFi.SSID.1 SSID1	10001
8	Bridging.Bridge.1.Port.4		WiFi. SSID.5 SSID1	10101
9	Bridging.Bridge.1.Port.5		WiFi. SSID.2 SSID2	10002
10	Bridging.Bridge.1.Port.6		WiFi.SSID.6 SSID2	10102
11	Bridging.Bridge.2.Port.1		Bridging.Bridge.2.Port.2 Bridging.Bridge.2.Port.3 Bridging.Bridge.2.Port.4 Bridging.Bridge.2.Port.5 Bridging.Bridge.2.Port.6 Bridging.Bridge.2.Port.7	
12	Bridging.Bridge.2.Port.2		WiFi.SSID.3 SSID3	10003
13	Bridging.Bridge.2.Port.3		WiFi.SSID.7 SSID3	10103
14	Bridging.Bridge.2.Port.5		WiFi.SSID.4 SSID4	10004
15	Bridging.Bridge.2.Port.6		WiFi.SSID.8 SSID4	10104
16	Bridging.Bridge.2.Port.4		TT.Tunnel.1.Interface.1	400
17	Bridging.Bridge.2.Port.7		TT.Tunnel.2.Interface.1	401
17	WiFi.SSID.1 WiFi.SSID.3 WiFi.SSID.5 WiFi.SSID.7	10001 10003 10101 10103	WiFi.Radio.1	10000
18	WiFi.SSID.2 WiFi.SSID.4 WiFi.SSID.6 WiFi.SSID.8	10002 10004 10102 10104	WiFi.Radio.2	10100
19	Wifi.Radio.1	10000		
20	Wifi.Radio.2	10100		

G.1 Use Case for Data Traffic Flow for Both Private and Public SSIDs

An eRouter that supports both private and public SSIDs must manage the private and public traffic separately. The following narrative describes how the private and public traffic traverses the physical and logical interfaces supported by the eRouter. There are four scenarios that will be described here: Private network outbound from the LAN, Private network inbound from the WAN, Public traffic outbound from a user on a public SSID, Public traffic inbound to a user on a public SSID. Each of these traffic flows is described below.

G.1.1 Private Network Outbound From the LAN

In this scenario, a user on the private network is connected via a private SSID and is attempting to connect to an outside network via the eRouter WAN interface. The following is an example of how the data traffic would flow.

G.1.2 Private Network Inbound From the WAN

In this scenario, traffic associated with a private SSID is routed through the eRouter WAN interface. The following is an example of how the data would flow from the Operator network through the internet gateway (i.e., eCM/eRouter/AP) to the private WiFI end-user device.

Traffic enters via Ethernet.Interface.1 and Ethernet.Link.1 to IP.Interface.1 (eRouter WAN interface). From here, it is routed to IP.Interface.2 (eRouter LAN interface) and to Ethernet.Link.2. Traffic is the directed to a logical bridge (Bridging.Bridge.2.Port.1) and bridged to the private SSID (WiFi.SSID.3) where it is passed to the private user via the WiFi.Radio.1 (2.4G).

Similarly, any other private LAN traffic (e.g., Ethernet, MoCA, etc.) would travel through the same logical bridge, eRouter, etc. as the private wireless traffic—the only difference is the customer facing network interface type.

G.1.3 Community WiFi User Outbound Via Public SSID

A public user connects via WiFi.Radio.2 (5G) and associates with WiFi.SSID.2. The traffic enters a logical bridge (Bridging.Bridge.1.Port.4) and is switched to another port within the bridge (Bridging.Bridge.1.Port.2). All ingress traffic is classified based on provider provisioned packet and/or port criteria. In this example, any ingress traffic via SSID2 will be tagged with an 802.1Q service tag VLAN ID 200 and bridged to the egress port appropriate egress port. The egress port is logically connected to the GRE tunnel interface (TT.Tunnel.1.Interface.1) that, in turn, is mapped to a tunnel instance (TT.Tunnel.1). The tunnel endpoint is the eRouter WAN interface (IP.Interface.1). Traffic exits the eRouter via Ethernet.Link.1 and Ethernet.Interface.1 (eCM interface).

NOTE: This example describes a single tunnel interface for a single user. However, there can be multiple users on a single tunnel interface, or multiple interfaces, depending upon a service operator's preference for managing traffic or vendor implementation. Such differentiation of traffic management is out of scope for this use case.

G.1.4 Community Wifi User Inbound Via Public SSID

802.1Q service frames tagged with VLAN ID 200 and destined for the Community WiFi end-user enters the gateway via the eCM and is passed to the Ethernet.Interface.1 and Ethernet.Link.1 to IP.Interface.1 (eRouter WAN interface). It is then placed on the tunnel (TT.Tunnel.1) and mapped to the logical tunnel interface (TT.Tunnel.1.Interface.1). Traffic then ingresses to the logical bridge (Bridging.Bridge.1.Port.2) and switched to (Bridging.Bridge.1.Port.2) where the VLAN tag is stripped and the frame is bridged to the public SSID (WiFi.SSID.2). Traffic is then transmitted to the user device using WiFi.Radio.2 (5G).

Appendix I Acknowledgements (Informative)

On behalf of the cable industry and our member companies, CableLabs would like to thank the following individuals for their contributions to the development of this specification.

Contributor	Company Affiliation
John Berg	CableLabs
Ben Bekele	Cox
Amol Bhagwat	CableLabs
Ralph Brown	CableLabs
John Brzozowski	Comcast
Eduardo Cardona	CableLabs
Margo Dolas	Broadcom
Chris Donley	CableLabs
Ralph Droms	Cisco Systems
Alain Durand	Comcast
Toerless Eckert	Cisco Systems
Kirk Erichsen	Time Warner Cable
Doc Evans	ARRIS
Roger Fish	Broadcom
Chris Grundemann	CableLabs
Deepak Kharbanda	CableLabs
Michael Kloberdans	CableLabs
Satish Kumar	Intel
Michelle Kuska	CableLabs
Diego Mazzola	Texas Instruments
John McQueen	Broadcom
Jean-Francois Mule	CableLabs
Harsh Parandekar	Cisco Systems
Michael Patrick	Motorola
Saifur Rahman	Comcast
Lakshmi Raman	CableLabs
Ryan Ross	Juniper
Lisa Ruby	Arris
Matt Schmitt	CableLabs
Ron da Silva	Time Warner Cable
Madhu Sudan	Cisco Systems
Dan Torbet	ARRIS
Greg White	CableLabs

We would particularly like to thank Ralph Droms (Cisco) and Doc Evans (ARRIS) for their extensive contributions to the specification and also for defining the eRouter Provisioning details for DHCPv4 and DHCPv6. We would like to thank Ryan Ross (Juniper) for his detailed work on the eRouter NAPT and IPv6 data-forwarding behavior. We would like to thank Margo Dolas (Broadcom) for her work in numerous areas of the specification, including the eRouter IPv4 data forwarding functionality. We would like to thank Diego Mazzola and Deepak Kharbanda for defining the eRouter IP Multicast functionality. We would like to thank John McQueen (Broadcom) for helping define the eRouter DHCPv4 server behavior. We would like to thank Dan Torbet (ARRIS), Chris Donley (CableLabs), and Kirk Erichsen (Time Warner Cable) for their work on developing the QoS functionality. We thank Deepak Kharbanda (CableLabs) who led the IPv6 Focus Team that developed this specification. And finally, many thanks go out to all the active members of the IPv6 Focus Team who contributed to this specification.

Appendix II Revision History

II.1 Engineering Change incorporated into CM-SP-eRouter-I02-070223:

ECN Identifier	Accepted Date	Title of EC
eRouter-N-06.0352-1	1/10/2007	eRouter Multicast Examples

II.2 Engineering Change incorporated into CM-SP-eRouter-I03-070518:

ECN Identifier	Accepted Date	Title of EC
eRouter-N-06.0396-1	3/14/2007	IPv6 Provision of CPE Devices Clarifications

II.3 Engineering Change incorporated into CM-SP-eRouter-I04-100611:

ECN Identifier	Accepted Date	Title of EC
eRouter-N-09-0877-2	12/30/2009	IPv6 Router Update

II.4 Engineering Change incorporated into CM-SP-eRouter-I05-110210:

ECN Identifier	Accepted Date	Title of EC
eRouter-N-10.0974-2	01/05/2011	eRouter Lease Renewal for IPv6 Prefix stability

II.5 Engineering Change incorporated into CM-SP-eRouter-I06-110623:

ECN Identifier	Accepted Date	Title of EC
eRouter-N-11.0991-1	05/04/2011	Clarification to ICMP Destination Unreachable

II.6 Engineering Changes incorporated into CM-SP-eRouter-I07-111117:

ECN Identifier	Accepted Date	Title of EC
eRouter-N-11.1014-3	10/19/2011	Adding TR-069 support for eRouter
eRouter-N-11.1015-2	10/19/2011	Update from RFC 6204 and address DHCP release

II.7 Engineering Change incorporated into CM-SP-eRouter-I08-120329:

ECN Identifier	Accepted Date	Title of EC
eRouter-N-12.1034-1	02/29/12	Fragmenting eRouter TLV

II.8 Engineering Changes incorporated into CM-SP-eRouter-I09-130404:

ECN Identifier	Accepted Date	Title of EC	Author
eRouter-N-12.1080-4	12/12/2012	eRouter TLV 11 addition	Dolas
eRouter-N-12.1081-2	12/19/2012	Detailed Simple Security Requirements	Grundemann
eRouter-N-12.1084-2	1/9/2013	Clarify eRouter container option usage	Mudugere
eRouter-N-12.1085-2	1/9/2013	LAN Multicast Forwarding	Grundemann

ECN Identifier	Accepted Date	Title of EC	Author
eRouter-N-13.1088-2	2/20/2013	Router Advertisement Behavior in eRouter	Grundemann
eRouter-N-13.1090-5	2/27/2013	Recursive DHCPv6 PD	Grundemann
eRouter-N-13.1091-4	2/27/2013	eRouter Initialization Mode Updates	Grundemann
eRouter-N-13.1099-1	3/20/13	Correction to eRouter-N-13.1091-4	Grundemann

II.9 Engineering Changes incorporated into CM-SP-eRouter-I10-130808:

ECN Identifier	Accepted Date	Title of EC	Author
eRouter-N-13.1101-3	5/22/2013	RA Transmission Frequency	Grundemann
eRouter-N-13.1102-3	5/22/2013	I09 Corrections	Grundemann
eRouter-N-13.1108-2	7/3/2013	eRouter Reset Object	Grundemann

II.10 Engineering Change incorporated into CM-SP-eRouter-I11-131120:

ECN Identifier	Accepted Date	Title of EC	Author
eRouter-N-13.1121-1	8/21/2013	Prefix Changes	Grundemann

II.11 Engineering Changes incorporated into CM-SP-eRouter-I12-140403:

ECN Identifier	Accepted Date	Title of EC	Author
eRouter-N-13.1127-4	2/26/2014	eRouter Omnibus EC	Berg
eRouter-N-14.1132-1	2/19/2014	New Annex C for eRouter specification	Berg
eRouter-N-14.1134-2	2/26/2014	Alignment with RFC 7084	Berg

II.12 Engineering Changes incorporated into CM-SP-eRouter-I13-140729:

ECN Identifier	Accepted Date	Title of EC	Author
eRouter-N-14.1150-2	7/9/2014	eRouter DS-Lite	Kloberdans
eRouter-N-14.1152-3	7/9/2014	eRouter CER-ID	Kloberdans
eRouter-N-14.1154-2	7/9/2014	eRouter Event Messages and Appendix C	Kloberdans
eRouter-N-14.1156-2	7/9/2014	eRouter Service Discovery	Kloberdans

II.13 Engineering Changes incorporated into CM-SP-eRouter-I14-150305:

ECN Identifier	Accepted Date	Title of EC	Author
eRouter-N-14.1155-3	10/15/2014	eRouter Link-ID	Kloberdans
eRouter-N-14.1162-2	9/3/2014	Updates to Architecture diagrams to accommodate requirements for community WiFi	Berg
eRouter-N-14.1178-1	10/1/2014	Modify IPv6 Address Release Behavior	Berg
eRouter-N-14.1179-2	1/7/2015	Align eRouter specification with TR-181 issue 2 amendment 7	Berg
eRouter-N-14.1217-3	1/7/2015	Updates to Annex A to define eRouter interface numbering	Berg
eRouter-N-14.1233-1	1/21/2015	Removal of eRouter spec instances of lower-case 'must'	Kloberdans
eRouter-N-15.1234-2	2/4/2015	Modify TLV 202.11	Berg

ECN Identifier	Accepted Date	Title of EC	Author
eRouter-N-15.1235-3	2/4/2015	Add requirements to eRouter specification for GRE support and new GRE MIBs	Berg
eRouter-N-15.1236-2	2/4/2015	Add architecture diagram and use case for GRE tunneling of public SSID to eRouter specificiation	Berg