Enterprise

Software Functional Specification

UDP Relay

Author: Anthony Chow, Bruce Jones and

Ashok Kumar Rajendran

Revision: 2.12

Date: September 6, 2018

Copyright © 2018 ALE USA, Incorporated  
All Rights Reserved

ALE USA, Inc.

26801 West Agoura Road, Calabasas, CA 91301

818 880 3500

This specification contains information of a confidential and/or proprietary nature. Neither this specification nor any of the information contained herein may be reproduced, used or disclosed to or for the benefit of any other person or entity without the express written consent of ALE USA, Incorporated.

Table of Contents

[REVISION HISTORY 7](#_Toc522204809)

[1. INTRODUCTION 9](#_Toc522204810)

[1.1 Purpose 9](#_Toc522204811)

[1.2 Scope 9](#_Toc522204812)

[1.3 Referenced Documents 9](#_Toc522204813)

[1.3.1 Internal Documents 9](#_Toc522204814)

[1.3.2 External Documents 9](#_Toc522204815)

[2. Requirements Overview 10](#_Toc522204816)

[2.1 Hardware Requirements and Limitations 10](#_Toc522204817)

[2.2 Software Requirements and Limitations 11](#_Toc522204818)

[2.3 Security Requirements 11](#_Toc522204819)

[2.4 Redundancy and Hot Swap Requirements 11](#_Toc522204820)

[3. FUNCTIONAL DESCRIPTION 12](#_Toc522204821)

[3.1 Basic Operation 12](#_Toc522204822)

[3.2 Basic Architecture 13](#_Toc522204823)

[3.2.1 System Startup on the Primary CMM. 14](#_Toc522204824)

[3.2.1.1 UDP Relay CMM Service 15](#_Toc522204825)

[3.2.1.1.1 Setting up connections on the CMM 15](#_Toc522204826)

[3.2.1.1.2 Interface with the MIP Library 15](#_Toc522204827)

[3.2.1.1.3 Interface with Chassis Supervision 15](#_Toc522204828)

[3.2.1.1.4 Setting up socket with the Linux IP Stack 16](#_Toc522204829)

[3.2.1.1.5 Setting up Reactor Connection with IP CMM 16](#_Toc522204830)

[3.2.1.1.6 Setting up Reactor Connection with Source Learning 16](#_Toc522204831)

[3.2.1.1.7 Sending configuration to the IPNI task 16](#_Toc522204832)

[3.2.2 Task restart 17](#_Toc522204833)

[3.2.3 System startup on the Secondary CMM 17](#_Toc522204834)

[3.2.4 Saving runtime configuration changes 17](#_Toc522204835)

[3.2.5 VRF Support 17](#_Toc522204836)

[3.2.6 Interface with the Linux IP Stack 17](#_Toc522204837)

[3.2.7 Interface with the IPNI 18](#_Toc522204838)

[3.2.8 Operation on the NI 18](#_Toc522204839)

[3.2.9 System Takeover 20](#_Toc522204840)

[3.2.10 Functionality provided by the UDP Relay task 20](#_Toc522204841)

[3.2.10.1 DHCP Relay Agent Service 21](#_Toc522204842)

[3.2.10.1.1 Standard Mode 21](#_Toc522204843)

[3.2.10.1.2 Per VLAN Mode 21](#_Toc522204844)

[3.2.10.1.3 DHCP Relay Agent Information Option support (RFC 3046) 21](#_Toc522204845)

[3.2.10.1.4 PXE Support 21](#_Toc522204846)

[3.2.10.1.5 Obtaining an IP address for VLAN 1 on system startup 21](#_Toc522204847)

[3.2.10.2 Generic UDP Relay Service 22](#_Toc522204848)

[3.2.11 Packet Flow 24](#_Toc522204849)

[3.2.11.1 DHCP Relay Agent Service packet flow 24](#_Toc522204850)

[3.2.11.2 Generic UDP Relay Service packet flow 24](#_Toc522204851)

[3.3 Platforms suppoprted 26](#_Toc522204852)

[3.4 Management Interfaces 27](#_Toc522204853)

[3.4.1 SNMP 27](#_Toc522204854)

[3.4.2 Command Line Interface 27](#_Toc522204855)

[3.4.2.1 List of Commands 27](#_Toc522204856)

[3.4.2.2 CLI output Examples 39](#_Toc522204857)

[3.4.3 Web Based Interface 41](#_Toc522204858)

[3.4.2.3 Relevant screenshots 42](#_Toc522204859)

[3.4.2.4 7.x Changes 42](#_Toc522204860)

[4. SYSTEM IMPACT 43](#_Toc522204861)

[4.1 Existing Systems 43](#_Toc522204862)

[4.2 Performance 43](#_Toc522204863)

[4.3 Resources 43](#_Toc522204864)

[4.4 Installation 43](#_Toc522204865)

[4.5 Security Concerns 43](#_Toc522204866)

[5. Test and Verification 44](#_Toc522204867)

[5.1 Test Methodology 44](#_Toc522204868)

[5.2 Acceptance Criteria 44](#_Toc522204869)

[6. Migration Path and Future Enhancements 45](#_Toc522204870)

[7. PRODUCT RELEASE 46](#_Toc522204871)

[7.1 Release Observations 46](#_Toc522204872)

[7.2 Documentation Requirements 46](#_Toc522204873)

[Appendix A: MIB 47](#_Toc522204874)

[Appendix B: 6.x UDP Relay CLI commands Not Used in 7.x 61](#_Toc522204875)

[Appendix C: RTR 2418 for 7.2.1.R01 62](#_Toc522204876)

[Appendix D: RTR 3257 DHCP-Snooping for 8.1.1.R01 79](#_Toc522204877)

[8. DHCP Snooping 79](#_Toc522204878)

[8.1 Introduction 79](#_Toc522204879)

[8.1.1 Referenced Documents 79](#_Toc522204880)

[8.2 Requirements Overview 80](#_Toc522204881)

[8.2.1 DHCP-Snooping 80](#_Toc522204882)

[8.2.2 Ingress Source Filtering 80](#_Toc522204883)

[8.2.3 DHCP Relay Agent Information Option-82 80](#_Toc522204884)

[8.2.4 Access Guardian 81](#_Toc522204885)

[8.2.5 Hardware Requirements and Limitations 81](#_Toc522204886)

[8.2.6 Software Requirements and Limitations 81](#_Toc522204887)

[8.2.7 Security Requirements 81](#_Toc522204888)

[8.2.8 Redundancy and Hot Swap Requirements 81](#_Toc522204889)

[8.3 Functional Description 82](#_Toc522204890)

[8.3.1 DHCP Snooping 82](#_Toc522204891)

[8.3.1.1 VLAN Level DHCP Snooping 83](#_Toc522204892)

[8.3.1.2 Switch Level DHCP Snooping 84](#_Toc522204893)

[8.3.1.3 DHCP Snooping Traffic Violation Statistics 85](#_Toc522204894)

[8.3.1.4 DHCP Snooping Option-82 Policy 86](#_Toc522204895)

[8.3.2 Basic Architecture 88](#_Toc522204896)

[8.3.2.1 DHCP Snooping VLAN Table 88](#_Toc522204897)

[8.3.2.2 DHCP Snooping Port Table 88](#_Toc522204898)

[8.3.2.3 DHCP Snooping Binding Database/Table 89](#_Toc522204899)

[8.3.3 Ingress Source Filtering 91](#_Toc522204900)

[8.3.4 DHCP Option-82 93](#_Toc522204901)

[8.3.5 Access Guardian 94](#_Toc522204902)

[8.3.6 Management Interfaces 94](#_Toc522204903)

[8.3.6.1 SNMP 94](#_Toc522204904)

[8.3.6.2 Web-View 95](#_Toc522204905)

[8.3.6.3 Command Line Interface 96](#_Toc522204906)

[8.3.6.3.1 Global Commands 96](#_Toc522204907)

[8.3.6.3.2 VLAN Table Commands 99](#_Toc522204908)

[8.3.6.3.3 Port Table Commands 100](#_Toc522204909)

[8.3.6.3.4 Binding Table Commands 101](#_Toc522204910)

[8.3.6.3.5 Ingress Source Filtering Commands 101](#_Toc522204911)

[8.3.6.3.6 Show Commands 102](#_Toc522204912)

[Appendix E : UDP Relay to a Specific IP 104](#_Toc522204913)

[APPENDIX F: 7.3.4.R01 DHCP Snooping 112](#_Toc522204918)

[9. DHCP Snooping – RTR 3783 112](#_Toc522204919)

[9.1 Introduction 112](#_Toc522204920)

[9.2 Design Change 112](#_Toc522204921)

[9.3 Database file synchronization 114](#_Toc522204922)

[9.4 Bug fixes after 8.1.1.R01 114](#_Toc522204923)

[9.5 Recommendation 115](#_Toc522204924)

[9.6 Limitation 115](#_Toc522204925)

[9.7 Sample Use case to explain new Design 116](#_Toc522204926)

[Appendix G: PVLAN support for DHCP-Snooping & ISF in 8.3.1.R01 118](#_Toc522204927)

[10. DHCP Snooping 118](#_Toc522204928)

[10.1 Introduction 118](#_Toc522204929)

[10.1.1 Referenced Documents 118](#_Toc522204930)

[10.1.2 Platforms supported 118](#_Toc522204931)

[10.2 Requirements Overview 118](#_Toc522204932)

[10.2.1 DHCP-Snooping support in PVLAN 118](#_Toc522204933)

[10.2.1.1 When DHCP-Snooping is enabled/disabled on primary PVLAN then it has to propogate to all the secondary PVLANS (implicit enable) 118](#_Toc522204934)

[10.2.1.2 Don’t allow to explicitly configure/ unconfigure dhcp-snooping on Secondary PVLAN 118](#_Toc522204935)

[10.2.1.3 Whenever Primary PVLAN deletion occurs disable dhcp-snooping on primary and all secondary PVLANS 118](#_Toc522204936)

[10.2.1.4 Whenever secondary Pvlan gets added to primary PVLAN then enable the dhcp-snooping on secondary if enabled on the primary 119](#_Toc522204937)

[10.2.1.5 DHCP-snooping Binding entries for the secondary PVLAN has to get created is enabled on the primary PVLAN 119](#_Toc522204938)

[10.2.2 Ingress Source Filtering in PVLAN 119](#_Toc522204939)

[10.2.2.1 When ISF is enabled/disabled on primary PVLAN then it has to propogate to all the secondary PVLANs (impilict enable) 119](#_Toc522204940)

[10.2.2.2 Don’t allow user to configure/ unconfigure ISF on Secondary PVLAN 119](#_Toc522204941)

[10.2.2.3 Whenever primary PVLAN deletion occurs disable ISF on primary and all secondary PVLANs as well 119](#_Toc522204942)

[10.2.2.4 Whenever secondary PVLAN is addedenable ISF on secondary PVLAN if enabled on the primary vlan 119](#_Toc522204943)

[10.2.3 Hardware Requirements and Limitations 119](#_Toc522204944)

[10.2.4 Software Requirements and Limitations 119](#_Toc522204945)

[10.2.5 Security Requirements 119](#_Toc522204946)

[10.2.6 Redundancy and Hot Swap Requirements 120](#_Toc522204947)

[10.2.7 Assumptions 120](#_Toc522204948)

[10.3 Design Changes 120](#_Toc522204949)

[10.3.1 Basic Architecture 120](#_Toc522204950)

[10.3.1.1 Interface with NI modules 120](#_Toc522204951)

[10.4 Use Cases 120](#_Toc522204952)

[10.4.1 When dhcp server is connected to promiscuous port 120](#_Toc522204953)

[10.4.2 When dhcp server is connected to promiscuous port of other switch 121](#_Toc522204954)

[APPENDIX H: 8.3.1.R02 DHCP Snooping 122](#_Toc522204955)

[Appendix I: RTR 4513, 5138 New CLI command to display DHCP-Snooping Binding entry for 8.4.1.R01, Config Snapshot in 85R1 122](#_Toc522204956)

[11. DHCP Snooping Binding entry 122](#_Toc522204957)

[Appendix J: RTR 4975 Requirements and Limitations for 6560 122](#_Toc522204958)

[Appendix K: RTR 4975 Requirements and Limitations for 6465 123](#_Toc522204959)

[Appendix L: Enhanced DHCP Snooping Troubleshooting (RTR 5149, 5150 in 85R2). 124](#_Toc522204960)

[1.1 INTRODUCTION 124](#_Toc522204961)

[1.1.1 Purpose 124](#_Toc522204962)

[1.1.2 Scope 124](#_Toc522204963)

[1.1.3 Intended Audience 124](#_Toc522204964)

[1.1.4 Referenced Documents 124](#_Toc522204965)

[1.1.4.1 Internal Documents 124](#_Toc522204966)

[1.1.4.2 External Documents 125](#_Toc522204967)

[1.2 FUNCTIONAL DESCRIPTION 125](#_Toc522204968)

[1.2.1 Basic Overview 125](#_Toc522204969)

[1.3 FUNCTIONAL REQUIREMENTs. 125](#_Toc522204970)

[1.4 PLATFORM SUPPORTED 126](#_Toc522204971)

[1.5 MANAGEMENT INTERFACE 126](#_Toc522204972)

[1.5.1 Command Line Interface (CLI) 126](#_Toc522204973)

[1.5.1.1 : Clear command for DHCP snooping violation counters. 126](#_Toc522204974)

[1.5.1.2 : On demand debugging for a given port or client MAC-address. 126](#_Toc522204975)

[1.5.2 Command to display/ clear the DHCP snooping/ Relay global counters. 129](#_Toc522204976)

[1.5.3 Commands for ISF statistics. 129](#_Toc522204977)

[1.5.4 SNMP/MIBs DHCP SNOOPING CLEAR COUNTERS: 131](#_Toc522204978)

[1.5.5 SNMP TRAP FOR ISF COUNT (EVERY 1 HOUR): 132](#_Toc522204979)

[1.5.6 SNMP TRAP FOR TCAM FAILURE: 133](#_Toc522204980)

[1.6 Takeover/Reload. 134](#_Toc522204981)

[1.7 Scalability and Limitations 134](#_Toc522204982)

[1.8 Testability. 135](#_Toc522204983)

[1.8.1.1 To test whether the clear command added for clearing DHCP snooping violation counter is clearing the counters properly. 135](#_Toc522204984)

[1.8.1.2 To test whether the DHCP snooping global counters are getting displayed properly. Certain counters are for internal errors which may not get incremented in normal scenarios. Hence incrementing this counters while system testing may not be possible. 135](#_Toc522204985)

[1.8.1.3 To test whether on demand debug logs are getting populated for the specified client. The logs added for error cases may not be verifiable at system testing since the internal errors occur very rarely. 135](#_Toc522204986)

[1.8.1.4 To test whether swlog and SNMP trap is initiated for binding failure due to TCAM resource failure. 135](#_Toc522204987)

[1.9 WebView Support. 135](#_Toc522204988)

[1.10 SNMP Support. 135](#_Toc522204989)

[1.11 SYSTEM DESIGN (APIs and Data Structures Added/Modified). 135](#_Toc522204990)

[Appendix M: DHCP Relay Over IP Interface For Service 141](#_Toc522204991)

[1.1 INTRODUCTION 141](#_Toc522204992)

[1.1.1 Purpose 141](#_Toc522204993)

[1.1.2 Scope 141](#_Toc522204994)

[1.1.3 Intended Audience 141](#_Toc522204995)

[1.1.4 Referenced Documents 142](#_Toc522204996)

[1.1.4.1.1 Internal Documents 142](#_Toc522204997)

[1.1.4.1.2 External Documents 142](#_Toc522204998)

[1.2 FUNCTIONAL DESCRIPTION 142](#_Toc522204999)

[1.2.1 Basic Overview 142](#_Toc522205000)

[1.3 FUNCTIONAL REQUIREMENTs. 142](#_Toc522205001)

[1.4 PLATFORM SUPPORTED 142](#_Toc522205002)

[1.5 MANAGEMENT INTERFACE 142](#_Toc522205003)

[1.5.1 Command Line Interface (CLI) 142](#_Toc522205004)

[1.2 Scalability and Limitations 143](#_Toc522205005)

[1.3 Testability. 143](#_Toc522205006)

[1.4 WebView Support. 144](#_Toc522205007)

[1.5 SNMP Support. 145](#_Toc522205008)

[1.6 SYSTEM DESIGN (APIs and Data Structures Added/Modified). 146](#_Toc522205009)

[APPENDIX N: RTR 5241 - DHCP Snooping with CVLAN option for Option 82 ASCII support 147](#_Toc522205010)

[12. System Requirement Specifications 147](#_Toc522205011)

[12.1 System Requirements 147](#_Toc522205012)

[12.1.1 Hardware Requirements 147](#_Toc522205013)

[12.1.2 Software Requirements 147](#_Toc522205014)

[12.1.3 Platforms suppoprted 147](#_Toc522205015)

[12.2 Management Requirements 147](#_Toc522205016)

[12.2.1 MIB 147](#_Toc522205017)

[12.2.2 CLI 147](#_Toc522205018)

[12.2.2.1 CLI to configure Remote ID 147](#_Toc522205019)

[12.2.2.2 CLI to remove Circuit ID 148](#_Toc522205020)

[12.2.2.3 Modification of “show dhcp-snooping” 149](#_Toc522205021)

[12.2.2.4 Modification of “show configuration snapshot dhcp-snooping” 150](#_Toc522205022)

[12.2.3 SNMP 150](#_Toc522205023)

[12.2.4 WebView 151](#_Toc522205024)

[12.3 Impacted AOS Modules 153](#_Toc522205025)

[12.4 Use Cases 153](#_Toc522205026)

[13. FUNCTIONAL SPECIFICATIONS 154](#_Toc522205027)

[13.1 Implementation Details 154](#_Toc522205028)

[14. PACKET CAPTURES 155](#_Toc522205029)

[15. MANAGEMENT INTERFACE CROSS-REFERENCE 159](#_Toc522205030)

[16. LIMITATION 160](#_Toc522205031)

# REVISION HISTORY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Agile Revision** | **Update By** | **Revision Description** | **Release** | **Status** |
| Aug 3, 2009 | 1.0 | A. Chow | Initial creation.  Agile P/N: 011739-00 | 7.1.1.R01 | Review Draft |
| Oct 10, 2009 | 1.1 | A. Chow | Update after SFS review and discussion with software architecture group. | 7.1.1.R01 | Update |
| Mar 10, 2010 | 1.2  Agile  Rev. A  DCO1391 | A. Chow | Update packet flow. CLI example and clean up MIB. | 7.1.1.R01 | Released |
| June 28, 2010 | 1.3 | A. Chow | Option 82  Agile P/N: 011739-01 | 7.2.1.R01 | Review draft |
| Feb 1, 2011 | 1.4 | A. Chow | Option 82 | 7.2.1.R01 | Update |
| March 7, 2011 | 1.5  Agile  Rev. B  DCO1726 | A.Chow | Update based on Product Marketing’s input | 7.2.1.R01 | Released |
| April 8, 2011 | 1.6  Agile  Rev. C  DCO1954 | A. Chow | Update per PLM’s request to change the show ip udp relay statistics output format and to add a new show command to track option 82 related error per port and per vlan. | 7.2.1.R01 | Update |
| Oct 10, 2013 | 1.7  Agile Rev.  C1 | B. Jones, A. Chow | Add Appendix D RTR 3257 DHCP-Snooping for 8.1.1.R01 | 8.1.1.R01 | Update for review |
| Oct 18, 2013 | 1.8 | A. Chow | Update after SFS review | 8.1.1.R01 | Update after review |
| Jan 10, 2014 | 1.9 | A. Chow | Update after test plan review | 8.1.1.R01 |  |
| Aug 12th, 2014 | 1.7  Agile Rev. 1 | Srivatsa B. Kedilaya | Adding Appendix E for UDP Relay to a Specific IP  Agile P/N: 011739-03 | 7.4.R01 | Update |
| Jan 5, 2015 | 2.0  Agile  Rev. 1 | Ashok Kumar Rajendran | Updated Appendix F  DHCP Snooping on 7.3.4.R01 RTR 3783  Agile P/N: 011739-05 | 7.3.4.R01 | Initial draft |
| Feb 23, 2015 | 2.01  Agile  Rev. 2 | A. Chow | Minor changes: update max value for binding entries on VLAN level ISF over linkagg port. | 7.3.4.R01 | Update |
| Nov 18th 2015 | 1.8  Agile Rev. A | Ramesh Reddy M | Adding Appendix G  Updating DHCP Snooping support for PVLAN  **Agile P/N: 011739-03** | 8.3.1.R01 | Final |
| Apr 27, 2016 | 2.02  Agile  Rev. 2 | Srivatsa,  Narayanan | Minor update : Added limitation for port level ip-source-filter.  Added supported platforms | 8.3.1.R01 | Update |
| May 23, 2016 | 2.02  Agile  Rev. B | Srivatsa,  Narayanan | Minor update: increasing the limit of maximum next hop IP or VLAN for RTR - 4311 | 8.3.1.R01 | Update |
| May 27, 2016 | 2.03  Agile  Rev. 1 | Srivatsa,  Narayanan | Minor update: scalability limit of maximum next hop IP or VLAN increased further for RTR - 4311 | 8.3.1.R01 | Update |
| Sep 16,2016 | 2.04  Agile  Rev. C | Saumya Panda | Minor Update: DHCP Snooping Support in 831R02 for 9900 (RTR:4355) | 8.3.1.R02 | Release |
| Oct,27,2016 | 2.05  Agile  Rev.2 | Srivatsa,  Narayanan | Minor update: new show command to display binding table in snapshot format for RTR - 4513 | 8.4.1.R01 | Update |
| Feb,07,2017 | 2.06  Agile  Rev.2 | Srivatsa, Ramesh,  Narayanan | Minor update:  New platform support.  Change in scalability limits for 6560. | 8.4.1.R02 | Update |
| Dec,06,2017 | 2.07  Agile  Rev.2 | Srivatsa, Ramesh | Minor update:  New platform support.  Change in scalability limits for 6465 | 8.5R1 | Update |
| Mar,01, 2018 | 2.08  Agile  Rev.A | Saumya,  Ramesh,  Srivatsa | Enhanced DHCP Snooping Troubleshooting (Added Appendix L), Updated Appendix J, K for ISF support  Agile P/N: 011739-08 | 8.5.R2 | Update |
| Mar,12,2018 | 2.09  Agile  Rev.A1 | Narayanan | Minor update: Updated Appendix J, K for maximum next hop address  Agile P/N: 011739-08 | 8.5.R2 | Release |
| July 4, 2018 | 2.10  Agile  Rev.B | Pavan Iddalagi,  Srivatsa Kedilaya | Added Appendix-M  Support for DHCP Relay Over Services  Migrating to configuration on IP interface for service | 8.5.R3 | Draft |
| July,13,2018 | 2.11  Agile  Rev.A | Ramesh | Minor Update for the scalability numbers of 6465 | 8.5.R4 | Release |
| July,13,2018 | 2.11  Agile  Rev.A | Ramesh,  Saumya | Added for the cvlan option in option82 | 8.5.R4 | Initial |
| Sep 6,2018 | 2.12  Agile  Rev.A1 | Ramesh | Added Appendix p for Netboot support | 8.6.R1 | Initial |
| Oct 5,2018 | 2.13 | Ramesh | Appendix O updated | 8.5R4 | Update |
| Oct 29, 2018 | 2.15 | Ramesh | Appendix S added in the part of RTR-5335  Appendix T is updated in the part of RTR-5371 | 8.6R1 | updated |
| Nov 14,2018 | 2.16 | Ramesh | Appendix O updated  Appendix Q added, Section 8.3.3 updated for ISF scalability and Section 3.3 updated for DHCP Relay support on YUKON (6900 V72, 6900 C32). | 8.5R4 | Update |
| Nov 14,2018 | 2.17 | Pavan | Appendix-R:RTR-5372  Generic UDP Relay Over Services | 8.6R1 | Initial |
| Nov 15,2018 | 2.18 | Pavan | Appendix-U RTR-5372  IPv4 UDP/DHCP Relay commands consolidation/migration, | 8.6R1 | Initial |
| Nov 27,2018 | 2.19 | Amaresh, Pavan | Appendix-R:RTR-5434  UDP Relay  over SPB using inline routing over SPB using loopback cable over traditional routing | 8.5R4 | Update |
| Jan 23, 2019 | 2.20 | Ramesh | Appendix T is updated in the part of RTR-5371 | 8.6R1 | updated |
| Feb 15, 2019 | 2.21 | Ramesh | Parity changes in the part of RTR 5455  8.3.6.3.1 is updated for the binding timeout and the save option for the dhcp-snooping binding action CLI.  8.3.2.3 is updated for the static Bindding entry precedence over the dynamic. | 8.6R1 | updated |

# INTRODUCTION

## Purpose

This document is prepared in response to the Rushmore System Architecture Specification (SAS) 011293-00, preliminary revision. This document provides the product definition and functional specification for implementing the desired features as outlined in the SRS.

### CLI Syntax Conventions

|  |  |
| --- | --- |
| Convention | Description |
| <  > | Brackets indicate that the information requested is required |
| {   } | Braces indicates that the user is required to choose from a set of options |
| | | A vertical bar separates a set of options |
| [   ] | Square brackets indicate that the information requested is optional |

Table 1: CLI Syntax Conventions

## Scope

The scope of this document is to present the complete functional description of the **UDP Relay**product as anticipated for its initial release.

## Referenced Documents

### Internal Documents

1. Engineering Development Process. Agile™, 020004-00
2. Rushmore System Requirements Specification (SRS), 011-434-00
3. Rushmore System Architecture Specification (SAS), 011293-00
4. Rushmore Software Architecture Specification (SWSAS).

### External Documents

RFC 2131: Dynamic Host Configuration Protocol.

RFC 3046: DHCP Relay Agent Information Option.

# Requirements Overview

The requirement is to provide UDP packet Relay services on the Rushmore and TOR chassis.

Currently these 5 functionalities are supported in the 6.x UDP Relay Task:

* DHCP Relay Agent (ip helper).
  + Standard mode
  + Per VLAN mode
  + AVLAN mode
* Generic UDP Relay
* DHCP Option 82 support (RFC 3046).
* IP address assignment for VLAN 1 on system boot up.
* DHCP Snooping.

For release 7.1.1.R01 the following is supported:

* DHCP Relay Agent (ip helper).
  + Standard mode
  + Per VLAN mode
* Generic UDP Relay
* DHCP Relay Agent Information Option support (RFC 3046).
* IP address assignment for VLAN 1 on system boot up.

The following functionalities are NOT supported in 7.1.1.R01 and 7.2.1.R01:

* **DHCP Relay Agent AVLAN mode,**
* **DHCP Snooping.** (This feature is being added as part of 8.1.1.R01 release. See Appendix D: RTR 3257 DHCP-Snooping for 8.1.1.R01 for details.)

## Hardware Requirements and Limitations

There is no special hardware requirement for this feature. FFP may be required to trap the UDP Relay packet to the CPU. As of the submission of this document for approval, it has not been decided if we are allowing all broadcast packet to go into the NI CPU. If this is the case FFP is not required. This, however, will not affect the over all behavior of UDP Relay in 7.X.

## Software Requirements and Limitations

* UDP Relay relies on the IP stack to send the DHCP packet to the next hop destination. There must be an IP interface configured for the VLAN in order for the device on the VLAN to use the DHCP Relay functionality.
* Generic UDP Relay also requires the VLAN that the packet comes in from and the VLAN to be forwarded to have an ip interface configured.
* UDP Relay task needs to be VRF aware.
* Maximum number of next hop IP address or VLAN/next hop pair is**1536 [RTR 4311]**.
* Maximum number of Generic UDP Relay Service is **30**.

## Security Requirements

No special security requirement for this feature. This will be the same as the security requirement for 6.x.

## Redundancy and Hot Swap Requirements

The requirement is the same as the Rushmore and TOR project as a whole.

# FUNCTIONAL DESCRIPTION

## Basic Operation

This feature provides 2 services to the network users. The first one is the **DHCP Relay Agent Service** and the second one is the **Generic UDP Relay Service**.

***DHCP Relay Agent Service*** is to allow end station to dynamically obtain an IP address from the DHCP server that resides on a different VLAN.

***Generic UDP Relay Service*** is to forward broadcast packet with pre-configured destination UDP port to destination VLAN or VLANS.

## Basic Architecture

This section describes how the UDP Relay feature fits in with the overall system and interacts with it.

Basically, the UDP Relay feature consists of one task on the CMM (UDP Relay Service) with the support of the IP NI task and the IP CMM task.

**UDP Relay Service**

This task is the forwarding engine of this UDP Relay feature for the DHCP Relay Agent Service and the generic UDP Relay Service. Based on the VLAN that the packet is coming in from and the configured IP destination or VLANs, this task will forward the data packet accordingly.

This task is to setup connections to Chassis Supervision, the MIP Library, IP CMM, Linux IP Stack.

***TBD(This is needed only if we are blocking all broadcast to the NI CPU)***: UDP Relay needs to send QOS manager a list of UDP ports to be programmed to the FFP so that UDP packet of interest can be trapped to CPU for processing. To avoid burning un-necessary FFP the UDP ports are not VRF specific. For example if VRF 0 has configured to forward UDP port 123 and 456 to VLAN 10 and VLAN 20. Packets with UDP port 123 or port 456 that comes in from VLANs that belongs to VRF 1 will also be trapped by the FFP to the CPU. Software will have to consolidate the UDP list before sending to QOS manager.

Interface is kept simple with one UDP Relay task setting up connections to IPNI task across all the max number of NIs. There could be up to 64 VRFs and if all has DHCP Relay or Generic UDP Relay configured then there will be (64 \* max number of Nis) connections established between the CMM and the NIs.



### System Startup on the Primary CMM.

On system startup Chassis Supervision is to spawn the ***UDP Relay CMM task***.

#### UDP Relay CMM Service

This task is the forwarding engine of the DHCP Relay Agent Service. It is also responsible to forwarding the Generic UDP Relay traffic to the configured destination VLANs.

##### Setting up connections on the CMM

On system startup, the UDP Relay Service task on the primary CMM will setup Reactor connection with the following tasks:

* MIP library (implicitly via the MIP registration library call)
* Chassis Supervision.
* IP CMM.
* Linux IP Stack.
* Source Learning

##### Interface with the MIP Library

On system startup, the UDP Relay CMM task will register itself to the MIP Library. This will implicitly sets up connection with the MIP Library. On the MIP registration message the UDP Relay Service is to pass the VRF, application ID and the list of MIB tables that UDP Relay is interested in.

UDP Relay is to send the registration multiple times depending on the number of VRFs configured on the system.

UDP Relay Service is to send a “MIP\_GET\_CONFIG” message to pull in the configuration for both the DHCP Relay Agent Service and the Generic UDP Relay Service for each VRF when VRF\_ADD message is received from Chassis Supervision task.

At this time it has not be decided that if there will be a single socket with the MIP Library or on socket per VRF with the MIP Library. The idea does not change and it is for UDP Relay Service to obtain configuration and to process the show command request on a per VRF basis.

##### Interface with Chassis Supervision

On system startup, the UDP Relay CMM task is to establish connection to Chassis Supervision and to send the CSLIB\_UNBLOCK message to the Chassis Supervision with the “Task Ready” status.

This task is to subscribe to Chassis Supervision the following events:

* NI\_UP
* NI\_DOWN
* VRF Creation
* VRF Deleted
* Takeover

##### Setting up socket with the Linux IP Stack

UDP Relay Service is to listen to UDP port 67 and 68 on each VRF that DHCP Relay Agent is configured with the IP Address **IPADDR\_ANY** 0.0.0.0.

If PXE is enable on a VRF, additional sockets are created based on the number of ip interface defined on that VRF. Each socket will be bound to the IP Address of the ip interface.

##### Setting up Reactor Connection with IP CMM

UDP Relay task is to subscribe to IP CMM on ip interface creation and deletion.

To support PXE, UDP Relay needs to open sockets and bind the socket to the IP Address of the ip interfaces of a particular VRF. UDP Relay is to connect to appid = ***APPID\_IP*** and snapid ***= SNAPID\_IPCMM*** and sends a ***IPCMM\_API\_IF\_VRF\_ADD*** message with vrf = -1 (all VRFs) to receive ip interface creation and deletion event so that UDP Relay task can manage the sockets created for PXE.

Based on configuration, if “Obtaining an IP address for VLAN 1 on System Startup” is enable, UDP Relay is to inform IP CMM the IP address for VLAN 1 as if user will normally configured an IP interface for VLAN 1.

##### Setting up Reactor Connection with Source Learning

This connection is also optional. These are needed only when the IP Address on boot-up for VLAN 1 is enabled.

The connection is to allow UDP Relay to temporarily add the MAC address to the L2 table when VLAN is trying to acquire an IP address from the DHCP Server on VLAN 1.

Setting the MAC address on the L2 table is to allow the DHCP reply packet to be trapped to the NI CPU and to be sent to UDP Relay task on the CMM via IPNI.

##### Sending configuration to the IPNI task

When IPNI is started on the NI, it will establish a connection to the UDP Relay CMM task on the CMM. When this connection is established, UDP Relay CMM task is to send down the list of UDP ports that it is interested in so that IPNI can filter the IP packets and sent the DHCP packets and UPD packets to the UDP Relay task for forwarding.

The IP NI task is also responsible for flooding the Generic UDP Relay packet to destination VLANs as well as sending the DHCP reply packet to the DHCP clients.

### Task restart

There is no special requirement for task restart. When a UDP Relay Service task is terminated, Chassis Supervision will restart the task. On task restart, it will perform the about operation as if on a system startup.

### System startup on the Secondary CMM

System startup on the secondary CMM is similar to the system startup on the primary CMM. Connections to the tasks on the CMM are established.

The UDP Relay CMM task is to perform the MIP registration and to send the **GET\_MIB\_CONFIG** message to pull the configuration from the boot.cfg file just like its counterpart on the primary CMM.

The UDP Relay CMM task is then started in the “waiting” state waiting for the takeover event from Chassis Supervision.

During run time if the configuration is changed dynamically, the changed configuration is synchronized to the secondary CMM. The Configuration Manager on the secondary CMM will then push the new configuration to the application. The UDP Relay Service is to listen to events from the MIP layer.

### Saving runtime configuration changes

When UDP Relay CMM received the **MIP\_GET\_ASCII** message from the Configuration Manager, it is going to write the running configuration to the Configuration Manager.

### VRF Support

VRF Support UDP Relay in 7.X will continue to be VRF aware. User is able to configure the DHCP Relay Agent Service and the Generic UDP Relay Service on a per VRF basis.

### Interface with the Linux IP Stack

The DHCP Relay Agent Service relies on the IP stack to forward the upstream DHCP packet to the next hop IP address.

Generic UDP Relay Service does not need to interface with the native Linux IP stack.

This operation is done for each VRF instance.

### Interface with the IPNI

IPNI acts as the filter for UDP Relay. The Packet Driver is to send all IP packets to IPNI. With the list given from UDP Relay, IPNI sends the UDP packets directly to UDP Relay CMM for processing.

IPNI is responsible for forwarding Generic UPD Relay packets to the destination VLANS as well as to forward the DHCP replay packet back to the DHCP client on the port that it is connected.

### Operation on the NI

All broadcast packets are to be sent to the NI CPU, the packet driver will inform IPNI that packet is available for processing.

The packet driver is to send all IP packets to IPNI. Based on the UDP port list IPNI is to forward the IP packet with the configured UDP port to UDP Relay CMM task for processing.

Packets processed by the UDP Relay CMM task is to be sent to the DHCP server via the IP stack on the CMM or to be sent to the client trying to acquire an IP address or to the destination VLAN via the IP NI task.



### System Takeover

All the tasks on the secondary CMM are started in the same way as in the primary CMM. Connections to all the tasks on the CMMs are made. Configuration is applied both on the primary and secondary CMM.

On a system takeover, Chassis Supervision on the new primary CMM is to inform all the tasks on the new primary which are in the waiting state that a system takeover event occurred.

UDP Relay CMM task is to establish new connections to the Nis for the configuration path.

When the old CMM is reset, the connections to the NIs are disconnected. NIs will re-establish connects with the new CMMs.

On a system takeover, NI blades are unaffected in the Rushmore system.

### Functionality provided by the UDP Relay task

The UDP Relay task in 7.x provides 2 kinds of services:

* ***DHCP Relay Agent Service***: Support of ***RFC 2131***.
* ***Generic UDP Relay Service***: Forwarding broadcast packets to destination VLANs.

Both of these services are for UDP packets. The operation or the forwarding decision is different between the 2 services. The destination for DHCP Relay Agent Service is a set of specific IP destinations while the destinations for the Generic UDP Relay Service are for a set of destination VLANS.

The DHCP Relay Agent Service is to support the BOOTP/DHCP protocol, specifically UDP port 67 and 68.

For UDP port 67 and 68, BOOTP/DHCP Relay Service and Generic UDP Relay Service are mutually exclusive. User can only configure either ip helper or ip udp relay service BOOTP but not both at the same time.

When configuring the “Per VLAN Mode” of the DHCP Relay Agent Service, the VLANs are the source VLANs.

When configuring the Generic UDP Relay Service, the VLANs are destination VLANs.

#### DHCP Relay Agent Service

This service is the support for RFC 2131. This service allows a device to obtain an IP address using the DHCP protocol. User is to specify the next hop IP address so that the DHCP Relay Agent and forward the DHCP packet from the device trying to obtain an IP address to the DHCP server. The next hop does not have to be a DHCP server.

##### Standard Mode

In the “Standard Mode”, UDP Relay in 7.x is to forward to the configured DHCP next hop regardless of which VLAN the DHCP packets are coming in from.

This is the default mode for DHCP Relay Agent.

##### Per VLAN Mode

In the “Per VLAN Mode”, all DHCP packets are forwarded to the DHCP next hop based on the VLAN the DHCP packet is coming from. In this mode, if there is not DHCP server configured for a particular VLAN, all DHCP packets that come in from the VLAN are discarded.

##### DHCP Relay Agent Information Option support (RFC 3046)

The UDP Relay task in 7.x is to support RFC 3046 to process the DHCP packet accordingly when that packet has the DHCP Relay Agent Information Option attribute.

When the policy is REPLACE, the UDP Relay Agent is to replace the Option 82 with VLAN id, ifIndex and the Base MAC address of the switch that the Relay Agent is running on.

By default, this is disabled.

##### PXE Support

This is PXE support requires the control of the source IP address to be used when 7.x sends the DHCP packet to the next hop IP address. Exiting management command (CLI/SNMP/WebView) is to control if this is being used.

To support this feature the UDP Relay Service task is to open additional sockets based no the ip interface of a VRF and bound the socket to the IP Address of the ip interface.

By default, this is disabled.

##### Obtaining an IP address for VLAN 1 on system startup

This is to obtain an IP address for VLAN 1 on system startup. This feature can be enabled and disabled but is not meant to be used dynamically when the system is already running. The default setting of this feature is DISABLED. This feature is specifically for VLAN 1 only. The DHCP server has to be on VLAN 1 and this does not use the DHCP Relay Agent on the switch.

When this feature is enabled, the switch is to sent out DHCP Discovery packet on VLAN 1 and try to reach the DHCP server that is on VLAN 1. Unlike a real DHCP client where the IP acquisition process timeout after certain time, the switch is to retry infinitely with a retry interval of one per second. If VLAN 1 is disabled or does not exit on system startup, the system will not perform the IP acquisition process.

The existing code in 6.x is not robust. I am not sure if this feature is used in the field but we don’t want to take out feature that “works” in 6.x from 7.x. We will have to limit this feature to work only on system startup and not dynamically.

Also, in 6.x there are times that when the DHCP Offer packets are not sent to the CMM. In 7.x, this may happen.

#### Generic UDP Relay Service

This service support the forwarding of broadcast packet to destination VLANs based on the destination UDP port. User is to configure the destination UDP port and the destination VLANs that the packet is to be forwarded to.

In 7.x, user can specify the UDP port either thru the predefined Well-known UDP port or user specified UDP port.

On CLI only, when one UDP port is to forward to multiple VLANs, and the VLANs are not in a range, user will have to separately configure it. For example if UDP port 1234 is to be forward to VLAN 10, VLAN 20 and VLAN 30 user will have to type in:

***ip udp relay port 1234***

***ip udp relay port 1234 vlan 10***

***ip udp relay port 1234 vlan 20***

***ip udp relay port 1234 vlan 30***

While if UDP port 1122 is to be forward to vlan 5, 6, 7 and 8, user need only to type in:

***ip udp relay port 1122 vlan 5-8***

In 6.x user can specify well-known UDP port by name instead of specific UDP port number. CLI is to continue to support this functionality. The predefined well-known ports are:

* TACACS - port 49
* DNS - port 53
* BOOTP - port 67 and 68
* TFTP - port 69
* NTP - port 123
* NBNS – port 137
* NBDD - port 138
* ***NBNS/NBDD – port 137 and 138 <- not supported in 7.x.***

The data packet with source UDP port equals to 67 or 68 is only processed by either the DHCP Relay Agent Service or the Generic UDP Relay Service and not by both.

For user convenience and with the Coronado chip set in the Falcon switch, user is able to specify NBNS/NBDD (both port 137 and 138) together. In 7.x, the internal data structure is made simple for processing efficiency; user will have to configure NBNS and NBDD separately. There is no change in functionality for NBNS and NBDD.

### Packet Flow

This packet flow assumes the end station is directly connected to the Rushmore/TOR unit. In the customer network, the Rushmore or TOR will most likely not directly connect to end stations but to edge switches. This packet flow description is described what happens when the UDP packets entered the Rushmore/TOR unit.

Also assume the end station is on **VLAN 30** on **VRF 2** with the ip interface for VLAN 30 is **10.30.0.254**.

#### DHCP Relay Agent Service packet flow

1. Device sends out DHCP Discovery packet.
2. DHCP Discovery packet is trapped to CPU by the FFP.
3. Packet Driver sends the packet to IPNI.
4. IPNI sends the packet to UDP Relay CMM based on the UDP list.
5. UDP Relay CMM received this packet from the socket opened.
6. UDP Relay CMM looks up the VRF that VLAN 30 belongs to.
7. UDP Relay CMM looks up the next hop IP address form its configuration.
8. For Standard Mode, it will be all the next hop IP addresses.
9. For Per VLAN Mode, it will look up if there is any next hop IP address configured for VLAN 30. If there is none, the packet is discarded.
10. Packet is sent to the Native Linux IP stack to be forwarded to the IP destination (the next hop IP address).
11. DHCP server reply with a DHCP Offer packet.
12. This DHCP Offer packet is sent to ip interface 10.30.0.254.
13. DHCP Offer packet is sent to the IP NI since the DHCP Server is to reply to the ip interface of the DHCP Relay Agent in Rushmore/TOR.
14. UDP Relay CMM looks up the network port that based on the destination MAC address of the DHCP Offer packet.
15. UDP Relay CMM looks up the NI that the network port resides.
16. UDP Relay CMM sends the DHCP Offer packet to the IP NI where the network port is.
17. IP NI sends the frame to Packet Driver where the packet is sent out into the network.
18. All upstream DHCP packets from the device to the DHCP server follow the DHCP Discovery packet.
19. All downstream DHCP packets fromthe DHCP server to the device follow the DHCP Offer packet.

#### Generic UDP Relay Service packet flow

1. Device sends out a broadcast UDP frame with destination port 1122.
2. Packet is trapped to CPU based on the destination UDP port.
3. Packet is sent to IPNI from Packet Driver.
4. IPNI sends the packet to UDP Relay CMM.
5. UDP Relay CMM looks up the VRF of VLAN 30.
6. UDP Relay CMM looks up the configuration for the UDP port 1122 and determined that this packet is to be forwarded to VLAN 40, 50 and 60.
7. UDP Relay CMM will send the packet to IP NI and then to the Packet Driver 3 times (one for VLAN 40, one for VLAN 50 and one for VLAN 60).
8. Packet Driver is able to flood the packet to all the ports that belongs to VLAN 40, VLAN 50 and VLAN 60.

## Platforms suppoprted

This SFS is for the Rushmore (OS10K) TOR (OS6900), Medora (OS9900), Shasta (OS6860), Everest (OS6865), YUKON (6900 V72, 6900 C32) platforms

## Management Interfaces

### SNMP

There is no trap for the UDP Relay task.

MIB is included in Appendix A.

### Command Line Interface

**There are 5 CLI changes in 7.x for UDP Relay:**

* All CLI commands are now applicable for all VRF.
* **ip helper vlan <vlan number> address <IP Address>** makes more sense than ip helper address <IP Address> vlan <vlan number>
* **show ip udp relay** and **show ip udp relay statistic** can have 3 optional parameters.
  + If no parameter is passed, all the ports are shown.
  + Users can choice to use pre-defined well-know UDP services or
  + UDP port number to select individual record.

In 6.x the option is to specify well-known service name, port number or <cr>. In 7.x, we are to add keyword “**service**” and “**port**” so that it is clearer to the user. For example: ***show ip udp relay TFTP*** will be changed to **show ip udp relay service TFTP**or**show ip udp relay port 69**.

* **show ip udp relay destination** is taken out because **show ip udp relay services** displays the same information and if no vlan is configured for the destination, the vlan column will be empty*.*
* **There are now 4 ways to reset ip helper statistics**.
  + **All statistics**
  + **Global statistics only**
  + **All server statistics**
  + **Specific server related statistics**

#### List of Commands

The following CLI commands are supported and are VRF specific.

***DHCP Relay Agent Service***

show ip helper

ip helper < standard | per-vlan >

ip helper address 172.6.5.1 vlan1

no ip helper address 172.6.5.1 🡨 *standard mode*

no ip helper address vlan 20 address 172.6.20.1 🡨 *per VLAN mode*.

ip helper forward delay

ip helper maximum hops

ip helper address 210.10.1.100 162.3.5.4

show ip helper statistics

no ip helper statistics <cr>

no ip helper statistics global-only

no ip helper statistics server-only

no ip helper statistics address <ipv4addr> 🡨 *standard mode*

no ip helper statistics vlan <vlan id> address <ipv4addr> 🡨 *per vlan mode*

ip helper agent-information <enable | disable>

ip helper agent-information <replace | keep | drop>

ip helper boot-up <enable | disable>

ip helper PXE-support <enable | disable>

***Generic UDP Relay Service***

ip udp relay {**service <**name> | **port** < number>}

ip udp relay {**service <**name> | **port** < number>} vlan [number]

ip udp relay no {**service <**name> | **port** < number>}

ip udp relay {**service <**name> | **port** < number>} no vlan [number }

show ip udp relay statistics [**service** <name>| **port** [<number>]

show ip udp relay [**service** <name>| **port** [<number>]

ip udp relay no statistics

***\*service name***: well know UDP port by its protocol name such as NTP, DNS. See section 3.2.12.2 for the supported well know service name supported.

**show ip udp relay service TFTP**or**show ip udp relay port 69**has the same result.

In 7.1.1R01, it is still required just like in the 6.x, to create the UDP port to forward either by well known service name or UDP port and then the destination VLANS or VLAN range.

**ip udp relay no {service <name> | port <number>}**deletes the entire associated destination VLANs configuration for that service or port.

In 7.1 the term “service” means a Generic UDP Relay Service which will be translated to an UDP port***.***

For the destination VLAN both CLI and WebView can specify a range. SNMP user will have to configure the VLAN one at a time.

***DHCP Relay Agent Service related CLI commands:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ip helper mode <standard | per-vlan-only >*** | | | | | |
| **Object ID:** iphelperForwardOption | | | |  | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | | Description |
| mode | Int32 | 1..2 | mip\_iphelperForwardOption | | This controls the mode of operation of the DHCP Relay Agent Service. |
| **Dependencies:**  N/A. | | | | | |
| **Description:** This command specifies the mode of operation for the DHCP Relay Agent Service. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ip helper vlan <vlan>address <IP Address>*** | | | | | |
| **Table ID:** iphelperTable | | | | iphelperNextHopIpAddress, iphelperVlan | |
| **Object ID:** iphelperNextHopIpAddress | | |  | | |
| **Object ID:** iphelperVlan | | |  | | |
|  | | |  | | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | | Description |
| VLAN id | Int32 | 1..4096 | mip\_ iphelperVlan | | VLAN id. Meaningful only when **iphelperForwardOption** is per-vlan mode. |
| IP Address | IP address | Valid IP address | mip\_ iphelperNextHopIpAddress | | The host IP address of the next hop. |
| **Dependencies:**  iphelperForwardOption | | | | | |
| **Description:** This command specifies the next hop IP address for the DHCP Relay Agent Service. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ip helper forward-delay <value>*** | | | | | |
| **Object ID:** iphelperForwardDelay | | | |  | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | | Description |
| value | Int32 | 1..65535 | mip\_iphelperForwardDelay | | The value to be used by the DHCP Relay Agent Service to check against the “**secs**” field in the DHCP frame. |
| **Dependencies:**  N/A. | | | | | |
| **Description:** This command specifies the “Forward Delay” for the DHCP Relay Agent Service. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ip helper max-hop <value>*** | | | | | |
| **Object ID:** iphelperMaxHop | | | |  | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | | Description |
| value | Int32 | 1..65535 | mip\_iphelperMaxHop | | The value to be used by the DHCP Relay Agent Service to check against the maximum hop the DHCP packet can travel before it is discarded. |
| **Dependencies:**  N/A. | | | | | |
| **Description:** This command specifies the “Max Hop” for the DHCP Relay Agent Service. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***no ip helper statistics {Global-only |Server-only| <cr>}*** | | | | | |
| **Object ID:** iphelperResetAllStats | | | |  | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | | Description |
| value | Int32 | 0..3 | mip\_iphelperResetAllStats | | The value when set to one (1) will reset all the DHCP Relay Agent related statistics  Two (2) reset all the Global statistics  Three (3) reset all the server statistics. |
| **Dependencies:**  N/A. | | | | | |
| **Description:** This command resets all the DHCP Relay Agent Relay Service related statistics. | | | | | |
| **no ip helper statistics vlan <vlan id> address <ipv4addr>** | | | | | |
| **Object ID:** iphelperResetSrvStats | | | |  | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | | Description |
| value | Int32 | 0..1 | mip\_iphelperResetSrvStats | | The value when set to one (1) will reset all the DHCP Relay Agent related statistics |
| **Dependencies:**  N/A. | | | | | |
| **Description:** This command resets all the statistics for the specific server/next hop IP address. | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***ip helper information-agent <enable | disable>*** | | | | |
| **Object ID:** iphelperAgentInformation | | | |  |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| value | Int32 | 1..2 | mip\_iphelperAgentInformation | The value turns on and off the Relay Agent Information (Option 82) according to RFC 3046 |
| **Dependencies:**  N/A. | | | | |
| **Description:** This command turns on and off the Relay Agent Information (Option 82) processing. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***ip helper information-agent policy <replace | keep |drop>*** | | | | |
| **Object ID:** iphelperAgentInformationPolicy | | | |  |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| value | Int32 | 1..3 | mip\_iphelperAgentInformationPolicy | The value defines the policy to use when the DHCP Relay Agent received a DHCP packet with Option 82 attribute. |
| **Dependencies:**  iphelperAgentInformation | | | | |
| **Description:** This command turns on and off the Relay Agent Information (Option 82) processing. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***ip helper boot-up <enable | disable>*** | | | | |
| **Object ID:** iphelperBootupOption | | | |  |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| value | Int32 | 1..2 | mip\_iphelperBootupOption | The value turns on and off the getting the IP address for VLAN 1 on system startup. |
| **Dependencies:**  N/A. | | | | |
| **Description:** This command turns on and off the feature getting IP address for VLAN 1 on system startup. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***ip helper boot-up pkt-format {bootp | dhcp }*** | | | | |
| **Object ID:** iphelperBootupPacketOption | | | |  |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| value | Int32 | 1..2 | mip\_iphelperBootupPacketOption | The value defines the packet format (bootp or dhcp) to be used for obtaining an IP address for VLAN 1. |
| **Dependencies:**  iphelperBootupOption. | | | | |
| **Description:** This command controls the type of packet used (bootp or dhcp) to obtain the IP address for VLAN 1. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***ip helper PXE-support <enable | disable>*** | | | | |
| **Object ID:** iphelperPXESupport | | | |  |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| value | Int32 | 1..2 | mip\_iphelperPXESupport | The value turns on and off the PXE support for the DHCP Relay Agent. |
| **Dependencies:**  N/A. | | | | |
| **Description:** This command turns on and off the PXE support for the DHCP Relay Agent. | | | | |

***Generic UDP Relay Service related CLI commands:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***ip udp relay [service <service-name>|port <UDP port number> vlan<VLAN id>*** | | | | |
| **Table ID:** genericUdpRelayTable | | | genericUdpRelayUDPport | |
| **Object ID:** genericUdpRelayUDPport | | |  | |
| **Object ID:** genericUdpRelayVlan | | |  | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| UDP port number. | Int32 | 1..65535 | mip\_genericUdpRelayUDPport | UDP port. |
| VLAN id. | Int32 | 1..4096 | mip\_genericUdpRelayVlan | Destination VLAN that the packet with genericUdpRelayUDPport is forwarded to. |
| **Dependencies:**  N/A. | | | | |
| **Description:** This command specifies the configuration for the Generic UDP Relay Service. | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ip udp relay no statistics*** | | | | | |
| **Object ID:** genericUdpRelayStatReset | | | |  | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | | Description |
| value | Int32 | 0..1 | mip\_genericUdpRelayStatReset | | The value when set to one (1) will reset all the Generic UDP Relay related statistics |
| **Dependencies:**  N/A. | | | | | |
| **Description:** This command resets all the Generic UDP Relay Service related statistics. | | | | | |

***DHCP Relay Agent Service related show commands:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***> show ip helper*** | | | | |
| **Table ID:** iphelperTable | | iphelperNextHopIpAddress, iphelperVlan | | |
| **Object ID:** iphelperForwDelay | | **Object ID:** iphelperMaxHops | | |
| **Object ID:** iphelperForwardOption | | **Object ID:** iphelperAgentInformation | | |
| **Object ID:** iphelperAgentInformationPolicy | |  | | |
| CLI Display Columns | MIB Object Name | | Display Format | Description |
| Forward Delay | mip\_iphelperForwardDelay | | Number | Forward Delay. |
| Max hop count | mip\_iphelperMaxHops | | Number | Max hop count. |
| Relay Agent Information | mip\_iphelperAgentInformation | | String | Option 82 (RFC 3046) enabled or disabled. |
| Relay Agent Information Policy | mip\_iphelperAgentInformationPolicy | | String | When Relay Agent Information is enabled this will show the policy. Either Replace or Drop or Keep. |
| Forward Option | mip\_iphelperForwardOption | | String | The forwarding option. Standard or Per-VLAN mode. |
| Vlan Number | mip\_iphelperVlan | | Number | Valid for per-vlan mode. The vlan that the Next hop IP address is associated with. |
| Next Hop IP Address | iphelperNextHopIpAddress | | IP Addr | IP address for the next hop the agent to forward the packet to. |
| **Description:** This command shows the DHCP Relay Agent Service configuration. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***> show ip helper statistics*** | | | | |
| **Table ID:** iphelperStatTable | | iphelperNextHopIpAddr | | |
| CLI Display Columns | MIB Object Name | | Display Format | Description |
| Reception From Client | iphelperRcvdFromClient | | Number | Number of packets received from clients for this DHCP Relay Agent |
| Forward Delay Violation | iphelperForwDelayViolation | | Number | Number of Forward Delay violation |
| Max Hop Violation | iphelperMaxHopViolation | | Nubmer | Number of Max hop count violation |
| Agent Information Violation | iphelperAgentInfoViolation | | Number | When Relay Agent Information is enabled this show the number of Agent Information violation |
| Invalid IP Gateway | iphelperInvalidGatewayIP | | Number | Number of Invalid Gateway IP Address. |
| Next Hop IP Address | iphelperStatNextHopIpAddr | | IP Address | IP address of the next hop |
| Number of packets transmitted to Server | iphelperStatTxToServer | | Number | Number of packet transmitted to the next hop identified by iphelperStatNextHopIpAddr |
| Invalid Agent Information from Server. | iphelperStatInvalidAgentInfoFromSvr | | Number | When Relay Agent Information is enabled this show the number of Invalid Agent information from DHCP servers as seen by this DHCP Relay Agent. |
| **Description:** This command shows the DHCP Relay Agent Service specific statistics. | | | | |

***Generic UDP Relay Service related show commands:***

The ***show ip udp relay*** and ***show ip udp relay statistics*** can have the option to show all if no parameter given or to show individual entry based on well known service name or UDP port number. Supported well known UDP service name can be found in section 3.2.12.2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***> show ip udp relay [service < service name> | port < port number>]*** | | | | |
| **Table ID:** genericUdpRelayTable | | genericUdpRelayUDPport | | |
| CLI Display Columns | MIB Object Name | | Display Format | Description |
| Service name | mip\_genericUdpRelayName | | String. | The service name. Some well know UDP ports such as NTP, NBNS or “Port 1122”. |
| UDP port | mip\_genericUdpRelayUDPport | | Number | The UDP port. |
| VLAN ID. | mip\_genericUdpRelayVlan | | Number | The VLAN id. |
| **Description:** This command shows configuration of the Generic UDP Relay Service. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***> show ip udp relay statistics [service <service name> | port <port number>]*** | | | | |
| **Table ID:** genericUdpRelayTable | | genericUdpRelayUDPport | | |
| CLI Display Columns | MIB Object Name | | Display Format | Description |
| Service name | mip\_genericUdpRelayName | | String. | The service name. Some well know UDP ports such as NTP, NBNS or “Port 1122”. |
| UDP port | mip\_genericUdpRelayUDPport | | Number | The UDP port. |
| VLAN ID. | mip\_genericUdpRelayVlan | | Number | The VLAN id. |
| Rx from client | mip\_genericUdpRelayRxFromClient | | Number | Number of packets received from clients. |
| Tx to destination. | mip\_genericUdpRelayTxToVlan | | Number | Number of packets transmitted to this VLAN. |
| **Description:** This command shows the statistics of the Generic UDP Relay Service. | | | | |

#### 

#### CLI output Examples

***Sample output of* show ip helper *- DHCP Relay Agent configured for standard mode, Relay Agent Information enabled:***

Rushmore switch> **show ip helper**

Ip helper:

Forward Delay (seconds) = 0,

Max number of hops = 4,

Relay Agent Information = Enabled,

Relay Agent Information Policy = Drop

PXE support = Disabled,

Forward option = standard

Forwarding Address:

10.145.44.254

172.0.3.1

***Sample output of* show ip helper *- DHCP Relay Agent configured for per-vlan mode, Relay Agent Information disabled:***

Rushmore switch> **show ip helper**

Ip helper:

Forward Delay (seconds) = 0,

Max number of hops = 4,

Relay Agent Information = Disabled,

PXE support = Disabled,

Forward option = per-vlan

Forwarding Address:

Vlan Number 1

Forwarding Address:

172.0.3.1

Vlan Number 11

Forwarding Address:

10.145.44.254

***Sample output of the* show ip helper statistics *CLI command:***

Rushmore switch> **show ip helper statistics**

Global Statistics:

Reception From Client:

Total Count = 0, Delta = 0,

Forward Delay Violation:

Total Count = 0, Delta = 0,

Max Hops Violation:

Total Count = 0, Delta = 0,

Agent Info Violation:

Total Count = 0, Delta = 0,

Invalid Gateway IP:

Total Count = 0, Delta = 0,

Server Specific Statistics:

Server - 10.145.44.254

Tx Server:

Total Count = 0, Delta = 0,

Invalid Agent Info From Server:

Total Count = 0, Delta = 0

Server - 172.0.3.1

Tx Server:

Total Count = 0, Delta = 0,

Invalid Agent Info From Server:

Total Count = 0, Delta = 0

***Sample output of the* show ip udp relay *CLI command:***

Rushmore switch > **show ip udp relay**

Service Name Port Vlans

---------------------+------+-----------------------------

TFTP 69 20

UDP port 1122 1122 1 20 30

***Sample output of the* show ip udp relay statistics *CLI command:*\***

Rushmore switch > **show ip udp relay statistics**

Port  Service        Pkts Recvd Pkts Sent Dst Vlan   
-----+--------------+----------+--------+----------   
 1122 UDP port 1234           0   
                                       0          2   
                                       0          3   
                                       0          4

**\***This is the best way to show how many packets are received by the switch with the specific UPD port and how many packets are sent to the various destinations VLAN. In a switch there can be 32 VLANs but for UDP port 1122, it is configured to forward to VLAN 2, 3 and 4 only. We are keeping this output format in 7.1.1.R01.

In 7.2.1.R01 per PLM’s request, the output format of this show command is changed to have packet received as the first column and then packet sent on the second column and then the destination vlan.

### 3.4.3 Web Based Interface

The MIB for this feature has changed in 7.x so the WebView layout for UDP Relay has changed as well. In 6.x we are constrained by using one data structure for both the DHCP Relay Agent and the Generic UDP Relay.

7.x UDP Relay’s Web Management interface is as follows. (WebView Path convention: Left toolbar folder button > Left toolbar icon > Menu > Sub-menu(s))

Networking > UDP Relay >

Relay (Home page)

DHCP Relay Agent > Configuration (previously Services > BOOTP/DHCP)

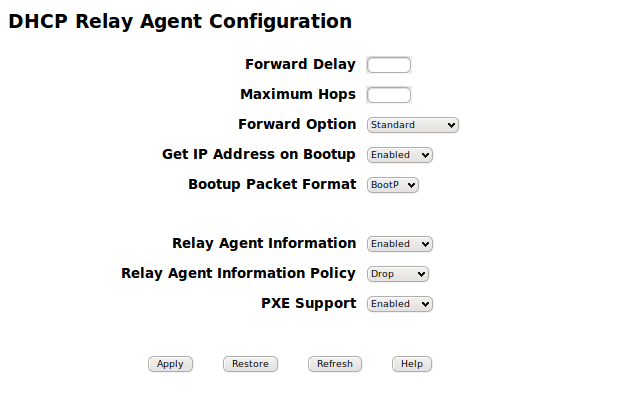
DHCP Relay Agent > Destination (previously Services > Destination)

UDP Relay Service (previously Services > Configuration)

Statistics > DHCP Relay Agent (previously Statistics > BOOTP/DHCP)

Statistics > UDP Relay Service (previously Generic Services)

#### Relevant screenshots



#### 7.x Changes

7.x UDP Relay changes to the Web Management correspond to those described in the previous SNMP section.

The entire section of DHCP Snooping has been removed from the Relay Home page as well as the menu.

# SYSTEM IMPACT

Specify details on impact of this product/feature on other areas of the system including Hardware, Software, and Management Interfaces.

## Existing Systems

7.x is built on top of a new Operation System and new hardware platform than 6.x.

## Performance

Performance for the DHCP Relay Agent and the Generic UDP Relay Service is unknown at this time. The UDP Relay task should be able to handle 500 packet/second but will not be able to know until we can test this on real hardware.

## Resources

There is no special resource requirement for this feature.

## Installation

This feature will be in the Rbase.img for the CMM.

## Security Concerns

There is no known security concern at this time.

# Test and Verification

## Test Methodology

Test plan for this feature will be in the unit test plan Intranet.

For basic CLI and show command and the initial data path, the Rushmore and TOR simulator will be used.

## Acceptance Criteria

Acceptance criteria for this feature will be to pass the unit test plan posted in the unit test plan Intranet.

# Migration Path and Future Enhancements

Migration path and future enhancements are driven by customer feed back as well as input from the Marketing department.

# PRODUCT RELEASE

## Release Observations

This feature is part of the Rushmore and TOR, Shasta, Medora and Everest release.

## Documentation Requirements

User Guide, Installation Guide, Trouble Shooting Guide, MIB documentation, is needed.

Appendix A: MIB

**We do not need a VRF specific MIB object. The implementation of VRF in 7.x will take care of this one. Individual application when registered to the MIP layer will specify the VRF that it is in. MIP layer is awake of which application is in the “VRF context”.**

ALCATEL-IND1-UDP-RELAY-MIB DEFINITIONS ::= BEGIN

IMPORTS

IpAddress,

MODULE-IDENTITY,

OBJECT-TYPE,

OBJECT-IDENTITY,

Unsigned32,

Counter32 FROM SNMPv2-SMI

MacAddress, RowStatus,

TEXTUAL-CONVENTION FROM SNMPv2-TC

SnmpAdminString FROM SNMP-FRAMEWORK-MIB

MODULE-COMPLIANCE,

OBJECT-GROUP FROM SNMPv2-CONF

routingIND1UdpRelay FROM ALCATEL-IND1-BASE

InterfaceIndex FROM IF-MIB;

alcatelIND1UDPRelayMIB MODULE-IDENTITY

LAST-UPDATED "200704030000Z"

ORGANIZATION "Alcatel -Architects Of An Internet World "

CONTACT-INFO

"Please consult with Customer Service to ensure the most appropriate

version of this document is used with the products in question:

Alcatel-Lucent, Enterprise Solutions Division

(Formerly Alcatel Internetworking, Incorporated)

26801 West Agoura Road

Agoura Hills, CA 91301-5122

United States Of America

Telephone: North America +1 800 995 2696

Latin America +1 877 919 9526

Europe +31 23 556 0100

Asia +65 394 7933

All Other +1 818 878 4507

Electronic Mail: support@ind.alcatel.com

World Wide Web: http://alcatel-lucent.com/wps/portal/enterprise

File Transfer Protocol: ftp://ftp.ind.alcatel.com/pub/products/mibs"

DESCRIPTION

"This module describes an authoritative enterprise-specific Simple

Network Management Protocol (SNMP) Management Information Base (MIB):

For the Birds Of Prey Product Line

UDP Relay to forward BOOTP/DHCP requests across VLANs

The right to make changes in specification and other information

contained in this document without prior notice is reserved.

No liability shall be assumed for any incidental, indirect, special, or

consequential damages whatsoever arising from or related to this

document or the information contained herein.

Vendors, end-users, and other interested parties are granted

non-exclusive license to use this specification in connection with

management of the products for which it is intended to be used.

Copyright (C) 1995-2007 Alcatel-Lucent

ALL RIGHTS RESERVED WORLDWIDE"

REVISION "200704030000Z"

DESCRIPTION

"The latest version of this MIB Module."

::= {routingIND1UdpRelay 1}

alcatelIND1UDPRelayMIBObjects OBJECT-IDENTITY

STATUS current

DESCRIPTION

"Branch For UDP Relay

Subsystem Managed Objects."

::= { alcatelIND1UDPRelayMIB 1 }

alcatelIND1UDPRelayMIBConformance OBJECT-IDENTITY

STATUS current

DESCRIPTION

"Branch For UDP Relay

Subsystem Conformance Information."

::= { alcatelIND1UDPRelayMIB 2 }

alcatelIND1UDPRelayMIBGroups OBJECT-IDENTITY

STATUS current

DESCRIPTION

"Branch For UDP Relay

Subsystem Units Of Conformance."

::= { alcatelIND1UDPRelayMIBConformance 1 }

alcatelIND1UDPRelayMIBCompliances OBJECT-IDENTITY

STATUS current

DESCRIPTION

"Branch For UDP Relay

Subsystem Compliance Statements."

::= { alcatelIND1UDPRelayMIBConformance 2 }

iphelperMIB OBJECT IDENTIFIER ::= { alcatelIND1UDPRelayMIBObjects 1 }

genericUdpServiceMIB OBJECT IDENTIFIER ::= { alcatelIND1UDPRelayMIBObjects 2 }

iphelperTable OBJECT-TYPE

SYNTAX SEQUENCE OF IphelperEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A list of next hop IP Address for the DHCP Relay Agent."

::= { iphelperMIB 1 }

iphelperEntry OBJECT-TYPE

SYNTAX IphelperEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

" An entry in the ip helper table"

INDEX { iphelperVlan, iphelperNextHopIpAddress }

::= { iphelperTable 1 }

IphelperEntry ::= SEQUENCE {

iphelperVlan

Unsigned32,

iphelperNextHopIpAddress

IpAddress,

iphelperResetSrvStats

Unsigned32,

iphelperRowStatus

RowStatus

}

iphelperVlan OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This object is applicable when iphelperForwardOption is

equal to perVlan (3). On Per VLAN mod the next hop

IP Address corresponds to specific source VLAN. Based on

the VLAN that the DHCP packet comes to the DHCP Relay

Agent the packet is forwarded to the next hop defined

for that VLAN. When iphelperForwardOption is standard (1),

this field will be zero (0)."

::= { iphelperEntry 1 }

iphelperNextHopIpAddress OBJECT-TYPE

SYNTAX IpAddress

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This specifies the next hop IP Address of the

DHCP Relay Agent is to forward the DHCP packet."

::= { iphelperEntry 2 }

iphelperResetSrvStats OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"When set to 1, will reset all the statistic for this

server. On standard mode vlan will be zero. After the

reset operation, system will change this back to a zero (0).

Subsystem will always return zero (0)."

::= { iphelperEntry 3 }

iphelperRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Row Status for creating/deleting"

::= { iphelperEntry 4 }

iphelperStatTable OBJECT-TYPE

SYNTAX SEQUENCE OF IphelperStatEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This keeps statistics for each next hop IP Address."

::= { iphelperMIB 2 }

iphelperStatEntry OBJECT-TYPE

SYNTAX IphelperStatEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the stat table."

INDEX { iphelperStatsVlan, iphelperStatsNextHopIpAddr}

::= { iphelperStatTable 1 }

IphelperStatEntry ::= SEQUENCE {

iphelperStatsVlan

Unsigned32,

iphelperStatsNextHopIpAddr

IpAddress,

iphelperTxToNextHop

Counter32,

iphelperInvalidAgentInfoOptFrmSrver

Counter32

}

iphelperStatsVlan OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This object is zero (0) when iphelperForwardOption is

equal to standard (1). On Per VLAN mod the next hop

IP Address corresponds to specific source VLAN. Based on

the VLAN that the DHCP packet comes to the DHCP Relay

Agent the packet is forwarded to the next hop defined

for that VLAN. When iphelperForwardOption is standard (1),

this field will be zero (0)."

::= { iphelperStatEntry 1 }

iphelperStatsNextHopIpAddr OBJECT-TYPE

SYNTAX IpAddress

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This specifies the unique server address."

::= { iphelperStatEntry 2 }

iphelperTxToNextHop OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

" This keeps track of the number of packets transmitted to the server."

::= { iphelperStatEntry 3}

iphelperInvalidAgentInfoOptFrmSrver OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This keeps track of the number of packets dropped due to

invalid from DHCP server with Relay Agent Information option

in the DHCP packet."

::= { iphelperStatEntry 4}

iphelperRxFromClient OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This keeps track of the number of packets recieved from the client."

::= { iphelperMIB 3 }

iphelperMaxHopsViolation OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This keeps track of the number of packets dropped due to

max hops violation."

::= { iphelperMIB 4}

iphelperForwDelayViolation OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This keeps track of the number of packets dropped due to

forward delay violation."

::= { iphelperMIB 5}

iphelperAgentInfoViolation OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This keeps track of the number of packets dropped due to

DHCP packet with giaddr field not equal to zero and

Relay Agent Information option is present and also the

Relay Agent Information Policy is set to DROP."

::= { iphelperMIB 6}

iphelperInvalidGatewayIP OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This keeps track of the number of packets dropped due to

giaddr matching a local subnet and Relay Agent Information

option is present in the DHCP packet."

::= { iphelperMIB 7}

iphelperForwDelay OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This sets the BOOTP/DHCP relay's forwarding delay and

is only used by the BOOTP/DHCP service. It is typically

set as seconds, but the value is totally client dependent.

This relay will not forward frames until client frames

have 'secs' field set to atleast the value iphelperForwDelay."

::= { iphelperMIB 8 }

iphelperMaxHops OBJECT-TYPE

SYNTAX INTEGER (1..16)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This sets the BOOTP/DHCP relay's maximum hops

forwarding limit and is only used by the BOOTP/DHCP service.

If a frame arrives with hopcount greater than or equal

to iphelperMaxHops, it will be dropped."

::= { iphelperMIB 9 }

iphelperForwardOption OBJECT-TYPE

SYNTAX INTEGER

{

standard(1),

perVlan(2)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This is the forwarding option for the BOOTP/DHCP Relay Agent.

Default value is standard (1)."

DEFVAL { standard }

::= { iphelperMIB 10 }

iphelperResetAllStats OBJECT-TYPE

SYNTAX INTEGER {

noOperation (0),

resetAllStats (1),

resetAllGlbStats (2),

resetAllSrvStats (3)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This parameter resets all the ip helper statistics.

1 - reset all stats. Both Global and server stats.

2 - reset all global statistics.

3 - reset all server statistics.

When the reset operation is done, subsystem will

change the value to zero (0).

Subsystem will always return zero (0)."

::= { iphelperMIB 11}

iphelperBootupOption OBJECT-TYPE

SYNTAX INTEGER

{

enable(1),

disable(2)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object allows the user to enable or disable the

functionality of the relay to get an IP address at the

time of system boot-up and assign that IP address for

the ip interface of the default VLAN. When this is diabled

and then enable, the object iphelperBootupPacketOption is

reset to its default value.

Default of this option is Disable (2)."

DEFVAL { disable }

::= {iphelperMIB 12}

iphelperBootupPacketOption OBJECT-TYPE

SYNTAX INTEGER

{

bootp(1),

dhcp(2)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object is used in conjunction with iphelperBootupOption.

This specify the packet format with the choices of BOOTP or DHCP

to be used to get an IP address at the time of system boot-up.

Default option is DHCP"

DEFVAL { dhcp }

::= {iphelperMIB 13}

iphelperAgentInformation OBJECT-TYPE

SYNTAX INTEGER

{

enable(1),

disable(2)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object allows user to enable or disable the functionality

of inserting the relay agent information option to the DHCP

option field according to RFC 3046. When is is disbled and then

enabled, the iphelperAgentInformationPolicy will be reset to its

default value."

DEFVAL { disable }

::= {iphelperMIB 14}

iphelperAgentInformationPolicy OBJECT-TYPE

SYNTAX INTEGER

{

drop(1),

keep(2),

replace(3)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object allows the user to select the policy of either

drop, keep or replace the relay agent information option if

this option is already present in the DHCP packet."

DEFVAL { drop }

::= {iphelperMIB 15}

iphelperPXESupport OBJECT-TYPE

SYNTAX INTEGER {

enabled(1),

disabled(2)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"When enabled the relay agent will replace the source IP address

of the packet with the gateway IP address from the DHCP packet.

The default value is disabled."

DEFVAL { disabled }

::= { iphelperMIB 16 }

genericUdpServiceTable OBJECT-TYPE

SYNTAX SEQUENCE OF GenericUdpServiceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table defines the Generic UDP Relay Service."

::= { genericUdpServiceMIB 1 }

genericUdpServiceEntry OBJECT-TYPE

SYNTAX GenericUdpServiceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in Generic UDP Relay table"

INDEX { genericUdpServiceUdpPort}

::= { genericUdpServiceTable 1 }

GenericUdpServiceEntry ::= SEQUENCE {

genericUdpServiceUdpPort

Unsigned32,

genericUdpServiceDescription

SnmpAdminString,

genericUdpServiceStatRxFromClient

Counter32,

genericUdpServiceRowStatus

RowStatus

}

genericUdpServiceUdpPort OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"UDP port for the service."

::= { genericUdpServiceEntry 1 }

genericUdpServiceDescription OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..30))

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The name and/or description of the service. If null string is passed

the default name will be used."

::= { genericUdpServiceEntry 2 }

genericUdpServiceStatRxFromClient OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This keeps track of the number of packets received from the

client with UDP destination port matching genericUdpServiceUdpPort."

::= { genericUdpServiceEntry 3 }

genericUdpServiceRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Row Status for creating/deleting"

::= { genericUdpServiceEntry 4 }

genericUdpServiceDstTable OBJECT-TYPE

SYNTAX SEQUENCE OF GenericUdpServiceDstEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table defines the destination VLAN for the

Generic UDP Relay Service. UDP packet with destination port

genericUdpServiceUdpPort are forwarded to VLAN defined in

genericUdpServiceDstVlan."

::= { genericUdpServiceMIB 2 }

genericUdpServiceDstEntry OBJECT-TYPE

SYNTAX GenericUdpServiceDstEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in Generic UDP Relay table"

INDEX { genericUdpServicePort, genericUdpServiceDstVlan}

::= { genericUdpServiceDstTable 1 }

GenericUdpServiceDstEntry ::= SEQUENCE {

genericUdpServicePort

Unsigned32,

genericUdpServiceDstVlan

Unsigned32,

genericUdpServiceStatTxToVlan

Counter32,

genericUdpServiceDstTblRowStatus

RowStatus

}

genericUdpServicePort OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"UDP port for the service."

::= { genericUdpServiceDstEntry 1 }

genericUdpServiceDstVlan OBJECT-TYPE

SYNTAX Unsigned32 (1..4096)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Destination VLAN that the UPD port is to be forwarded to."

::= { genericUdpServiceDstEntry 2 }

genericUdpServiceStatTxToVlan OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This keeps track of the number of packets transmitted to the

destination VLAN with UDP destination port matching

genericUdpServicePort."

::= { genericUdpServiceDstEntry 3}

genericUdpServiceDstTblRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Row Status for creating/deleting the Generic Service Destination Table"

::= { genericUdpServiceDstEntry 4 }

genericUdpServiceStatReset OBJECT-TYPE

SYNTAX INTEGER {

noOperation (0),

resetAllStats (1)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This parameter resets all the Generic UDP Relay related

statistics. Subsystem always returns zero (0)."

::= { genericUdpServiceMIB 3 }

alcatelIND1UDPRelayMIBCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

"Compliance statement for UDP Relay"

MODULE

MANDATORY-GROUPS

{

iphelperGroup,

iphelperStatGroup,

iphelperMiscGroup,

genericUdpServiceGroup,

genericUdpServiceDstGroup,

genericUdpServiceMiscGroup

}

::={ alcatelIND1UDPRelayMIBCompliances 1}

iphelperGroup OBJECT-GROUP

OBJECTS

{

iphelperResetSrvStats,

iphelperRowStatus

}

STATUS current

DESCRIPTION

" Collection of objects for the management of parameters of UDP Relay."

::= { alcatelIND1UDPRelayMIBGroups 1}

iphelperStatGroup OBJECT-GROUP

OBJECTS

{

iphelperTxToNextHop,

iphelperInvalidAgentInfoOptFrmSrver

}

STATUS current

DESCRIPTION

" Collection of objects for management of statistics for DHCP Relay Agent."

::= { alcatelIND1UDPRelayMIBGroups 2}

iphelperMiscGroup OBJECT-GROUP

OBJECTS

{

iphelperRxFromClient,

iphelperMaxHopsViolation,

iphelperForwDelayViolation,

iphelperAgentInfoViolation,

iphelperInvalidGatewayIP,

iphelperForwDelay,

iphelperMaxHops,

iphelperForwardOption,

iphelperResetAllStats,

iphelperBootupOption,

iphelperBootupPacketOption,

iphelperAgentInformation,

iphelperAgentInformationPolicy,

iphelperPXESupport

}

STATUS current

DESCRIPTION

" Other independent objects of UDP Relay."

::= { alcatelIND1UDPRelayMIBGroups 3}

genericUdpServiceGroup OBJECT-GROUP

OBJECTS

{

genericUdpServiceDescription,

genericUdpServiceStatRxFromClient,

genericUdpServiceRowStatus

}

STATUS current

DESCRIPTION

"Objects for Generic UDP Servce table"

::= { alcatelIND1UDPRelayMIBGroups 4}

genericUdpServiceDstGroup OBJECT-GROUP

OBJECTS

{

genericUdpServiceStatTxToVlan,

genericUdpServiceDstTblRowStatus

}

STATUS current

DESCRIPTION

"Objects for Generic UDP Servce Destination table"

::= { alcatelIND1UDPRelayMIBGroups 5}

genericUdpServiceDstGroup OBJECT-GROUP

OBJECTS

{

genericUdpServiceStatReset

}

STATUS current

DESCRIPTION

"Objects for Generic UDP Servce Destination table"

::= { alcatelIND1UDPRelayMIBGroups 6}

END

Appendix B: 6.x UDP Relay CLI commands Not Used in 7.x

|  |
| --- |
| ip helper avlan only |
| ip helper traffic-suppression {enable | disable} |
| ip helper dhcp-snooping {enable | disable} |
| ip helper dhcp-snooping mac-address verification {enable | disable} |
| ip helper dhcp-snooping option-82 data-insertion {enable | disable} |
| ip helper dhcp-snooping option-82 data-insertion format [base-mac | system-name | user-string string] |
| ip helper dhcp-snooping bypass option-82-check {enable | disable} |
| ip helper dhcp-snooping vlan vlan\_id [mac-address verification {enable | disable}] [option-82 data-insertion {enable | disable}] |
| no ip helper dhcp-snooping vlan vlan\_id |
| ip helper dhcp-snooping port slot1/port1[-port1a] {block | client-only | trust} |
| ip helper dhcp-snooping linkagg num {block | client-only | trust} |
| ip helper dhcp-snooping port slot1/port1[-port1a] traffic-suppression {enable | disable} |
| ip helper dhcp-snooping port slot1/port1[-port1a] ip-source-filtering {enable | disable} |
| ip helper dhcp-snooping port binding {[enable | disable] | [mac\_address port slot/port address ip\_address vlan vlan\_id]} |
| no ip helper dhcp-snooping port binding mac\_address port slot/port address ip\_address vlan vlan\_id |
| ip helper dhcp-snooping port binding timeout seconds |
| ip helper dhcp-snooping port binding action {purge | renew} |
| ip helper dhcp-snooping binding persistency {enable | disable} |
| show ip helper dhcp-snooping vlan |
| show ip helper dhcp-snooping port |
| show ip helper dhcp-snooping binding |
| show ip udp relay destination [BOOTP | NBDD | NBNSNBDD | DNS | TACACS | TFTP | NTP | port] |

Appendix C: RTR 2418 for 7.2.1.R01

This feature is to allow users to configure the format of the Option 82 field. The possible formats are

1. User string (up to 64 characters)
2. System name
3. Interface alias
4. Base MAC
5. Auto interface alias (in the format system\_name\_slot\_port)
6. ASCII

For the ASCII format it can be a combination of up to 5 of the following fields as long as the ASCII string does not exceed 128 characters:

1. VLAN
2. User string (up to 64 characters)
3. System name
4. Interface alias
5. Interface (in the format of slot/port)
6. Base MAC

User is also able to configure a delimiter that will be used between the different ASCII options. Valid characters are:

1. | (pipe)
2. \ (backward slash)
3. / (forward slash)
4. - (dash)
5. \_ (underscore) and
6. “ “ (space)

In 7.2.1.R01, the default Option-82 format is base-MAC. The default delimiter in 7.2.1.R01 is “ “ (space).

Operation of Option-82 in 7.2.1.R01 is to insert the Option 82 attribute to the DHCP packet when Relay Agent Information is enabled and based on the Relay Agent Information Policy. Instead of a fixed format in 7.1.1.R01, user is able to select what to be put into the Option 82 field. There are 2 sub-id fields for Option 82 – circuit-id and remote-id. Depending on the format type configured, the circuit-id and remote-id are constructed differently.

For format type is user string, system name, interface alias, base MAC or auto interface alias:

Circuit-id = vlan + slot + port + the configured format

Remote-id = configured format

For format type equals to ASCII:

Circuit-id = ASCII string as configured

Remote-id = Host name from DHCP option 12 if any.

In 6.X, Option 82 is inserted into the DHCP packet when

* Agent-information is enabled (based of the value of the giaddr or the presence of the Option 82 along with user configured agent information policy).
* DHCP snooping is enable and data insertion is also enabled (by default when DHCP snooping is enable data insertion is also enabled unless explicitly disabled by user).

In release 7.1.1.R01 and 7.2.1.R01, there is no DHCP snooping and Option 82 is inserted into the DHCP packet when:

* Agent-information is enabled (based of the value of the giaddr or the presence of the Option 82 along with user configured agent information policy).

Option 82 format configuration is only meaningful only when the Relay Agent Information is enabled. User can change the Option 82 format when Relay Agent Information is disabled but will have no effect. Also when user changed the Relay Agent Information from enabled to disable the Option 82 format is automatically reverted back to the default value – Base MAC. Also the option 82 related statistics are cleared when user change the Relay Agent Information from enable to disable.

The UDP Relay Agent uses the Relay Agent Information Policy to decide how to handle the DHCP packet before sending to the DHCP server when the Relay Agent Information Option is **enabled**:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Giaddr = zero** | **Option 82 present** | **Action to take (Forward or Drop)** |
| Case #1 | True | No | **Forward** the packet to the DHCP server after adding Option 82 attribute with the configured Option 82 format to the DHCP packet. |
| Case #2 | True | Yes | Based on the Relay Agent Information Policy to **keep**, **replace** or **drop** the DHCP packet. When policy is **replace** the relay agent will based on the configured Option 82 format replacing the existing Option 82 while the policy is **keep** the relay agent is to keep the existing Option 82 attribute and **forward** the packet to the DHCP server in these 2 cases. When policy is **drop**, the packet is **dropped**. |
| Case #3 | False | Yes | **Drop** the DHCP packet if the giaddr address is matching the subnet of a local ip interface. Otherwise, **forward** the packet. |
| Case #4 | False | No | **Forward** the DHCP packet without adding the Option 82 attribute to the packet. |

New MIB to support this feature:

iphelperDhcpOption82FormatType OBJECT-TYPE

SYNTAX INTEGER {

macAddress(1),

systemName(2),

userString(3),

interfaceAlias(4),

autoInterfaceAlias(5),

ascii(6)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"DHCP option-82 defines the type of information carried in circuit id

and remote id sub option fields. If the type selected is string the actual

value of the string can be found in iphelperDhcpOption82StringValue.

Format type ASCII will insert the configured fields in ASCII format."

DEFVAL { macAddress }

::= { iphelperMIB 17 }

iphelperDhcpOption82StringValue OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..63))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The value of the string that will be used in the circuit id and remote id

sub options."

::= { iphelperMIB 18 }

iphelperDhcpOption82FormatASCIIField1 OBJECT-TYPE

SYNTAX IphelpereOption82ASCIIFieldType

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The type of the first field in the Option 82 string in ASCII format,

which will be copied to Option-82 circuit id of the DHCP packet.

This Field is applicable only, if the option 82 format type is ASCII"

DEFVAL { none }

::= { iphelperMIB 19 }

iphelperDhcpOption82FormatASCIIField1StringValue OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..63))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The string value of the first field in the Option 82 string in ASCII

format, which will be copied to Option-82 circuit id of the DHCP packet.

This Field is applicable only, if the option 82 format type is ASCII"

::= { iphelperMIB 20 }

iphelperDhcpOption82FormatASCIIField2 OBJECT-TYPE

SYNTAX IphelperOption82ASCIIFieldType

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The type of the second field in the Option 82 string in ASCII format,

which will be copied to Option-82 circuit id of the DHCP packet.

This Field is applicable only, if the option 82 format type is ASCII"

DEFVAL { none }

::= { iphelperMIB 21 }

iphelperDhcpOption82FormatASCIIField2StringValue OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..63))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The string value of the second field in the Option 82 string in ASCII

format, which will be copied to Option-82 circuit id of the DHCP packet.

This Field is applicable only, if the option 82 format type is ASCII"

::= { iphelperMIB 22 }

iphelperDhcpOption82FormatASCIIField3 OBJECT-TYPE

SYNTAX IphelperOption82ASCIIFieldType

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The type of the third field in the Option 82 string in ASCII format,

which will be copied to Option-82 circuit id of the DHCP packet.

This Field is applicable only, if the option 82 format type is ASCII"

DEFVAL { none }

::= { iphelperMIB 23 }

iphelperDhcpOption82FormatASCIIField3StringValue OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..63))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The string value of the third field in the Option 82 string in ASCII

format, which will be copied to Option-82 circuit id of the DHCP packet.

This Field is applicable only, if the option 82 format type is ASCII"

::= { iphelperMIB 24 }

iphelperDhcpOption82FormatASCIIField4 OBJECT-TYPE

SYNTAX IphelperOption82ASCIIFieldType

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The type of the fourth field in the Option 82 string in ASCII format,

which will be copied to Option-82 circuit id of the DHCP packet.

This Field is applicable only, if the option 82 format type is ASCII"

DEFVAL { none }

::= { iphelperMIB 25 }

iphelperDhcpOption82FormatASCIIField4StringValue OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..63))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The string value of the fourth field in the Option 82 string in ASCII

format, which will be copied to Option-82 circuit id of the DHCP packet.

This Field is applicable only, if the option 82 format type is ASCII"

::= { iphelperMIB 26 }

iphelperDhcpOption82FormatASCIIField5 OBJECT-TYPE

SYNTAX IphelperOption82ASCIIFieldType

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The type of the fifth field in the Option 82 string in ASCII format,

which will be copied to Option-82 circuit id of the DHCP packet.

This Field is applicable only, if the option 82 format type is ASCII"

DEFVAL { none }

::= { iphelperMIB 27 }

iphelperDhcpOption82FormatASCIIField5StringValue OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE (0..63))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The string value of the fifth field in the Option 82 string in ASCII

format, which will be copied to Option-82 circuit id of the DHCP packet.

This Field is applicable only, if the option 82 format type is ASCII"

::= { iphelperMIB 28 }

iphelperDhcpOption82FormatASCIIDelimiter OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..63))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The value of the delimiter that is inserted between the fields in the

Option 82 string in ASCII format, which will be copied to Option-82

circuit id of the DHCP packet. This value is applicable only, if the

option 82 format type is ASCII"

::= { iphelperMIB 29 }

iphelperResetAllOpt82ErrStats OBJECT-TYPE

SYNTAX INTEGER {

noOperation (0),

resetAllStats (1)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This parameter resets all the ip helper Option 82 error statistics.

When the reset operation is done, subsystem will

change the value to zero (0).

Subsystem will always return zero (0)."

::= { iphelperMIB 30}

iphelperOption82ErrStatsTable OBJECT-TYPE

SYNTAX SEQUENCE OF IphelperOption82ErrStatsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A list of Option 82 related error statistic count per port and per vlan."

::= { iphelperMIB 31 }

iphelperOption82ErrStatsEntry OBJECT-TYPE

SYNTAX IphelperOption82ErrStatsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

" An entry in the ip helper table"

INDEX { iphelperOpt82ifIndex, iphelperOpt82vlan }

::= { iphelperOption82ErrStatsTable 1 }

IphelperOption82ErrStatsEntry ::= SEQUENCE {

iphelperOpt82ifIndex

InterfaceIndex,

iphelperOpt82vlan

Unsigned32,

iphelperOpt82agentInfoViolationCnt

Counter32,

iphelperOpt82invalidGatewayIPAddrCnt

Counter32,

iphelperOpt82resetErrStats

INTEGER

}

iphelperOpt82ifIndex OBJECT-TYPE

SYNTAX InterfaceIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The interface where the DHCP packet ingress the switch that

caused the Agent Info Violation or the Invalid Gateway

IP Address error."

::= { iphelperOption82ErrStatsEntry 1 }

iphelperOpt82vlan OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The vlan where the DHCP packet ingress the switch that

caused the Agent Info Violation or the Invalid Gateway

IP Address error."

::= { iphelperOption82ErrStatsEntry 2 }

iphelperOpt82agentInfoViolationCnt OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of Agent Info Violation seen on this interface and vlan."

::= { iphelperOption82ErrStatsEntry 3 }

iphelperOpt82invalidGatewayIPAddrCnt OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of Invalid Gateway IP Address seen on this interface and vlan."

::= { iphelperOption82ErrStatsEntry 4 }

iphelperOpt82resetErrStats OBJECT-TYPE

SYNTAX INTEGER {

noOperation (0),

resetErrStats (1)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This parameter resets the ip helper Option 82 error statistics

on this interface and vlan. When the reset operation is done,

subsystem will change the value to zero (0). Subsystem will always return zero (0)."

::= { iphelperOption82ErrStatsEntry 5 }

A new cli command:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ip helper option-82 format [base-mac | system-name | user-string {“string”} | interface-alias | auto-interface-alias | ASCII {vlan | user-string {“string”} | system-name | interface-alias | interface | base-mac |delimiter {“string” } }]*** | | | | | |
| **Table ID:** MIP\_IPHELPERMIB | | mip\_iphelperDhcpOption82FormatType mip\_iphelperDhcpOption82StringValue  mip\_iphelperDhcpOption82FormatASCIIField1  mip\_iphelperDhcpOption82FormatASCIIField1StringValue  mip\_iphelperDhcpOption82FormatASCIIField2  mip\_iphelperDhcpOption82FormatASCIIField2StringValue  mip\_iphelperDhcpOption82FormatASCIIField3  mip\_iphelperDhcpOption82FormatASCIIField3StringValue  mip\_iphelperDhcpOption82FormatASCIIField4  mip\_iphelperDhcpOption82FormatASCIIField4StringValue  mip\_iphelperDhcpOption82FormatASCIIField5  mip\_iphelperDhcpOption82FormatASCIIField5StringValue  mip\_iphelperDhcpOption82FormatASCIIDelimiter | | | |
|  | |  | | | |
| Input | Type | | Range | Object | Description |
|  |  | |  |  |  |
| **Dependencies:** | | | | | |
| **Description:** Used to configure the type of data to be inserted to option-82 field. | | | | | |

The CLI command to configure the Option 82 format will be:

***ip helper option-82 format …***

For example:

ip helper option-82 format base-mac

ip helper option-82 format system-name

ip helper option-82 format interface-alias

ip helper option-82 format auto-interface-alias

ip helper option-82 format user-string “TOR switch #1”

ip helper option-82 format ascii vlan base-mac interface delimiter -

ip helper option-82 format ascii vlan base-mac delimiter */*

Show command when agent information is disabled:

-> show ip helper

Ip helper :

Forward Delay(seconds) = 0,

Max number of hops = 16,

Relay Agent Information = Disabled,

PXE support = Disabled,

Forward option = standard mode

Booptup Option = Disable

Forwarding address list (Standard mode):

10.255.50.100

Show command when agent information is enabled and format is ASCII:

-> show ip helper

Ip helper :

Forward Delay(seconds) = 0,

Max number of hops = 16,

Relay Agent Information = Enabled,

Relay Agent Information Policy = Drop

DHCP option-82 Format = ASCII,

DHCP option-82 ASCII Field1 = Base MAC,

DHCP option-82 ASCII Field1 String = 00:d0:95:07:f2:1c,

DHCP option-82 ASCII Field2 = Vlan,

DHCP option-82 ASCII Field2 String = - ,

DHCP option-82 ASCII Field3 = User String,

DHCP option-82 ASCII Field3 String = TOR switch data center 1,

DHCP option-82 ASCII Field4 = Interface,

DHCP option-82 ASCII Field4 String = - ,

PXE support = Disabled,

Forward option = standard mode

Booptup Option = Disable

Forwarding address list (Standard mode):

10.255.50.100

When the option selected is for interface or interface alias or auto interface alias, the value is specific to the port where the DHCP packet egress the switch and the value will show “-“ .

Show command when agent information is enable and the format is base MAC:

-> show ip helper

Ip helper :

Forward Delay(seconds) = 0,

Max number of hops = 16,

Relay Agent Information = Enabled,

Relay Agent Information Policy = Drop

DHCP option-82 Format = Base MAC,

DHCP option-82 String = 00:d0:95:07:f2:1c,

PXE support = Disabled,

Forward option = standard mode

Booptup Option = Disable

Forwarding address list (Standard mode):

10.255.50.100

PLM requested to add a show command to show the Option 82 related error as per port, per vlan or per port per vlan so that it will be easier to troubleshoot. This is the requirement from PLM is to display these 2 Option 82 related error from the client side:

**Display** **all ports that has any error count**

show ip helper option-82 error-count <cr>

**Display all non zero entries on this interface**.

show ip helper option-82 error-count port <slot>/<port> <cr>

**Display all non zero entries on this vlan**.

show ip helper option-82 error-count vlan <vlan ID> <cr>

**Display all non zero entries on a specific vlan on this interface**

show ip helper option-82 error-count port <slot>/<port> vlan <vlan ID> <cr>

We also need 4 variations to clear the statistics:

no ip helper option-82 error-count [port <slot>/<port> | vlan <vlan ID> | port <slot>/<port> vlan <vlan ID> | <cr>]

Same as the show command, user should be able to clear the Option 82 error statistics

* per port,
* per vlan,
* per vlan on a specific port or
* all stats.

There are 2 Option 82 related error counts from the client side. In 7.1.1.R01, the statistics are shown globally only.

**Agent Info Violation Count**: increment when Relay Agent Information Policy is drop and the incoming DHCP packets has the Option 82 attribute and the giaddr address is zero.

**Invalid Gateway IP Address Count**: increment when the incoming DHCP has the Option 82 attribute and the giaddr field is non zero and the giaddr address is matching to a subnet of a local ip interface.

To accommodate this request, a new show command is introduced in 7.2.1.R01 to show the option 82 error count from the client side namely the Agent Information Violation count and the Invalid Gateway IP address count. There is also a “no” command to clear the statistics based on the input of the cli command.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***> show ip helper opton-82 error-count [port < slot>/<port> | vlan < VLAN ID> | port <slot>/<port> vlan <VLAN ID>]*** | | | | |
| **Table ID:** iphelperOption82ErrStatsTable | |  | | |
| CLI Display Columns | MIB Object Name | | Display Format | Description |
| Slot | mip\_iphelperOpt82ifIndex | | Number | Slot number |
| port | mip\_iphelperOpt82ifIndex | | Number | Port number |
| VLAN ID. | mip\_iphelperOpt82vlan | | Number | Vlan ID |
| Agent Info Violation count | mip\_iphelperOpt82agentInfoViolationCnt | | Number | Agent Information Violation Count |
| Invalid Gateway IP Addr Count | mip\_iphelperOpt82invalidGatewayIPAddrCnt | | Number | Invalid Gateway IP address count |
| **Description:** This command shows Option 82 related error statistics on a per port and per vlan basis. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***no ip helper option-82 error-count [port <slot>/<port> | vlan<VLAN id> | port <slot>/port vlan <VLAN id>]*** | | | | |
| **Table ID:** iphelperOption82ErrStatsTable | | |  | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| Slot | Int32 | 1..Max slot for the hardware platform | mip\_iphelperOpt82ifIndex | Slot |
| Port | Int32 | 1..Max port for the hardware platform | mip\_iphelperOpt82ifIndex | Port |
| VLAN id. | Int32 | 1..4096 | mip\_iphelperOpt82vlan | VLAN ID. |
| **Dependencies:**  N/A. | | | | |
| **Description:** This command clears Option 82 error stats. If only slot/port (ifIndex) is given it will clear all the stats of that slot/port. If only vlan is passed, then all the stats on that vlan in any slot/port will be cleared. If both slot/port and vlan is passed the specific slot/port and vlan is cleared, other vlans on the specified slot/port are not cleared. | | | | |

Sample output format for show ip helper option-82 error-count vlan 100:

show ip helper option-82 error-count vlan 100

Slot/Port | VLAN ID | Agent Violation | Invalid IP Gateway

----------+---------+-----------------+-------------------  
 1/1         100                 400                   0

2/3         100                 500                   0

Sample output format for show ip helper option-82 error-count port 2/3

show ip helper option-82 error-count port 2/3

Slot/Port | VLAN ID | Agent Violation | Invalid IP Gateway

----------+---------+-----------------+-------------------  
 2/3         100                 500                   0

2/3         200                 0                  10

2/3        1100                 500                   0

2/3        3100                 500                   0

Sample output format for show ip helper option-82 error-count 2/3 vlan 100

show ip helper option-82 error-count port 2/3 vlan 100

Slot/Port | VLAN ID | Agent Violation | Invalid IP Gateway

----------+---------+-----------------+-------------------  
 2/3         100                 500                   0

Sample output format for show ip helper option-82 error-count port 2/3

show ip helper option-82 error-count

Slot/Port | VLAN ID | Agent Violation | Invalid IP Gateway

----------+---------+-----------------+-------------------  
 1/1         100                 400                   0

2/3         100                 500                   0

2/3         200                 0                  10

2/3        1100                 500                   0

2/3        3100                 500                   0

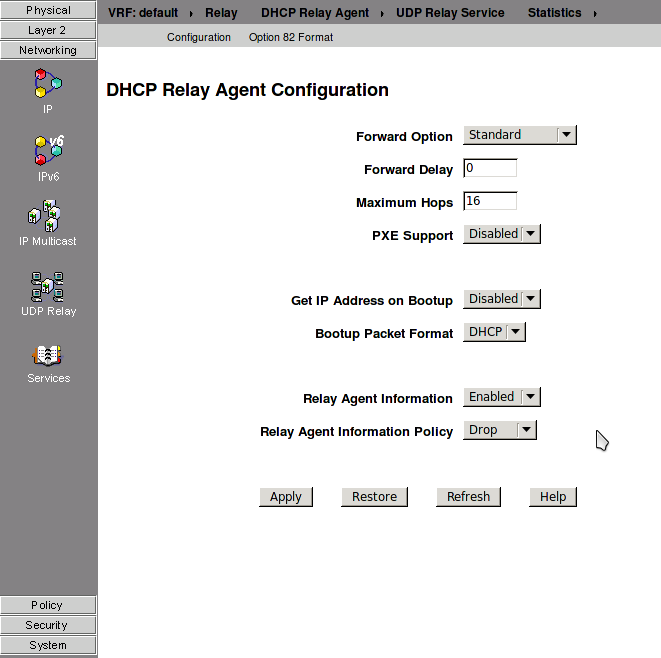
In 6.X the configuration of the Option 82 format is under the DHCP Snooping menu. In 7.2.1.R01, there will be a new menu item for Option 82 format. Sample screen shot for WebView Option 82 configuration/display.

***Figure 1*** is the default configuration for DHCP Relay agent.

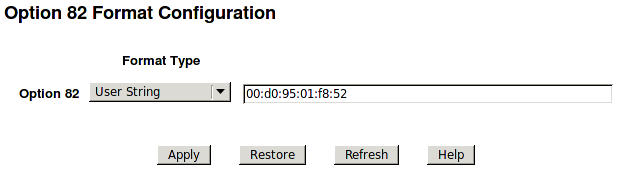
***Figure 2*** is the Option 82 format configuration screen when user selected User String. There will be box to display if any user string is configured. To change the string, replace the existing string with the new string on the display box and then hit the “apply” button to apply the change.

***Figure 3*** is when ASCII format is chosen and one of the ASCII is user string with delimiter being a space.

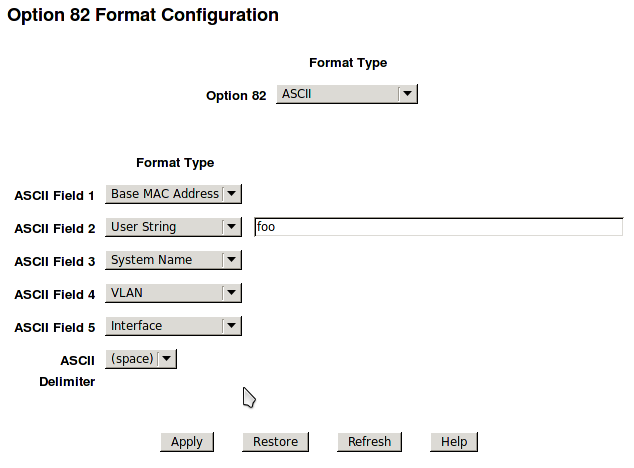
***Figure 4*** is the Option 82 format screen when agent information is disabled. There is a warning message to tell the user that agent information is disabled. Setting the format has not effect.

***Figure 1****:* 

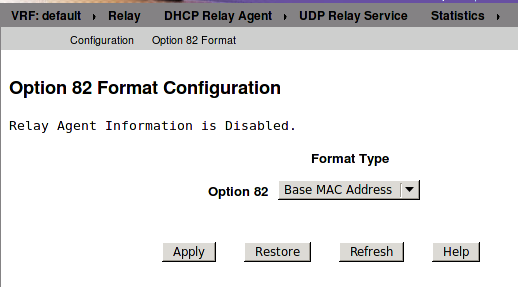
***Figure 2****:*



***Figure 3****:*



***Figure 4****:*



Appendix D: RTR 3257 DHCP-Snooping for 8.1.1.R01

# DHCP Snooping

## Introduction

DHCP Snooping will be added as part of the 8.1.1.R01 release. This is in response to Software System Requirements Specification (SRS) AOS Release 8.1.1.R01 (P/N 012225-00). The functionality provided will be that currently provided in release 6.4.5.R02. This effort includes support for

1. DHCP-Snooping of IP address that are seen on the switch/router.
2. Ingress Source Filtering (ISF) and support for
3. DHCP Relay Agent Information option 82 specified by RFC 3046.

There are also new requirements in 8.1.1.R01 for DHCP-Snooping to support Access Guardian (AG).

Most of the DHCP snooping information in this document is taken from “***DHCP Snooping Software Functional Specification P/N 011156-03***”.

Since the functionality of DHCP Snooping is provided by the UDP Relay task, the DHCP Snooping functionality description has been added to this SFS.

### Referenced Documents

* Software System Requirements Specification (SRS) AOS Release 8.1.1.R01, P/N 012225-00
* DHCP Snooping Software Functional Specification, P/N 011156-03
* Access Guardian 2.0 RTR 3215 System Functional and Design Specification, P/N 013525-00

## Requirements Overview

### DHCP-Snooping

DHCP snooping is a DHCP security feature that provides network security by filtering untrusted DHCP messages by building and maintaining a DHCP snooping binding database (a.k.a., DHCP snooping binding table).

DHCP snooping acts like a firewall between untrusted hosts and DHCP servers. The user can use DHCP snooping to differentiate between untrusted interfaces connected to the end user and trusted interfaces connected to a DHCP server or another switch.

An untrusted message is a message that is received from outside the network or firewall. When DHCP snooping is used in a service-provider environment, an untrusted message is sent from a device that is not in the service-provider network, such as a customer's switch. Messages from unknown devices are untrusted because they could be sources of traffic attacks.

### Ingress Source Filtering

In addition to filtering untrusted DHCP messages, DHCP Snooping can allow user to configure Ingress Source Filtering as a security feature. When **Ingress Source Filtering** (ISF) is enabled on a port or linkagg port, the initial packets permitted for traffic are DHCP, DNS and ARP so as to allow the client to obtain and IP address from the DHCP server in which a MAC-IP Binding entry is created in the DHCP Snooping task, it will then allow packets that match the IP address/MAC address/ port combination that is obtained from the DHCP snooping binding table entry. Other non-matching packets will be dropped.

When ISF is enabled on a vlan then the vlan ID is added to the matching criteria as an additional parameter that must be matched.

Ingress Source Filtering only works for clients that use DHCP to obtain their IP address. For clients that have static IP address configured, user will have to create static binding entry for these devices.

### DHCP Relay Agent Information Option-82

Option 82 is an option specified by RFC 3046 for DHCP relay agents to insert information in packets forwarded to DCHP servers and to remove that same information before relaying responses back to DHCP clients. Appendix C: RTR 2418 for 7.2.1.R01 in this document describes the current implementation of option-82 in the 7.x release.

The current implementation of option-82 in 7.x is mostly compatible with the feature as it exists in 6.x. There are some CLI changes that are detailed later in this document.

### Access Guardian

Access Guardian (AG) has the following requirements that must be provided by DHCP snooping:

* ~~DHCP snooping should be enabled on all UNP ports by default. MAC verification and option-82 insertion is not needed, only binding table creation~~.
* ~~AG needs to be able to direct DHCP snooping to turn on ISF for particular UNP ports.~~
* ~~If a DHCP binding entry does not exist, DHCP snooping must inform AG of the failure.~~

After the SFS review (Oct 17, 2013), a meeting was held with PLM, Architecture group and Engineering group and found that the reason for the previous requirement is for the wireless project that was in the 8.1.1.R01 release. Now that the wireless project is not part of 8.1.1.R01 the requirement to have MAC level ISF on AG port based on the edge profile setting is not necessary.

For now, there is no requirement from Access Guardian for DHCP Snooping. User is allowed to configure ISF on AG port just like the other non-AG ports. Detail of DHCP Snooping on an AG port will be described in detail in the later section of this document.

### Hardware Requirements and Limitations

No new hardware requirements exist.

### Software Requirements and Limitations

The maximum number of DHCP Snooping enabled VLANs is 64.

The maximum number of vlans on which vlan level ISF can be enabled is 32.

### Security Requirements

### Redundancy and Hot Swap Requirements

Must support existing takeover and vc-takeover requirements that exist for the switch.

## Functional Description

### DHCP Snooping

This feature prevents the normal flooding of DHCP Discover/Request and DHCP Offer packets. These packets will instead be delivered only to the appropriate DHCP server and client ports respectively.

In a service-provider network, a trusted interface is connected to a port on a device in the same network. An untrusted interface is connected to an untrusted interface in the network or to an interface on a device that is not in the network.

When a switch receives a DHCP packet on an untrusted interface and the interface belongs to a VLAN in which DHCP snooping is enabled, the switch performs filtering and screening, and drops/blocks a DHCP packet when one of these situations occurs:

* A packet from a DHCP server, such as a DHCPOFFER, DHCPACK, or DHCPNAK packet, is received from outside the network or firewall.
* A DHCP packet is received on an untrusted interface, and the source MAC address in the Ethernet header and the DHCP client hardware address in the packet do not match.
* The switch receives a DHCPRELEASE or DHCPDECLINE broadcast message that contains a MAC address in the DHCP snooping binding table, but the interface information in the binding table does not match the interface on which the message was received.
* A DHCP relay agent forwards a DHCP packet that includes a relay-agent IP address that is not 0.0.0.0.
* The relay agent forwards a packet that includes option-82 information to an untrusted port.

A port which belongs to a VLAN with DHCP Snooping enabled, can have three trust levels. Both DHCP Blocked and DHCP Client Only conceptually fall in the “untrusted” category.

1. **DHCP Blocked** (No DHCP traffic allowed);
2. **DHCP Client Only** (Default operation mode when DHCP Snooping is enabled on the VLAN);
3. **DHCP Trusted** (All DHCP traffic is allowed, where the DHCP server is connected). A trusted port behaves the same as if the port does not have DHCP Snooping enabled.

The **DHCP snooping binding database** contains the MAC address, the IP address, the lease time, the binding type, the VLAN number, and the interface information that corresponds to the local untrusted interfaces of a switch.

The DHCP snooping binding database/table needs to be persistent to survive the switch reboot/takeover.

DHCP snooping also has the following related requirement.

* Ingress Source Filtering, when enabled, only the data originating/incoming from the client’s MAC address, port and IP address will be allowed. All other packets will be dropped by default. The system uses the dhcp snooping binding table for the necessary information.

It is worth noting that the normal operation of DHCP Snooping relies on the following assumptions:

* The DHCP clients and server are on separate VLANs. Hence, the Relay Agent is always defined and used on the switch.
* The Option-82 data insertion is always enabled as part of DHCP Snooping.
* The DHCP Sever has to support Option-82 (as a minimum, preserving and echoing back the option-82 data field).

The DHCP Snooping feature can be enabled/activated at two levels, namely:

* The vlan level
* The switch level

**Note:** The Switch Level DHCP Snooping and the VLAN Level DHCP Snooping are mutually exclusive. That is, they cannot be configured/enabled at the same time.

#### VLAN Level DHCP Snooping

Once DHCP Snooping is enabled on the VLAN, this implies that, all the DHCP packets through the interfaces that belong to this VLAN are subject to be filtered/screened. In AOS, we say that those interfaces of the vlan are in “DHCP Client Only” mode.

The following incoming DHCP packets to those “**DHCP Client Only**” interfaces will be dropped:

* All the DHCP Server packets: DHCP-OFFER, DHCP-ACK, DHCP-NACK, DHCP-LEASE-QUERY, etc.;
* A DHCP packet that includes a relay agent IP address that is not 0.0.0.0;
* A DHCP packet that includes the DHCP Option-82 information.

Should it be necessary, an interface can be configured to block all the DHCP packets (DHCP port in Blocked mode). Or it can be configured to allow all the DHCP without performing any filtering (DHCP port in trusted mode).

Once DHCP Snooping is enabled for a VLAN, the following parameters can also be configured against the VLAN:

* **Enable/Disable DHCP Option-82 data insertion**: By default, the Option-82 data insertion is enabled. Once enabled, the DHCP reply packets will be sent to the interface/port that the original DHCP request comes in from, instead of flooding to all the interfaces/ports associated with the VLAN.
* **Enable/Disable MAC address verification**: By default, it is enabled. When enabled, the switch will verify/compare the source MAC address in the Ethernet header and the DHCP client hardware address in the packet, if the addresses do not match, the DHCP packet will be dropped.

When DHCP snooping is enabled on a vlan, flooding of DHCP broadcast packets must be prevented on that vlan. DHCP packets will only be sent to the configured server via the UDP relay task. Suppression of vlan flooding of DHCP broadcast packets is implemented via a QOS API.

#### Switch Level DHCP Snooping

When DHCP Snooping is enabled against the switch, all the DHCP packets through all the interfaces are subject to be filtered/screened. By default, all the interfaces will allow “**DHCP Client only**” traffic. In this case, the DHCP-trusted interface(s) need to be properly configured to allow DHCP server traffic to go through. In addition, there will be the following two global/switch-level parameters:

* DHCP Option-82 data insertion
* MAC address verification

The behavior of DHCP Snooping at the switch level, and the impact of enabling/disabling of the above two parameters are the same as at the VLAN level. Please refer to Section 8.3.1.1 for detailed description.

**Note**: Both switch level and vlan level DHCP Snooping can operate at layer 2 and layer 3. The sending of DHCP packet to trusted ports is operating on layer 2. The concept of a “trusted” port is not applicable when the DHCP Server resides on another vlan and a Relay Agent is used to send the DHCP packet to the DHCP Server. Regardless if the DHCP server is on the same VLAN as the client, DHCP packets are processed in the CMM and MAC address to IP address binding entries are created. Same as in 6.X, DHCP Snooping and the Internal DHCP server are mutually exclusive.

#### DHCP Snooping Traffic Violation Statistics

DHCP Snooping traffic filtering/blocking statistics are kept per port. There are five counters:

* **MAC Address violation counter**. This counter is incremented when an DHCP packet is received on an untrusted interface, and the Ethernet source MAC address and the DHCP client hardware address do not match.
* **DHCP Server packets violation counter**. This counter is incremented when a DHCP packet from a DHCP server, such as a DHCPOFFER, DHCPACK, DHCPNAK, or DHCPLEASEQUERY packet, is received on an untrusted port.
* **DHCP binding violation counter**. This counter is incremented when the switch receives a DHCPRELEASE or DHCPDECLINE broadcast message that contains a MAC address in the DHCP snooping binding table, but the interface information in the binding table does not match the interface on which the message was received.
* **DHCP Option 82 violation counter**. This counter is incremented when a relay agent forwards a packet that includes option-82 information to an untrusted port.
* **DHCP Relay Agent counter**. This counter is incremented when a DHCP relay agent forwards a DHCP packet that includes a relay-agent IP address that is not 0.0.0.0.

**Note:** The above statistics violation counters are applicable for both switch-level and vlan-level DHCP Snooping. And they are only applicable when the port is in the “Client-Only” trust mode. When the port mode is change from “Client-Only” to “Blocked/Trusted”, the counters are reset to 0.

#### DHCP Snooping Option-82 Policy

A new command has been introduced under DHCP snooping option-82 configuration.

The command format is as below.



*-> dhcp-snooping option-82 policy*

*drop     keep     replace*

This command enables user to specify policy to keep / replace / drop the option-82 field of DHCP packet entering the AOS switch.

Default policy is “Replace”, which is existing behavior.

Example Topology:

**M1 (AOS / Third-Party DHCP server) ------------ M2(AOS L2 Switch) -------------- M3(AOS L2 switch) -------- M4(Client Node)**

In the above topology, M3 and M2 are the AOS switches with option-82 fields configured.

Below table summarizes the behavior of the command.

|  |  |  |
| --- | --- | --- |
| **DHCP Cases** | **DHCP Snooping Opt-82 Policy** | **Behavior of M2 w.r.t opt-82** |
| Snooping Disabled | Keep/Replace/Drop | Ignores opt82 field in packet |
| Snooping Enabled &  Trusted port | Keep/Replace/Drop | Ignores opt82 field in packet |
| Snooping Enabled &  client-only port &  bypass-opt82-check disabled | Replace | Drops the packet with opt-82 |
| Snooping Enabled & client-only port & bypass-opt82-check enabled | Replace | Replaces opt-82 fields in packet |
| Snooping Enabled & client-only port &  Bypass-opt82-check enabled | Keep | Keeps opt-82 fields in packet |
| Snooping Enabled & client-only port &  Bypass-opt82-check enabled | Drop | Drops the packet with opt-82 |

This enhancement has been introduced from 8.5. R02 release onwards.

### Basic Architecture

The UDP Relay CMM task will maintain three (3) tables to store the DHCP Snooping related configuration data:

1. the DHCP Snooping VLAN table,
2. the DHCP Snooping Port table and
3. the DHCP Snooping Binding table.

The task will look up the configured data in these three tables to filter/screen the DHCP packets appropriately.

#### DHCP Snooping VLAN Table

The **DHCP Snooping VLAN table** is indexed by the VLAN id, on which DHCP Snooping has been enabled. And it contains three “VLAN-level” DHCP Snooping configuration flags:

* VLAN Id (index)
* Option-82 data insertion (enable/disable)
* MAC-Address verification (enable/disable)

The UDP Relay CMM task will register with Vlan Manager for vlan events. It needs information about vlan creation and deletion as well as vlan/port associations so that DHCP snooping can be enabled and configured on the appropriate vlans and ports.

The DHCP MAC-Address verification is achieved by looking up the source MAC address of the ETH header, and the Client Hardware address in the DHCP packet.

#### DHCP Snooping Port Table

The **DHCP Snooping Port table** is indexed by the port. And it contains the “DHCP trusted” state of the interface, as well as the DHCP traffic violation statistics:

* IfIndex (representing the interface).
* Trust state/mode (Client-only, Blocked or Trusted).
* IP Source Filtering
* MAC Address violation counter.
* DHCP Server packets violation counter.
* DHCP binding violation counter.
* DHCP Option 82 violation counter.
* DHCP Relay Agent violation counter.

#### DHCP Snooping Binding Database/Table

By default, once DHCP Snooping is enabled at either the switch-level or the vlan-level, the DHCP Snooping Binding Database capability will be enabled.

The DHCP Snooping Binding table is indexed by the physical port and the client’s MAC address. It contains the following data:

* Client’s MAC Address;
* Client’s IP Address assigned by the DHCP Server;
* The physical port where the DHCP request is coming from;
* The VLAN Id where the DHCP request packet is coming from;
* The lease time of the IP Address;
* The type/nature of how the binding entry is populated, either static or dynamic.

The binding table entries are usually populated by the UDP Relay software as it tracks the DHCP packets against the client H/W MAC address and the physical port. It does not require any human intervention. This type is called “dynamic” (dynamically learned). When the binding entry, for any specific reason, is created by a human admin, the type is called “static” (statically configured). The static binding entries take precedence over the dynamic entries. That is, if there exists a dynamic binding entry in the binding table, it will be replaced by a newly learned static entry; while if there exists a dynamic entry, when the user attempts to add a static entry with the same MAC Address and Slot/Port, the dynamic entry will be replaced with static entry.

Since the DHCP snooping binding database needs to be persistent to survive the switch reboot/takeover, the snooping binding table is periodically saved to a file. It is named ***dhcpBinding.db*** under the */flash/switch* directory. The synchronization period is configurable, and by default is 1 seconds. In addition, there will be a timestamp stating the last time the synchronization has been successfully performed. This file is also sent to the secondary CMM in a dual-CMM setup. This will have to be sent to the other chassis’s in a virtual chassis environment.

The dynamic binding entry is populated when the Relay Agent receives a DHCP-ACK packet. By default the Relay Agent will remove a binding entry when one of the following conditions occurs:

* Receiving a DHCP-RELEASE packet (**Note**, it is commonly seen that the Relay Agent does not receive the DHCP-RELEASE packets on Windows when *ipconfig /release* is performed);
* When the Relay Agent’s Lease Timer is decremented to 0;
* Receiving a NI-Detach event from port manager;
* Receiving a link-down event from port manager;
* If the MAC is aged out by source learning. This check is made at the time we sync the binding database to a file.

If binding persistency is enabled by the user (default is disabled) then the only events that will cause the binding entry to be removed are receiving a DCHP-RELEASE packet or the expiration of the lease timer. The other events that normally cause removal will be ignored.

**Note:** Due to the synchronization period, there will potentially be a discrepancy between the binding database in the memory and the flash binding database file. Also, for the same reason the binding table in the memory might not be removed promptly, since the MAC Address aging is only checked every synchronization time period.

There are three actions defined against the DHCP Snooping binding database. The purpose of those actions is mainly for re-synchronization of the binding table (in memory) and the database (in flash).

* The “**Purge**” action is to clear what’s in the memory;
* The “**Renew**” action is to populate the binding table in the memory based on the flash file.
* The “**Save**” action is to save the binding table in the memory based on to the flash file.
* The max number of Binding entries in the DHCP Snooping Binding Table is 4096. (This is a soft limit that is put in place for entries syncing to the secondary and/or slave chassis).
* DHCP Snooping Binding Table on the Master primary chassis resides in memory.  This table will be sync to flash based on the value of dhcpSnoopingBindingDatabasesyncTimeout value.  The default is 5 minutes.  The lowest value is 1 minute.
* Once DHCP Snooping Binding Table is written to flash on the Master primary CMM, the system will sync this to all the secondary/slave CMMs.
* If before the next sync to flash operation, there is a takeover action the new binding entries that are still in memory will not be saved to flash.  The new Master primary CMM will not have the new entries.
* The DHCP Snooping Binding Table Persistent flag is set as disable by default same as 6.X.
* Before writing to flash, the system will decrement lease time of each entry in the DHCP Snooping Binding Table that is in memory.  The system will delete those entries that the lease time expired.
* When the dhcpSnoopingBindingDatabasesyncTimeout is changed, the previous timer is stopped and the system will execute the timeout out with respect to the time that the timeout value is changed. (Start from fresh).
* Ingress Source Filtering can only be enabled on the “client-only” ports.

### Ingress Source Filtering

Ingress Source Filtering (ISF) can be enabled via the CLI. The Default mode is DISABLED.

On a system with DHCP Snooping (System or VLAN level), the MAC address on the AG port will follow the configured DHCP Snooping behavior after classification. Before classification, all unknown packets are sent to the UNP task for software classification.

Ingress Source Filtering (ISF) can be enabled via the CLI or other management interface such as SNMP and WebView.

Ingress Source Filtering relies on the content of the DHCP Snooping Binding Table:

* If there is no entry found in the DHCP Snooping Binding Table for a particular MAC on a given port, ISF cannot be created for the MAC address.
* Each port can have one entry for a given MAC address.
* For device using static IP address, user will have to manually create a “Static Binding entry” to DHCP Snooping Binding Table.

The total maximum number of DHCP binding entries that a system can learn is **4096**.

The maximum number of DHCP binding entries that can be learned on a system when ISF is enabled is tied to the number of TCAM entries that is required to be set on each NI.  The required number of TCAM entries varies depending on the configuration.   In all cases, we work off the number 256 – the number of TCAM entries available for DHCP snooping per NI.  In other words DHCP binding entry depends on available TCAM entry.  The system will not be able to learn a new DHCP binding entry when there is no TCAM entry available and in the case of linkagg all member ports will have to have TCAM entry available on the NI that it resides on.

There are 2 ways that a TCAM entry is being used up by a system.  The first one is the configuration of ISF – we will refer this as the **Bind Enablement** operation. The second is the creation of a DHCP binding entry – we will refer this as the **Bind Creation** operation.

**Port Level ISF**:

Port level ISF can be enabled on a fixed port or on a linkagg port.  For fixed port the Bind Enablement and Bind Creation operate on the NI that the port is on.  For linkagg port the Bind Enablement and Bind Creation operate on all member ports of the linkagg.

The TCAM for **Bind Enablement** works as a bitmap and therefore whether one or all ports on a NI has port level ISF enabled, 3 TCAM entries are used.

It is safe to assume that for port level ISF we have 254 TCAM entries to be use as DHCP binding resources.

If there is no port level ISF configured on any linkagg port we can learn up to **253 \* Number of NI on the systems** and not to exceed the total system max of 4096.

When port level ISF is configured on a linkagg port the number of DHCP binding entries that can be learned on a system depends on the location of the member ports.  Each member port will take up one TCAM entry.  The system will stop learning DHCP binding entries when any one of the NI with the linkagg’s member port is running out of TCAM entry.

If one linkagg has one member port on all of the NI on the system then the max number of DHCP binding entries will be 254.  If on one NI there are 2 member ports, 2 TCAM entries will be used up.  In this extreme case the max number of DHCP binding entries that can be learned on the system is 127.

If one NI had both fix port and member port of a linkagg with ISF enabled, we will have to calculate how many TCAM entries are used and if there is no TCAM entry available then the system will stop learning DHCP binding entries.

**VLAN level ISF**:

For VLAN level ISF, **Bind Enablement** for each VLAN will take up 3 TCAM entries on all NIs. User can only enable ISF on 32 VLANs.

On **Bind Creation**, each DHCP binding entry will require one TCAM entry for fixed port and one TCAM entry for each member port of the linkagg.

In the system with no linkagg port on the VLAN with ISF enabled, the system can learn up to 256 – (3 \* Num VLAN with ISF enabled).  For example with one VLAN with ISF enabled, the system can learn 253 entries on the system.

(256 – (3 \*1)) = 253

For linkagg port each member port requires one TCAM entry.  In the extreme case where there are 2 member ports on the same NI, the system will learn 127 DHCP binding entries.

Again, where there is a mix of fixed and linkagg port configured on a VLAN with ISF enabled then we have to calculate how many TCAM entries are used up to determine the number of DHCP binding entries that can be learned on a system.

### DHCP Option-82

Option 82 in **UDP Relay** (ip helper agent information) is to provide additional information for the DHCP server to know about the identity of the client for the purpose of assigning an IP address for the client.

Option 82 in **DHCP Snooping** can provide additional information for the DHCP server for IP address assignment but it is used for port identification for DHCP Snooping purpose.  Option 82 is inserted by DHCP snooping when sending the DHCP packet to the DHCP Server and the DHCP Snooping will strip off this option if it is present in the packet sent from the DHCP server to the client.  Some DHCP server does not echo back this option at all (i.e. DHCP Reply does not contain the Option 82 information from DHCP Request packet).

The usage of Option 82 in UDP Relay (ip helper agent information) and DHCP Snooping is different and thus they are mutually exclusive.  This is the behavior for Option 82 in 6.X

In 6.X, user can only configure Option 82 in the default VRF and the configuration can be applicable to either ip helper agent-information or ip helper dhcp-snooping but not to both at the same time.

In 8.1.1.R01, we separate the Option 82 configuration for ip helper and dhcp-snooping. The Option 82 for ip helper will be the same as in 7.X and is applicable for Agent Information only. If user wants to configure Option 82 for dhcp-snooping, the new CLI command ***dhcp-snooping*** *option-82 format* …. will be used.

In 6.x the command to specify the format of the option-82 information includes the dhcp-snooping keyword “*ip helper* ***dhcp-snooping*** *option-82 format* …”. In 7.x the command to do this is “*ip helper option-82 format …”.* We will maintain the 7.x syntax for Agent Information.

In 6.x the option-82 format command “*ip helper* ***dhcp-snooping*** *format ascii {field1 [field2 … field5 delimiter]}”* command has been deprecated and replaced with commands to explicitly configure the Circuit ID and Remote ID. This is carried over to 8.1.1.R01 as “*dhcp-snooping format ascii circuit-id {field1 [field2 … field5 delimiter*]}” and “*dhcp-snooping format ascii remote-id {field1 [field2 … field5 delimiter*]}”.

The Option 82 Format type in both 6.X and 7.X remains the same. The available Option 82 format types are:

1. MAC Address
2. System Name
3. User String
4. Interface Alias
5. Auto Interface Alias
6. ASCII

For Option 82 there is the “Format Type” and when the format type is ASCII there is the applicable “ASCII Type Format”.

The available ASCII Format Types in 6.X are:

1. Mac Address
2. System Name
3. User String
4. Interface Alias
5. VLAN
6. Interface
7. CVLAN
8. Remote-ID
9. Circuit-ID

Release 8.1.1 option 82 for DHCP Snooping will support all these 9 ASCII format types.

The **cvlan** is added as an available ASCII field to insert for the Circuit ID and/or the Remote ID. This new keyword will be added when the ASCII format is selected.

If explicit remote-id or circuit-id is configured, then option-82 fields will be populated based on this explicit configuration irrespective of the configuration done globally.

If explicit remote-id is configured, but circuit-id has not been configured, then circuit-id will be populated based on global configuration. In this case, the Circuit-ID will be filled up with vlan, chassis, slot and port followed by option-82 global values. For standalone unit the chassis field will be zero.

If explicit circuit-id is configured and remote-id is not configured, then the Remote ID content will be TLV with a value that equals the configured option 82 format.

If explicit remote-id and circuit-id are not set, then Circuit ID will be filled up with vlan, chassis, slot and port followed by Client Host Name if set, and remote-id will be populated with Client Host Name. For standalone unit, the chassis field will be zero

For Virtual Chassis support, we will support chassis/slot/port in the ASCII Interface.

### Access Guardian

As the requirement from Access Guardian to DHCP Snooping has changed, there is no interaction between Access Guardian and DHCP Snooping in 8.1.1.R01.

### Management Interfaces

#### SNMP

All features will be configurable via SNMP.

#### Web-View

All features will be configurable via Web View.

#### Command Line Interface

The 6.x commands have changed to follow the 7.x convention of using the keyword “*admin-state”* before the keyword “*enable/disable”.*

##### Global Commands

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dhcp-snooping admin-state <enable/disable>*** | | | | |
| **Table ID:** dhcpSnoopingTable | | | | |
| **Object ID:** dhcpSnoopingMode | | | | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| mode | Int32 | 1..3 | mip\_dhcpSnoopingMode | This controls the mode of operation of DHCP snooping |
| **Dependencies:**  N/A. | | | | |
| **Description:** This value indicates the dhcp-snooping mode. It is enabled implicitly for vlanLevel when DHCP snooping is enabled on an individual vlan. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dhcp-snooping mac-verification admin-state <enable/disable>*** | | | | |
| **Table ID:** dhcpSnoopingTable | | | | |
| **Object ID:** dhcpSnoopingMacAddrVerificationStatus | | | | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| mode | Int32 | 1..2 | mip\_dhcpSnoopingMacAddrVerificationStatus | This controls the mode of operation of DHCP snooping mac address verification |
| **Dependencies:**  DHCP snooping must be enabled. | | | | |
| **Description:** This command enables/disables dhcp snooping MAC-address verification at the switch level. Default is enabled. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dhcp-snooping opt82-insertion admin-state <enable/disable>*** | | | | |
| **Table ID:** dhcpSnoopingTable | | | | |
| **Object ID:** dhcpSnoopingOpt82InsertionStatus | | | | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| mode | Int32 | 1..2 | mip\_dhcpSnoopingOpt82InsertionStatus | Enables DHCP Option-82 data insertion on incoming DHCP packets at the switch level |
| **Dependencies:**  DHCP snooping must be enabled. | | | | |
| **Description:** This command enables/disables insertion of Option-82 data into incoming DHCP packets at the switch level. Default is enabled. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dhcp-snooping bypass-opt82-check admin-state <enable/disable>*** | | | | |
| **Table ID:** dhcpSnoopingTable | | | | |
| **Object ID:** dhcpSnoopingBypassOpt82CheckStatus | | | | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| mode | Int32 | 1..2 | mip\_dhcpSnoopingOpt82CheckStatus | Enable or disable the check of option-82 presence on untrusted ports. |
| **Dependencies:** | | | | |
| **Description:** This command enables/disables the check of option-82 presence on untrusted ports. Default is disabled. When enabled if a packet is with option-82 field on an untrusted port the existing option-82 information will be replaced by the vlan/gport information. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dhcp-snooping option-82 format [base-mac | system-name | user-string {“string”} | interface-alias | auto-interface-alias***  ***dhcp-snooping option-82 format ascii remote-id {[ base-mac | system-name | user-string {“string”} | interface-alias | auto-interface-alias | vlan | cvlan ]} [[base-mac | system-name | user-string {“string”} | interface-alias | auto-interface-alias | vlan | cvlan] delimiter {“string”}]***  ***dhcp-snooping option-82 format ascii circuit-id {[ base-mac | system-name | user-string {“string”} | interface-alias | auto-interface-alias | vlan | cvlan ]} [[base-mac | system-name | user-string {“string”} | interface-alias | auto-interface-alias | vlan | cvlan] delimiter {“string”}]***  ***no dhcp-snooping option-82 format ascii remote-id***  ***no dhcp-snooping option-82 format ascii circuit-id*** | | | | |
| **Table ID:** dhcpSnoopingOption82FormatASCIIConfigurableTable  See MIB for details on this table | | | | |
| **Object ID:** dhcpSnoopingOption82FormatASCIIConfigurableEntry | | | | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
|  |  |  |  |  |
| **Dependencies:** | | | | |
| **Description:** This command allows the user to configure the data inserted into the option-82 field of the DHCP packets from the clients. For the first form of the command only 1 field type may be selected. For the 2nd and 3rd forms of the command with the ascii keyword, the user may select up to 5 field names to use for circuit-id and remote-id. If more than 1 field is selected then the delimiter keyword and delimiter string are required. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dhcp-snooping binding admin-state <enable/disable>*** | | | | |
| **Table ID:** dhcpSnoopingTable | | | | |
| **Object ID:** dhcpSnoopingBindingStatus | | | | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| mode | Int32 | 1..2 | mip\_dhcpSnoopingBindingStatus | This controls the mode of operation of the DHCP snooping binding table |
| **Dependencies:**  DHCP snooping must be enabled. Default is enabled. | | | | |
| **Description:** This command enables/disables dhcp snooping binding table capability. Default is enabled. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dhcp-snooping binding timeout <number>*** | | | | |
| **Table ID:** dhcpSnoopingTable | | | | |
| **Object ID:** dhcpSnoopingBindingDatabaseSyncTimeout | | | | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| Value | Int32 | 1..600 | mip\_idhcpSnoopingBindingDatabaseSyncTimeout | Timeout value in seconds for synchronization between binding table and flash file |
| **Dependencies:**  DHCP snooping must be enabled. | | | | |
| **Description:** Configure the timeout value for synchronization between DHCP snooping binding table and flash file. Default value is 1 second. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dhcp-snooping binding action <purge/renew/save>*** | | | | |
| **Table ID:** dhcpSnoopingTable | | | | |
| **Object ID:** dhcpSnoopingBindingDatabaseAction | | | | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| Value | Int32 | 1..2 | mip\_idhcpSnoopingBindingDatabaseAction | Action to perform on binding table database |
| **Dependencies:**  DHCP snooping must be enabled. | | | | |
| **Description:** Actions for synchronization of binding table (in memory) and database (in flash). Purge: to clear what’s in memory. Renew: to populate the binding table in memory based on the flash file. Save: to save in flash what’s in the memory. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dhcp-snooping binding persistency admin-state <enable/disable>*** | | | | |
| **Table ID:** dhcpSnoopingTable | | | | |
| **Object ID:** dhcpSnoopingBindingPersistencyStatus | | | | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| Value | Int32 | 1..2 | mip\_dhcpSnoopingBindingPersistencyStatus | Enable/disable binding persistency |
| **Dependencies:**  DHCP snooping must be enabled. | | | | |
| **Description:** Enables or disables binding table persistency. When enabled, only a lease expiration will cause the binding entry to be removed. Default state is disabled. | | | | |

##### VLAN Table Commands

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dhcp-snooping vlan <vlanId>*** | | | | |
| **Table ID:** dhcpSnoopingVlanTable | | | | |
| **Object ID:** dhcpSnoopingVlanNumber | | | | |
| **Object ID:** dhcpSnoopingVlanStatus | | | | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| Vlan ID | Int32 | 1..4094 | mip\_dhcpSnoopingVlanNumber | VLAN ID |
|  | RowStatus | 1..6 | mip\_dhcpSnoopingVlanStatus | Create/delete status of DHCP snooping for specified vlan |
| **Dependencies:** | | | | |
| **Description:** Enables/disables DHCP snooping feature for specified vlan. Use *no dhcp-snooping vlan<vlanID>* to disable | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dhcp-snooping vlan <vlanId> mac-verification admin-state enable/disable>*** | | | | |
| **Table ID:** dhcpSnoopingVlanTable | | | | |
| **Object ID:** dhcpSnoopingVlanNumber | | | | |
| **Object ID:** dhcpSnoopingVlanMacAddrVerificationStatus | | | | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| Vlan ID | Int32 | 1..4094 | mip\_dhcpSnoopingVlanNumber | VLAN ID |
| Value | Int32 | 1..2 | mip\_dhcpSnoopingVlanMacAddrVerificationStatus | MAC address verification status |
| **Dependencies:** DHCP snooping must be enabled. | | | | |
| **Description:** Enables/disables MAC address verification status for received DHCP packets. Default state is enabled. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dhcp-snooping vlan <vlanId> opt82-insertion admin-state enable/disable>*** | | | | |
| **Table ID:** dhcpSnoopingVlanTable | | | | |
| **Object ID:** dhcpSnoopingVlanNumber | | | | |
| **Object ID:** dhcpSnoopingVlanOpt82DataInsertionStatus | | | | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| Vlan ID | Int32 | 1..4094 | mip\_dhcpSnoopingVlanNumber | VLAN ID |
| Value | Int32 | 1..2 | mip\_dhcpSnoopingVlanOpt82InsertionStatus | DHCP Option-82 data insertion status |
| **Dependencies:** DHCP snooping must be enabled. | | | | |
| **Description:** Enables/disables DHCP option-82 insertion on received DHCP packets for the specified vlan. Default state is enabled. | | | | |

##### Port Table Commands

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dhcp-snooping [port <chassis/slot/port> | linkagg <linkAggId>] <block/client-only/trust>*** | | | | |
| **Table ID:** dhcpSnoopingPortTable | | | | |
| **Object ID:** dhcpSnoopingPortIfIndex | | | | |
| **Object ID:** dhcpSnoopingPortTrustMode | | | | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| Chassis/slot/port | Int32 |  | mip\_dhcpSnoopingPortIfindex | Ifindex of port |
| Trust mode | Int32 | 1..3 | mip\_dhcpSnoopingPortStatusMode | Block/Client-only/Trust |
| **Dependencies:** DCHP snooping must be enabled. | | | | |
| **Description:** Sets the trust level of a port. Default is Client-only for all ports in vlan on which DHCP snooping is enabled. | | | | |

##### Binding Table Commands

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dhcp-snooping binding <mac-addr> port <chassis/slot/port> address <ip-address> vlan <vlanId>*** | | | | |
| **Table ID:** dhcpSnoopingBindingTable | | | | |
| **Object ID:** dhcpSnoopingBindingMacAddress | | | | |
| **Object ID:** dhcpSnoopingBindingIfIndex | | | | |
| **Object ID:** dhcpSnoopingBindingIpAddress | | | | |
| **Object ID:** dhcpSnoopingBindingVlan | | | | |
| **Object ID:** dhcpSnoopingBindingRowStatus | | | | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| MAC address | MAC Address |  | mip\_dhcpSnoopingBindingMacAddress | MAC address |
| Chassis/slot/port | Int32 |  | mip\_dhcpSnoopingIpAddress | Ifindex of port |
| Ip Address | IPAddress |  | mip\_dhcpSnoopingBindingIpAddress | IP Address |
| Vlan ID | Int32 | 1..4094 | mip\_dhcpSnoopingBindingVlan | Vlan ID |
|  | RowStatus | 1..6 | mip\_dhcpSnoopingBindingRowStatus | Create/delete a static entry in binding table |
| **Dependencies:** DCHP snooping must be enabled. | | | | |
| **Description:** Create/delete a static entry in the binding table | | | | |

##### Ingress Source Filtering Commands

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dhcp-snooping ip-source-filter [port <chassis/slot/port> | linkagg <linkAggId>] admin-state <enable/disable>*** | | | | |
| **Table ID:** dhcpSnoopingPortTable | | | | |
| **Object ID:** dhcpSnoopingPortIfIndex | | | | |
| **Object ID:** dhcpSnoopingPortIpSourceFiltering | | | | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| Chassis/slot/port | Int32 |  | mip\_dhcpSnoopingPortIfindex | Ifindex of port |
| Status | Int32 | 1..2 | mip\_dhcpSnoopingPortIpSourceFiltering | Enable/disable |
| **Dependencies:** DCHP snooping must be enabled. | | | | |
| **Description:** The status of IP source filtering on a specified port. Default is disabled. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***dhcp-snooping ip-source-filter vlan <vlanID> admin-state <enable/disable>*** | | | | |
| **Table ID:** dhcpSnoopingSourceFilterVlanTable | | | | |
| **Object ID:** dhcpSnoopingSourceFilterVlanNumber | | | | |
| **Object ID:** dhcpSnoopingSourceFilterVlanFilteringStatus | | | | |
| CLI Input | Object Syntax Type | Range | MIB Object Name | Description |
| Vlan ID | Int32 | 1..4094 | mip\_dhcpSnoopingSourceFilterVlanNumber | Vlan ID |
| Status | Int32 | 1..2 | mip\_dhcpSnoopingSourceFilterVlanFilteringStatus | Enable/disable |
| **Dependencies:** DCHP snooping must be enabled. | | | | |
| **Description:** Enables/disables ingress source filtering on the specified vlan. Default is disabled. | | | | |

##### Show Commands

|  |
| --- |
| ***show dhcp-snooping*** |
| **Table ID:** dhcpSnoopingMIB |
| **Description:** This command shows a list of global settings for DHCP snooping. The new output is included along with the current output showing the settings and status of udp relay. |

|  |
| --- |
| ***show dhcp-snooping vlan*** |
| **Table ID:** dhcpSnoopingVlanTable |
| **Description:** This command shows a list of vlans that have been enabled for DHCP snooping. Sample output is shown below:  *VLAN Opt-82 MAC Addr*  *ID Insertion Verification*  *-------+--------------+----------------------------*  *10 Enabled Enabled*  *20 Enabled Enabled*  *….* |

|  |
| --- |
| ***show dhcp-snooping port*** |
| **Table ID:** dhcpSnoopingPortTable |
| **Description:** This command shows a list of ports along with DHCP traffic violation statistics. Sample output is shown below:  *Chassis/ Trust Opt82 MAC Server Agent Binding*  *Slot/Port Mode Violation Violation Violation Violation Violation*  ----------+---------+---------+---------+----------+----------+---------  *1/1/1 Client 0 0 0 0 0*  *1/1/10 Trusted 0 0 0 0 0*  *….* |

|  |
| --- |
| ***show dhcp-snooping binding*** |
| **Table ID:** dhcpSnoopingBindingTable |
| **Description:** This command shows a list of DHCP client MAC address and IP address binding entries. Sample output is shown below:  *MAC Chassis/ IP Lease VLAN Binding*  *Address Slot/Port Address Time ID Type*  --------------------+----------+-----------+---------+-----+---------  *00:11:22:33:44:55 1/1/5 5.5.5.56 0 5 Static*  *00:23:45:67:89:0A 1/1/20 10.1.1.6 20000 20 Dynamic*  *….* |

|  |
| --- |
| ***show dhcp-snooping ip-source-filter vlan*** |
| **Table ID:** dhcpSnoopingPortTable |
| **Description:** This command shows a list of vlans that have ip source filtering enabled. Sample output is shown below:  *VLAN Ip Src*  *ID Filtering*  ------+----------  *5 Enabled*  *….* |

|  |
| --- |
| ***show dhcp-snooping ip-source-filter port*** |
| **Table ID:** dhcpSnoopingPortTable |
| **Description:** This command shows a list of ports that have ip source filtering enabled. Sample output is shown below:  *Chassis/ Ip Src*  *Slot/Port Filtering*  -----------+----------  *1/1/5 Enabled*  *….* |

Appendix E : UDP Relay to a Specific IP

This section briefly describes the feature to be implemented as part of RTR 3446 in 741R01. The supported platforms are OS6900 (TOR), OS10K (Rushmore), OS9900 (Medora), OS6860 (Shasta), Everest (OS6865) and Nandi (OS6560).

The existing AOS implementation redirects the broadcasted UDP packets to a destination VLAN(s). This enhancement provides an additional feature of relaying the UDP packets to the configured address as a unicast packet.

This enhancement requires configuration of UDP server address/ Next hop relay instead of destination VLAN for the configured custom serviced UDP ports. Please note that this enhancement is not applicable for generic UDP packets (not DHCP packets).

**Broadcast UDP**

10.1.1.10/24

192.168.58.0/24



**UDP Server2**

**192.168.58.21**

Router

Switch

**PC**

Vlan6

Vlan5

Vlan5

**UDP Server1**

**192.168.35.16**

**Relayed/Routed**

**Unicast UDP**

**Relayed/Routed**

**Unicast UDP**

Figure 5: UDP Relay Behavior

* Broadcast UDP packet comes from user.
* Switch recognizes custom configured UDP port based on user configuration.
* Switch decides to relay the UDP packet based on configuration (IP address is configured as 192.168.58.2) for the UDP port.
* Switch forwards the original packet to next hop based on routing decision (to reach 192.168.58.2).
* L2 and L3 headers are changed- Dest MAC is Router MAC of next hop and Dest IP is the address specified for the UDP port.
* Switch still floods the original packet to vlan 5.

## 

## Assumptions and Dependencies

* The IP address configured by the user should be valid and reachable. The feature will not validate the address.
* The feature requires one outgoing IP interface through which the packet gets routed /relayed.
* The source VLAN should have a valid IP address to make sure downstream packets are routed to UDP client directly.
* The custom configured UDP packets received from any source VLAN will be relayed to the configured address.
* This requirement is only for IPv4 UDP Relay.
* This service is unidirectional only. The response from UDP server must be unicast directly to the UDP client.
* If the configured UDP server and UDP client are located on the same VLAN, then the server receives duplicate packets as the switch will perform native broadcast and relaying also. The server should handle the duplicate packets in this case.
* One relay IP per UDP port is supported (per VRF).
* UDP packets will be routed only within the same VRF.

## Scenarios

**Scenario 1: UDP custom port configured with server IP address located on different VRF.**

Configuration:

vrf vrf1 ip udp relay 5001 address 192.168.25.17

If there is no reachability is vrf1, route lookup fails for the configured relay address and all the UDP packets with destination port as 5001 will be dropped by the switch.

**Scenario 2: UDP Port relay command is modified from L2 mode to L3 mode without removing the destination VLAN attached for a UDP custom port.**

Configuration:

**ip udp relay 5001**

**ip udp relay 5001 vlan 6**

ip udp relay 5001 address 192.168.58.21

ERROR: UDP port configured in L2 mode.

Switch will reject the configuration as udp port relay is already associated with vlan. At any given instance udp port relay can either associated in L2 mode or L3 mode.

**Scenario 3: UDP Port relay command is modified from L3 mode to L2 mode without deleting the IP address configured for a UDP custom port.**

Configuration:

ip udp relay 5001

ip udp relay 5001 address 192.168.58.21

ip udp relay 5001 vlan 6

ERROR: UDP port configured in L3 mode.

Switch will reject the configuration as udp port relay is already associated in L3 mode. At any given instance UDP relay can either associated in L2 mode or L3 mode.

**Scenario 4: UDP port relay command is modified in L3 mode and a different IP address configured.**

**Relay/ Routed Unicast UDP**

**Broadcast UDP**

Vlan7

Vlan5

Vlan6

**PC**

Switch

Router

**UDP Server1**

**192.168.35.16**

**UDP Server2**

**192.168.58.21**

**Relayed/Routed**

**Unicast UDP**

**Relayed/Routed**

**Unicast UDP**

10.10.10.0/24

192.168.58.0/24

Figure 6: UDP port relay command in L3 mode and a different IP address configured

Configuration:

ip udp relay 5001

ip udp relay 5001 address 192.168.58.21

ip udp relay 5001 address 192.168.35.16

Current Enhancement supports only one relay ip address per UDP port. Hence when different IP address is configured for the same UDP port, then switch will simply override it. So the udp packets will be relayed to UDP server 1 instead of server 2.

### Configuration commands

A new parameter is added to the existing CLI command to support this feature.

#### [vrf <vrfname>] ip udp relay {port <port> | service [TFTP TACACS NTP NBNS NBDD DNS]} {vlan<vlan\_id>| address <ipv4address>}

This command specifies UDP server address on which traffic destined for a UDP port is forwarded as unicast packets.

**Syntax Definitions**

*port*  A user specified port that is not a well-known port

*vlan\_id* A numeric value(1-4094) that uniquely identifies an individual VLAN.

*ipv4address* UDP server address to which the UDP packets are destined.

**Usage Guidelines**

* Use the no form of this command to remove the ip address associated with the UDP service port.
* Existing well known services supported by UDP Relay are TFTP TACACS NTP NBNS NBDD DNS.
* Only specify service port numbers that are not well known when using the port parameter with this command. For example, do not specify port 53 as it is the well-known port number for the DNS UDP service. Instead, please use the DNS parameter to enable relay for port 53.
* Only one IP address can be configured per UDP service port.

**Examples**

ip udp relay service tftp address 10.1.1.2

vrf vrf-one ip udp relay port 5001 address 20.1.1.2

**Changes to show commands.**

The show commands will be modified as below.

Switch > **show ip udp relay**

Service Name Port IP Address Vlans

---------------------+------+---------------+-----------------------------

TFTP 69 10.1.1.2

UDP port 1122 1122 1 20 30

***Sample output of the* show ip udp relay statistics *CLI command:*\***

Switch > **show ip udp relay statistics**

Port  Service        Pkts Recvd Pkts Sent Dst Vlan/ IP Address   
-----+--------------+----------+--------+---------------

69  TFTP 0 0 10.1.1.2

1122 UDP port 1234           0   
                                       0          2   
                                       0          3   
                                       0          4

**Related MIB Objects**

Pls refer to the MIB below.

**Error Messages**

"Vlan cannot be associated as UDP port %d is in L3 mode"

"IP address cannot be associated as UDP port %d is in L2 mode"

**MIB Changes**

The following table will be added to the existing MIB for UDP Relay:-

alaGenericUdpServiceDstIpTable OBJECT-TYPE

SYNTAX SEQUENCE OF GenericUdpServiceDstIpEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table defines the destination IP for the

Generic UDP Relay Service. UDP packet with destination port

alaGenericUdpServiceDstUdpPort are forwarded to the IP defined in

alaGenericUdpServiceDstIpAddress."

::= { genericUdpServiceMIB 4 }

alaGenericUdpServiceDstIpEntry OBJECT-TYPE

SYNTAX GenericUdpServiceDstIpEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in Generic UDP Relay destination Ip table."

INDEX {

alaGenericUdpServiceDstUdpPort,

alaGenericUdpServiceDstIpType,

alaGenericUdpServiceDstIpAddress

}

::= { alaGenericUdpServiceDstIpTable 1 }

GenericUdpServiceDstIpEntry ::= SEQUENCE {

alaGenericUdpServiceDstUdpPort

Unsigned32,

alaGenericUdpServiceDstIpType

InetAddressType,

alaGenericUdpServiceDstIpAddress

InetAddress,

alaGenericUdpServiceDstStatTxToIp

Counter32,

alaGenericUdpServiceDstIpRowStatus

RowStatus

}

alaGenericUdpServiceDstUdpPort OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"UDP port for the service."

::= { alaGenericUdpServiceDstIpEntry 1 }

alaGenericUdpServiceDstIpType OBJECT-TYPE

SYNTAX InetAddressType { ipv4(1), ipv6(2) }

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This specifies the IP Address type of the

UDP Relay Server to forward the UDP packet."

::= { alaGenericUdpServiceDstIpEntry 2 }

alaGenericUdpServiceDstIpAddress OBJECT-TYPE

SYNTAX InetAddress (SIZE(4|16))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This specifies the IP Address of the

UDP Relay Server to forward the UDP packet."

::= { alaGenericUdpServiceDstIpEntry 3 }

alaGenericUdpServiceDstStatTxToIp OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This keeps track of the number of packets transmitted to the

destination IP with UDP destination port matching

alaGenericUdpServiceDstUdpPort."

::= { alaGenericUdpServiceDstIpEntry 4}

alaGenericUdpServiceDstIpRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Row Status for creating/deleting the Generic Service Destination IP Table"

::= { alaGenericUdpServiceDstIpEntry 5 }

Please note that the alaGenericUdpServiceDstIpType is added for extensibility and needs to be IPv4 for now (only IPv4 address is currently supported).

**Webview Changes**

UDP relay web page is modified such that for an UDP port, either VLAN or IP address can be entered. Please note that only one IPv4 address is supported per UDP port.

**Design Changes**

The following changes will be needed in the existing data structures:-

**typedef struct**

**{**

**uint32 Ip;**

**uint32 ipTxCnt;**

**} DST\_IP\_CTL\_BLK**;

typedef struct

{

RBT\_NODE node;

uint32 udpPort;

uint32 rxFromClientCnt;

RBTREE dstList;

**DST\_IP\_CTL\_BLK relayIP; // relayIp.Ip non-zero in the case of L3 mode**

char name[MAX\_DESCRIPTION\_SIZE];

} UDP\_PORT\_CTL\_BLK;

The existing UDP\_PORT\_CTL\_BLK, obtained under &vrfCtlBlk->udpRelay.genericService, for the given VRF, UDP port, will be enhanced to include a server IP address (relayIp.Ip). This will be initialized to 0 and set only if we operate in the L3 mode (i.e. no vlan configured to relay the UDP packets to).

UDPRelay\_genericUdpServiceDstIpTblGet/ UDPRelay\_genericUdpServiceDstIpTblGetNext/ UDPRelay\_genericUdpServiceDstIpTblTest/ UDPRelay\_genericUdpServiceDstIpTblSet need to be written for the new table.

**Changes in genericHandleRequest() in UDP Relay CMM:-**

1. Check if there is any destination vlan configured. If yes the earlier flow holds good, if not proceed to step 2.
2. Decrement the TTL value in the IP header.
3. If TTL value is less than 1, drop the packet.
4. Retain the source IP address in IP header.
5. Stamp the IP destination address as the address configured for the UDP service.
6. Calculate checksum for IP header.
7. Send the packet to IPNI with a new type UDPRELAY\_ROUTED\_DATA\_TO\_IPNI. Pls note that this is different from UDPRELAY\_UNICAST\_DATA\_TO\_IPNI as the outgoing gport, gateway mac is not known and a route lookup is needed to the configured relay IP (in IPNI), followed by an arp lookup to the gateway.
8. Increment the ipTxCnt counter.

**Changes in IPNI:-**

A new type UDPRELAY\_ROUTED\_DATA\_TO\_IPNI has to be handled in IPNI (ipni\_udprelay\_engress\_pkt()), which will do a route and arp lookup and send out the packet (pls refer to ipni\_ipoutput()).

The following message will need to be enhanced to include the VRF (used only for UDPRELAY\_ROUTED\_DATA\_TO\_IPNI type). The destination IP would be populated in the payload.

typedef struct \_udp\_relay\_egress\_pkt\_s

{

MSG\_HDR\_S header;

uint32 srcGlobalPort;

uint32 dstGlobalPort;

int32 pkt\_length;

uint16 vlanId;

**uint16 vrf;**

uint8 macData[0];

}UDP\_RELAY\_EGRESS\_PKT\_S;

# APPENDIX F: 7.3.4.R01 DHCP Snooping

# DHCP Snooping – RTR 3783

## Introduction

DHCP Snooping feature is ported from 7.X.X.R01 (8.1.1.R01) to 7.3.4.R01; so it will be same as 8.1.1.R01 release as explained in APPENDIX D.

On top if the changes in 8.1.1.R01 following design change is done in 7.3.4.R01

## Design Change

UDP Relay task /DHCP Snooping shall process only packet with UDP src/dst port 68/67 and 67/68, expect the case where the destination IP address is switch IP address(Relayed reply or internal DHCP server packet) packet with UDP src/dst 67/67 port shall be processed in software. UDP port 67/67 is not trapped to CPU anymore. This is done to make sure binding entry is created only on the client VLAN, not on a L3 domain if snooping is enabled where packets shall be already relayed/routed as explained in Fig 9.2.a and Fig 9.2.b

In the below two examples since snooping is not enabled in VLAN 10 where client is connected, all the unicast renew and release DHCP packet shall reach switch B with UDP src/dst port 68/67. Here since snooping is enabled on Switch B, we don’t want to create binding entry for VLAN 10 subnet in Switch B under VLAN 20. So switch B should bridge the packet in case 1(Fig 9.2.a) and route the packet in case 2(Fig 9.2.b).

Switch A

Router

Switch B

L2 Snooping

DHCP

Client

DHCP Server

Vlan 10

Vlan 20

Vlan 20

Fig 9.2.a

Switch A

Router

Switch B

Snooping + Router

DHCP

Client

DHCP Server

Vlan 10

Vlan 20

Vlan 30

Fig 9.2.b

To identify whether a packet need to be transparently bridged or routed following logic is used:

If the destination IP address is not switch IP address or broadcast IP address and if the UDP src/dst port is 68/67or 67/68 and if Ethernet source address and client address present in DHCP header does not match then

* IF destination mac-address is router address

Then route the packet

* Else

Then bridge the packet

Since binding entry is not created on L3 Domain, if ISF is enabled then IP packets for the client VLAN shall be dropped, so even the client port needs be made as trusted port when snooping is enabled after a router.

Also if snooping enabled on a L3 domain and if snooping is not enabled on the actual client vlan, then the unicast DHCP Renew and Release packet from the client shall be sent directly to DHCP server with UDP port 68/67, since those packet shall be trapped to software, those packet shall be transparently bridged/routed to the DHCP server/client by the UDP Relay/DHCP Snooping task as illustrated above. Even though its bridge/routed transparently, since snooping is enabled, software shall allow only server packets on trusted ports. Also transparent bridging/routing will be allowed only on trusted ports, if the either client or server port is not trusted then those DHCP packet shall be dropped and mac-verification error counter of that port shall be incremented. Irrespective of mac-verification status if the port is not a trusted port then those packets identified for transparent bridging/routing shall be dropped and mac-verification error counter shall be incremented. This makes mandatory for all the interconnected switch ports in L3 Domain on which DHCP packet shall be received irrespective of client or server type, to be configured as trusted ports.

This limitation is added to avoid spoofing, if it is not added then anyone in the network can spoof a DHCP release packet for the entire host in the network. As a result to avoid spoofing, we allow transparent bridging only on trusted ports.

AOS supports PORT level IP Source Filtering (ISF) and VLAN level IP Source Filtering (ISF), but starting 7.3.4.R01 Port Level ISF and VLAN level ISF will be mutually exclusive. User shall be allowed to configure only either Port level ISF or VLAN level ISF at any given time. This is to avoid confusion on whether to give preference to Port level ISF or VLAN level ISF when both are configured. Also since we have only limited number of TCAM entries available for Ip Source Filtering, enabling both on multiple port/vlan will reduce the number user available of TCAM entries because of default entries created for VLAN/Port. Currently we don’t anticipate any customer using both VLAN level ISF and Port level ISF together. This shall be taken into consideration in a later release if there is requirement for supporting for both together.

## Database file synchronization

Since 7.3.4.R01 supports support chassis based product, code have been added to synchronize the DB file from primary to secondary. During the DB file sync timeout master primary will copy the DB file to Master secondary CMM from the Mast primary itself after sending message to all salve primaries initiate copy of the DB file from the mast primary. After copying the DB file from the primary master chassis, all the slave primary CMM will again try to copy the its own secondary CMM if exists.

Binding entries for the NI’s that are not present/UP due to some power or software failure during system ready were getting deleted or not populated from the binding database file. To avoid this new inactive binding database is created and during system ready all the entries for which NI is not present shall be added to this list and only entries for which NI up will be added to active binding list. Show command shall use the active binding list to display the binding entry. Once this NI comes-up those entries will be moved from inactive list to the active binding list and available for display in the CLI. If a particular NI is not up during the entire boot-up time of the switch or any takeover happened before the movement from binding entry from inactive list to active list, then upon subsequent boot-up/takeover those binding entries shall get lost because the binding database file in the flash shall be created using the active binding list not using the inactive binding list.

Binding timer shall also decrement the lease time in the inactive binding list and if the lease of a binding entry gets expired before a NI joins back (before movement of binding entry from inactive list to active list), those entries shall be removed from inactive binding list.

## Bug fixes after 8.1.1.R01

DHCP snooped packet that are forwarded by CPU/(From CMM) was getting learned on the CPU port itself instead of the actual port in which it was received in the initial 8.1.1.R01 code. This has been fixed in the code so that packet forwarded from CPU that is received on front panel port will be learned on the actual user port in which it was received instead of CPU port.

DHCP packets received on STP blocked ports were also getting trapped to CPU in the initial 8.1.1.R01 code and hence duplicate DHCP packets were generated. This has been now changed in QOS to accept DHCP packet only on STP forwarding ports.

For **Ingress Source Filtering** (ISF), accept entries for the binding entry is created for per port individually, So when the binding entry need to created on a linkagg port, it was getting created only on linkagg primary port. Code has been modified to add one entry for each member ports of the linkagg. This will in-turn reduce the number of ISF enabled bind entry created on a NI since QOS resource(FFP entry) are wasted by creating one entry for each member ports of the linkagg. **This code can be optimized to use the port bitmap in the QOS hardware entry instead of using one entry per linkagg member port in future release to avoid QOS resource wastage.**

## Recommendation

1. It always recommended enabling DHCP Snooping on the edge where the client’s are connected.
2. DHCP snooping shall be avoided after a relay agent, unless it is mandatory for any particular network design.

## Limitation

1. User need to manually add QOS user port rule for all client ports to drop any DHCP server packets from the client ports, to avoid rouge DHCP server on a core network (after relay agent).
2. Show command for binding entry shall display only the active (Port’s/NI’s up) the binding entry.
3. Port level ISF and VLAN level ISF are mutually exclusive.
4. DHCP-snooping ip-source-file port X/X/X (ISF enabled on port level) will apply to all the VLANs present on this port.

## Sample Use case to explain new Design

Switch A

Router

Switch B

L2 Snooping

DHCP

Client

DHCP Server

Vlan 10

Vlan 20

Vlan 20

In the above setup, since VLAN 10 is terminated in Switch A, Switch A should act as relay agent for VLAN 10.

When DHCP client sends the DHCP request, it will have UDP port 68/67, But when Switch A, relay the packet to the DHCP server, UDP port will be changed to 67/67.

In the new design change, even if DHCP snooping is enabled in Switch B, this DHCP request packet  will not be trapped o software in Switch B, because DHCP request packet will have  UDP port 67/67. It will be switch in hardware itself. So even if you don’t configure the DHCP server port as trusted port, client will still get IP address. Snooping happens only on the client VLAN not in any other VLAN in the L3 domain.

Here the problem is, anyone can spoof the packet with 67/67 and by-pass the snooping (Though it can be done only for edge subnet in the core network and granting access to someone who is going to spoof in core network is not a practical design). That’s why we recommend to configure QOS Userport policy to drop all the DHCP serve packet (any UDP packet with src port 67) on all the client only port. This is what listed in the limitation section.

So now client will get an IP address, irrespective of ports are trusted or not in Switch B (assuming QOS Userport policy is not configured). Now when the DHCP client sends DHCP renew packet or Release packet, DHCP client will directly send unicast packet to the server address where as all the previous transaction would have been to broadcast address, since client did not had an IP address. Since snooping is not enabled on Switch A, these unicast packet shall be routed in hardware itself in Switch A , so packet will still have UDP port 68/67 instead of 67/67(when it was relayed by Switch A) when it reached Switch B, since snooping is enabled, this packet shall be trapped to software.

If we do normal snooping process, this will end up creating binding entry for DHCP Client (which is on VLAN 10) on VLAN 20 in switch B, which we wanted to avoid. So we wanted to do transparent bridging/routing in software for this case, where mac-address for Ethernet header and client packet is not same and destination IP address is not switch IP address or broadcast IP address.

But if we do that for packets from all the ports, then assume if snooping is enabled on Switch A, and someone in connected to Switch A, sends DHCP release packet to all the clients on VLAN 10(since he/she is already in VLAN 10, he/she will have access all the IP address and DHCP server information) , it will end up in a bigger problem.

So we will do this transparent bridging only on administrator configured trusted ports.

Another hidden problem is when ISF is enabled on Switch B, client in VLAN 10, will not have access to the network , But when the port connected to Switch A is configured as trusted port, that problem will be automatically solved, since we don’t allow ISF on trusted ports.

Appendix G: PVLAN support for DHCP-Snooping & ISF in 8.3.1.R01

# DHCP Snooping

## Introduction

DHCP Snooping will be supported over a private vlan (PVLAN) in 8.3.1.R01 release. This effort includes support for

1. DHCP-Snooping of IP address that are seen on the switch/router in PVLAN.

2. Port/VLAN based Ingress Source Filtering (ISF) in PVLAN.

### Referenced Documents

* 012316-00\_AOS741R01\_Software\_SRS\_PramodaNallur\_rev\_10\_11\_10\_2014\_AgileRev\_B3.docx
* 014270-00\_831R01\_RTR-3400\_PVLAN\_SFS\_Rev.1.0\_19Jun2015\_AgileRev.1

### Platforms supported

8.3.1.R01 supports Medora (OS9900), OS6900,(Shasta) OS6860, OS10K ,Everest (OS6865), OS6560 (Nandi) and OS6465.

## Requirements Overview

### DHCP-Snooping support in PVLAN

#### When DHCP-Snooping is enabled/disabled on primary PVLAN then it has to propogate to all the secondary PVLANS (implicit enable)

#### Don’t allow to explicitly configure/ unconfigure dhcp-snooping on Secondary PVLAN

#### Whenever Primary PVLAN deletion occurs disable dhcp-snooping on primary and all secondary PVLANS

#### Whenever secondary Pvlan gets added to primary PVLAN then enable the dhcp-snooping on secondary if enabled on the primary

#### DHCP-snooping Binding entries for the secondary PVLAN has to get created is enabled on the primary PVLAN

### Ingress Source Filtering in PVLAN

In order to support ISF on PVLAN. We program the hardware with the secondary vlan id (if enabled on the primary).

#### When ISF is enabled/disabled on primary PVLAN then it has to propogate to all the secondary PVLANs (impilict enable)

#### Don’t allow user to configure/ unconfigure ISF on Secondary PVLAN

#### Whenever primary PVLAN deletion occurs disable ISF on primary and all secondary PVLANs as well

#### Whenever secondary PVLAN is addedenable ISF on secondary PVLAN if enabled on the primary vlan

### Hardware Requirements and Limitations

No new hardware requirements exist.

### Software Requirements and Limitations

* The maximum number of DHCP Snooping enabled VLANs (including normal vlans, PVLANs including primary and secondary private vlans)is 64.
* The maximum number of vlans (including normal vlans, PVLANs including primary and secondary private vlans) on which VLAN level ISF can be enabled is 32
* Maximum of 256 FP entries supported per NI across all platforms as earlier.

### Security Requirements

No new security requirements exist.

### Redundancy and Hot Swap Requirements

Must support existing takeover and VC-takeover requirements as earlier.

### Assumptions

Please note that the ISL port(s) has to configured as a trusted port if the server is reachable via promiscuous port on another chassis.

## Design Changes

### Basic Architecture

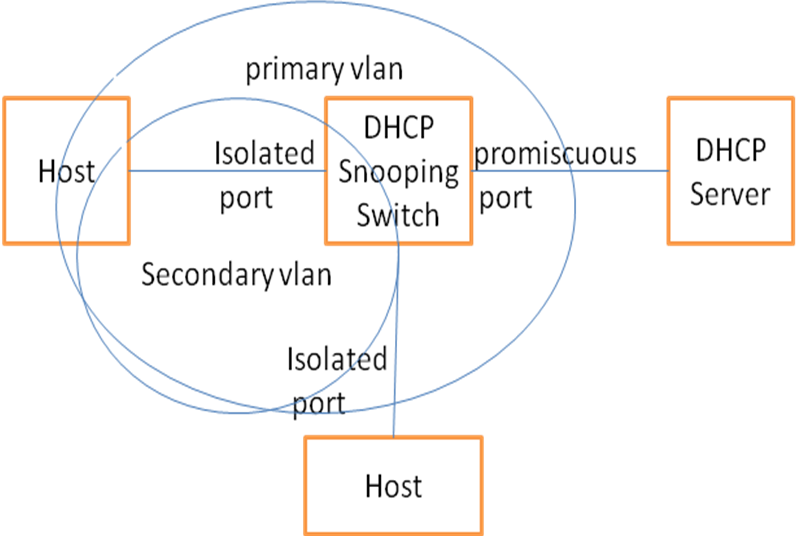
There is DHCP snooping functionality changes, such as binding table database for the dhcp-snooping will contain the secondary vlanId instead of the primary vlanid for PVLANs. And also ISF programs the secondary VLANs to the hardware. Whenever dhcp request packet is received on secondary vlan then the packet will be sent on the corresponding primary vlan trusted ports. In the same way mapping happens from primary to secondary when dhcp reply packets received on primary vlan.

#### Interface with NI modules

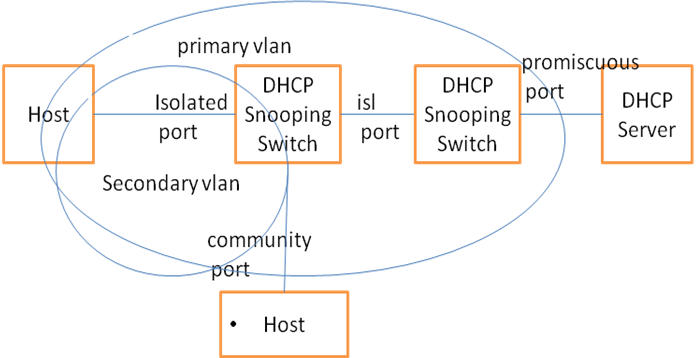
No change in interfacing with NI modules like IP-NI.

## Use Cases

### When dhcp server is connected to promiscuous port



### When dhcp server is connected to promiscuous port of other switch



# APPENDIX H: 8.3.1.R02 DHCP Snooping

DHCP Snooping feature is supported on 8.3.1.R02 as part of RTR: 4355 on platform 9900.

# Appendix I: RTR 4513, 5138 New CLI command to display DHCP-Snooping Binding entry for 8.4.1.R01, Config Snapshot in 85R1

# DHCP Snooping Binding entry

**Description:**

Binding entries are stored in a file and only display in a table format.

For specific reasons, there is a need to display the entries in the snapshot format so that the content can be copied-pasted for reconfiguration when needed.

**Requirements:**

New command is required to display the binding table with option to filter static and dynamic entries.

Example:

-> show dhcp-snooping binding [snapshot] [static|dynamic]

 dhcp-snooping binding 11:22:33:44:00:00 port 1/1 address 1.1.1.1 vlan 10  
 dhcp-snooping binding 11:22:33:44:11:11 port 1/2 address 1.1.1.2 vlan 10  
 dhcp-snooping binding 11:22:33:44:22:22 port 1/3 address 1.1.1.3 vlan 10  
 dhcp-snooping binding 11:22:33:44:33:33 port 1/4 address 1.1.1.4 vlan 10

Also the config snapshot would have the static binding entry displayed, which can be written to vcboot.cfg, starting 85R1

-> show configuration snapshot dhcp-snooping

! DHCP Snooping:

dhcp-snooping binding aa:aa:aa:aa:aa:aa port 1/1/1 address 1.1.1.3 vlan 80

dhcp-snooping binding admin-state enable

# Appendix J: RTR 4975 Requirements and Limitations for 6560

The below features will be available on 6560 (Nandi) as part of 841R02.

* Maximum number of next hop IP address or VLAN/next hop pair is limited to number of IP interface configurable. For Nandi (OS6560) this limit is 128.
* Maximum number of Generic UDP Relay Service is **30**.
* Maximum number of vlans enabled with dhcp-snooping are 64.

ISF will be supported on 6560 from 85R1 onwards with the following numbers:-

* Maximum number of vlans enabled with ip-source-filtering are 32.

The Maximum number of binding entries for ISF are as per the table mentioned below.

|  |  |  |
| --- | --- | --- |
|  | Vlan Level ISF | Port Level ISF |
| 6560 with  24 port | 32 Vlans with 96 Clients | 127 clients |
| 6560 with 48 Port | 16 Vlans with 96 Clients | 111 clients |

Please note that the scalability numbers are subject to TCAM entry availability due to resource contention.

# Appendix K: RTR 4975 Requirements and Limitations for 6465

The below features will be available on 6465 as a part of 85R1.

* Maximum number of next hop IP address or VLAN/next hop pair is limited to number of IP interface configurable. For OS6465 this limit is 8.
* Maximum number of Generic UDP Relay Service is **30**.
* Maximum number of vlans enabled with dhcp-snooping are 64.

The below features will be available on 6465 as a part of 85R3.

* Maximum number of vlans enabled with ip-source-filtering are 16.

The Maximum number of binding entries for ISF are as per the table mentioned below.

|  |  |  |
| --- | --- | --- |
|  | Vlan Level ISF | Port Level ISF |
| 6465 | 16 Vlans with 48 Clients | 62 clients |

Please note that the scalability numbers are subject to TCAM entry availability due to resource contention.

Because of the tcam architecture the interaction of dynamic binding entries with ISF enabled takes a delay to program in the hardware via tcam mgr, with the scalability number the delay is considerable amount (around 10 sec).

# Appendix L: Enhanced DHCP Snooping Troubleshooting (RTR 5149, 5150 in 85R2).

## INTRODUCTION

### Purpose

This document is prepared in response to the RTR 5149, 5150: Clear command for DHCP snooping violation counters and enhanced DHCP snooping troubleshooting. This document provides proposed design approach for implementing the requested functionality as outlined in the SRD.

### Scope

The scope of this document is to present the complete functional description of RTR 5149, 5150: Clear Command for DHCP Snooping violation counters and Enhanced DHCP Snooping Troubleshooting to be implemented in 85R2 release. This includes the enhancement of troubleshooting for IPv4 DHCP snooping. The intended audience for this document includes the engineering design team and the product test team.

### Intended Audience

This document is intended for the following audience:

* Engineering Design team.
* Product Testing Team.

### Referenced Documents

#### Internal Documents

* SFS, SRD documents
* 014949-00\_AOS-8.5R2-SW-SRD-PN-SM-v3\_01Feb2018\_AgileRev.3 (002)
* 011524-07\_672R02\_RTR-4859\_UDP\_Relay\_DHCP\_Snooping\_SFS\_rev\_5\_4\_14Jul2017\_AgileRev.B

#### External Documents

## FUNCTIONAL DESCRIPTION

### Basic Overview

Current implementation of AOS does not support CLI command that can help in detailed debugging of DHCP snooping feature. In production network, troubleshooting DHCP and ISF is cumbersome as not much information is available from CLI. The current debug logs available with swlog and systrace are not much helpful since it prints too many logs and is not easily interpretable to an end user. This enhancement focuses on providing new CLI debug and show commands that can help an end customer to troubleshoot the DHCP snooping feature easily. This enhancement addresses the below scenarios.

## FUNCTIONAL REQUIREMENTs.

This Section specifies the functional requirements added as a part of this enhancement.

DHCP-DEBUG-10**:** DHCP snooping feature displays the violation counters with “*show dhcp-snooping port”* command. A clear command is added to clear these counters.

DHCP-DEBUG-20**:** A debug command is added to globally enable on demand debugging of DHCP-Snooping for a specific client. A CLI command is added to specify the client for which debugging needs to be enabled. User can specify a particular MAC address or a port or a linkagg on which monitoring should be done.

If a linkagg or port is specified, all clients on this linkagg or port will be debugged. At a time, debugging will be done only on the last specified port/linkagg/MAC. A combination of this will not be logged.

DHCP-DEBUG-30**:** The above debug messages needs be stored in a buffer in a user readable format. A CLI command will be added to enable dumping of DHCP packets to the log buffer before and after processing.

DHCP-DEBUG-40**:** A show command will be provided to display the logs.

DHCP-DEBUG-50**:** A clear command will be provided to clear the log buffer.

DHCP-DEBUG-60**: A** CLI show will be provided to display the global counters that contains the number of different types of DHCP packets received/processed/transmitted/dropped by the switch, on the CMM.

DHCP-DEBUG-70**:** A clear command will be provided to clear the above global counters.

DHCP-DEBUG-80**:** The ISF drop count needs to be displayed in CLI**.**

DHCP-DEBUG-90**:** Swlog needs to be added for ISF drops. Switch log entry is limited to one per hour.

DHCP-DEBUG-100**:** SNMP trap messages needs to be added for ISF drops. Trap entry is limited on one per hour.

DHCP-DEBUG-110**:** Swlog needs to be added for binding entry creation failure due to TCAM resources unavailability.

DHCP-DEBUG-120**:** SNMP trap messages needs to be added for binding entry creation failure due to TCAM resources unavailability.

## PLATFORM SUPPORTED

OS6900, OS6860, OS6865, OS9900, OS6560 & OS6465

## MANAGEMENT INTERFACE

### Command Line Interface (CLI)

<List down the CLI commands implemented for this feature and its relevant MIB objects>

#### : Clear command for DHCP snooping violation counters.

*-> dhcp-snooping clear violation-counters {all | chassis <cid> slot <num> | linkagg <num> | port <chassis/slot/port>}*

Syntax Definitions

all Clear DHCP snooping violation counters on all ports.

<chassis */*slot/port> Clear DHCP snooping violation counter for the specified physical port.

chassis*/*slot <num> Clear DHCP snooping violation counter for all port of the specified slot.

linkagg <num> Clear DHCP snooping violation counter for the specified

linkagg.

SNMP:

SNMP support will be available for this command.

#### : On demand debugging for a given port or client MAC-address.

***->*** *debug dhcp admin-state {enable | disable}*

Syntax Definitions

enable Enable DHCP on demand debugging globally.

disable Disable DHCP on demand debugging globally.

***->*** *debug dhcp {mac-address <MAC-ADDRESS> | port <chassis/slot/port> | linkagg*

*<LINKAGG-ID>} | [vlan <VLAN-ID>]*

Syntax Definitions

mac-address Specifies the client for which the transactions needs to be

logged.

port Specifies the port on which DHCP transactions should be

logged.

linkagg Specifies the linkagg on which DHCP transactions should be

logged.

Vlan Specifies the VLAN on which the clients are connected. If vlan

is not specified, clients on all VLANs on the specified

port/linkagg/MAC will be logged.

Example: If mac-address is specified as 11:11:11:11:11:11 and

VLAN as 100.The DHCP transaction done by a client with mac-

address 11:11:11:11:11:11 and belong to VLAN 100 only will

be logged. If a client with same MAC address came on another

VLAN, that client’s transaction will not be logged.

If VLAN 100 was not specified, then both the clients will be

logged.

Note: VLAN option have limitations in DHCP relayed network

since relay transfers DHCP operation across VLANs.

***->*** *debug dhcp dump-packet admin-state {enable | disable}*

Syntax Definitions

enable Enables DHCP packet dump for the clients which are monitored.

disable Disables DHCP packet dump for the clients which are monitored.

***->*** *debug show dhcp*

Syntax Definitions

This command displays the logs collected. Maximum log line length will be 160 characters and maximum number of logs will be 600 lines.

Example:

-> debug dhcp admin-state enable

-> debug dhcp mac-address 11:22:33:44:55:66 vlan 100

-> debug dhcp dump-packet admin-state enable

-> debug show dhcp

Debug Configurations:

------------------------------------------------------

DHCP Debug : Enabled

DHCP Packet Dump : Enabled

Debugging on : MAC: 11:22:33:44:55:66

Debugging on VLAN : 100

DHCP Snooping Status : Switch Level Enabled

Date Time Log Message

---------+----------+------------------------------------------------------------------------------

12/18/00 22:18:39 DHCP Discovery: Received in DHCP Application:Port:1/1/47, Mac:11:22:33:44:55:66 VLAN:100

12/18/00 22:18:39 DHCP Packet Dump:

ff ff ff ff ff e8 e7 32 76 85 a4 81 00 00 64 08 00 45 00 01 48 00 00 00

00 40 11 f2 e0 c3 c3 c3 01 ff ff ff ff 00 43 00 44 01 34 6f 5d 02 01 06

00 00 00 00 00 00 00 00 00 00 00 00 00 c3 c3 c3 0a c3 c3 c3 01 00 00 00

00 e8 e7 32 1f 1d 9e 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00

1/18/15 2:12:40 DHCP Discovery: Processing Successfull: Port:1/1/47, Mac: 11:22:33:44:55:66 VLAN: 100

1/18/15 2:12:40 DHCP Discovery Received From: Port 1/1/47 Mac: 11:22:33:44:55:66 VLAN: 100

***->*** *debug dhcp clear log*

Syntax Definitions

Clears the debug log buffer.

SNMP:

SNMP support will not be available for these debug commands.

### Command to display/ clear the DHCP snooping/ Relay global counters.

*-> dhcp-snooping clear counters*

Syntax Definitions

This command clears the global counters for DHCP snooping/Relay.

*-> show dhcp-snooping counters [chassis <cid> slot <slot-id>]*

Syntax Definitions

This Command Displays the DHCP snooping/Relay global counters.

NI counters display the cumulative value from all NIs.

Example:

-> show dhcp-snooping counters

DHCP Discover Packets : 2,

DHCP Offer Packets : 0,

DHCP Request Packets : 0,

DHCP ACK Packets : 0,

DHCP NACK Packets : 0,

DHCP Release Packets : 0,

DHCP Decline Packets : 0,

DHCP Inform Packets : 0,

Total Packet received in CMM : 2,

Binding error (TCAM Unavailable) : 0,

Unknown/Malformed Packets Dropped : 0,

Packets received in CMM : 0,

Packets transmitted from CMM : 0,

Total ISF Packet Drop : 0

-> show dhcp-snooping counters chassis 1 slot 2

Packet received in CMM from NI : 0,

Packet transmitted from CMM to NI : 0,

Total ISF Packet Drop : 0,

SNMP:

SNMP support will not be available for show commands. But will available for clear commands.

### Commands for ISF statistics.

*-> dhcp-snooping clear isf-statistics*

Syntax Definitions

This command clears the isf drop counters.

*-> show dhcp-snooping isf-statistics [vlan <vid>]*

Syntax Definitions

This command displays the ISF drop counters.

Example

-> show dhcp-snooping isf-statistics

Mode: Vlan based ISF

Chassis Slot Vlan Packets Dropped

--------+--------+---------+-----------------

1 1 10 300

1 2 10 400

1 3 11 500

1 4 11 600

-> show dhcp-snooping isf-statistics vlan 10

Mode: Vlan based ISF

Chassis Slot Vlan Packets Dropped

--------+--------+---------+-----------------

1 1 10 300

1 2 10 400

-> show dhcp-snooping isf-statistics

Mode: Port based ISF

Chassis Slot Vlan Packets Dropped

--------+--------+---------+-----------------

1 1 NA 300

1 2 NA 400

2 1 NA 500

2 2 NA 600

If ISF statistics command executed on platform on which it is not supported below output will be thrown on the console.

-> show dhcp-snooping isf-statistics

Note: ISF Statistics command not supported on 6465 Platform

SNMP:

SNMP support will not be available for these debug/ show commands.

-> debug dhcp isf-statistics trap-duration <minutes>

Syntax Definitions

This debug command sets the ISF trap duration in minutes, minimum being 2 minutes and default being 60 minutes. This is intended only for the regression framework (PAL).

Example

-> debug dhcp isf-statistics trap-duration 5

Note: ISF Trap Set for 5 minutes.

### SNMP/MIBs DHCP SNOOPING CLEAR COUNTERS:

dhcpSnoopingClearCounters OBJECT-TYPE

SYNTAX INTEGER {

default(0),

reset(1)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Used to clear all DHCP snooping statistics counters.

By default, this object value is zero."

DEFVAL { default }

::= { dhcpSnoopingMIB 14 }

dhcpSnoopingClearViolationTable OBJECT-TYPE

SYNTAX SEQUENCE OF DhcpSnoopingClearViolationEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table is defined to clear DHCP snooping violation counters

on specific interface."

::= { dhcpSnoopingMIB 15 }

dhcpSnoopingClearViolationEntry OBJECT-TYPE

SYNTAX DhcpSnoopingClearViolationEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Each entry defines variables used to clear violation statistics for given

interface index"

INDEX { dhcpSnoopingClearViolationIfIndex }

::= { dhcpSnoopingClearViolationTable 1 }

DhcpSnoopingClearViolationEntry ::= SEQUENCE {

dhcpSnoopingClearViolationIfIndex

InterfaceIndex,

dhcpSnoopingClearViolationAction

INTEGER

}

dhcpSnoopingClearViolationIfIndex OBJECT-TYPE

SYNTAX InterfaceIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The interface of which DHCP snooping violation counters are cleared.

The maximum value 2147483647 means all interfaces."

::= { dhcpSnoopingClearViolationEntry 1 }

dhcpSnoopingClearViolationAction OBJECT-TYPE

SYNTAX INTEGER {

default (0),

reset (1)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Used to clear DHCP snooping violation statistics."

DEFVAL { default }

::= { dhcpSnoopingClearViolationEntry 2 }

### SNMP TRAP FOR ISF COUNT (EVERY 1 HOUR):

A swlog entry and an SNMP trap message is added for ISF drop. The switch log will include the start and end time of the interval for which the drop count was taken and the number of packets dropped per Vlan/NI. This swlog and trap will be limited for one per hour. A timer will be added to trigger the swlog and trap every hour in case of ISF drop.

alaDhcpIsfDrop NOTIFICATION-TYPE

OBJECTS {

alaDhcpIsfDropIntervalStartTimeStamp,

alaDhcpIsfDropIntervalStopTimeStamp,

alaDhcpIsfDropCount,

alaDhcpIsfChassisID,

alaDhcpIsfSlotID,

alaDhcpIsfVlanID

}

STATUS current

DESCRIPTION

"Trap message to notify ISF drop."

::= { alaDhcpClientTrapsDesc 0 6 }

alaDhcpIsfDropIntervalStartTimeStamp OBJECT-TYPE

SYNTAX OCTET STRING

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"This object specifies the start time of this ISF drop

monitoring interval."

::= { alaDhcpClientTrapsObj 3 }

alaDhcpIsfDropIntervalStopTimeStamp OBJECT-TYPE

SYNTAX OCTET STRING

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"This object specifies the end time of this ISF drop

monitoring interval. This is the time at which the

trap message will be initiated."

::= { alaDhcpClientTrapsObj 4 }

alaDhcpIsfDropCount OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"This object specifies the number of ISF drop in the time

period specified by alaDhcpIsfDropIntervalStartTimeStamp

and alaDhcpIsfDropIntervalStopTimeStamp."

::= { alaDhcpClientTrapsObj 5 }

alaDhcpIsfChassisID OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"This object specifies the Chassis number on which packets are dropped."

::= { alaDhcpClientTrapsObj 6 }

alaDhcpIsfSlotID OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"This object specifies the Slot number on which packets are dropped."

::= { alaDhcpClientTrapsObj 7 }

alaDhcpIsfVlanID OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"This object specifies the Vlan number on which packets are dropped."

::= { alaDhcpClientTrapsObj 8 }

### SNMP TRAP FOR TCAM FAILURE:

Trap will be send if DHCP binding entry creation failed because of TCAM resource allocation failure. But the TRAP will be send for first time resource allocation failure, for subsequent failure it will be ignored.

alaDhcpBindingTcamFail NOTIFICATION-TYPE

OBJECTS {

alaDhcpTcamFailMsg,

}

STATUS current

DESCRIPTION

"Trap to notify DHCP Snooping resource allocation failure."

::= { alaDhcpClientTrapsDesc 0 5 }

alaDhcpTcamFailMsg OBJECT-TYPE

SYNTAX OCTET STRING

MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"This object specifies binding entry creation fail due to TCAM resource failure."

::= { alaDhcpClientTrapsObj 6 }

## Takeover/Reload.

Since this is a debug framework, the DHCP Snooping debug logs and counters will not be synced across CMMs. These details will not be saved in any file in the flash memory. So the configuration, logs and counters will not be persistent across takeover/reload. User need to configure the debug again after reload/ takeover. The logs will be reinitialized on Master Primary after takeover/reload.

## Scalability and Limitations

1. All logs are limited in size.A new buffer will be added to log the DHCP message. Once the buffer gets filled up, the older logs will get overwritten(old logs from the beginning will be removed & new logs will be add at the end). This buffer will store maximum 600 lines of log, each line can be of maximum 160 character in length.
2. This logging command is for packets that reached UDP Relay CMM. It is possible that the packet is actually dropped at a lower level. The packet could be dropped by ACL, LPS etc. in hardware. Also the packet may be received on the CPU but dropped in software context.
3. In port/ linkagg based debugging where the DHCP requests are supposed to be seen. The DHCP response will be received on another port and DHCP does not bind such response to the client port. It can only bind the DHCP to the client mac-address.
4. ISF drop counters and trap will not be available for OS6465, due to TCAM limitation wrt counters.

## Testability.

#### To test whether the clear command added for clearing DHCP snooping violation counter is clearing the counters properly.

#### To test whether the DHCP snooping global counters are getting displayed properly. Certain counters are for internal errors which may not get incremented in normal scenarios. Hence incrementing this counters while system testing may not be possible.

#### To test whether on demand debug logs are getting populated for the specified client. The logs added for error cases may not be verifiable at system testing since the internal errors occur very rarely.

#### To test whether swlog and SNMP trap is initiated for binding failure due to TCAM resource failure.

## WebView Support.

Since this enhancement includes only debug commands, no WebView support is provided for this enhancement.

## SNMP Support.

SNMP trap messages are added for ISF drop and binding failure due to TCAM resource failure.

Since this enhancement includes only debug commands, no SNMP get/set support is provided for this enhancement.

## SYSTEM DESIGN (APIs and Data Structures Added/Modified).

* + 1. **Clear command for DHCP snooping violation counters.**

The DHCP snooping violation counter values are stored in the data structure “*DhcpSnoopingPortEntry*” and updated in the RB Tree. A new API is added to clear these counter values. This API will accept a particular port/slot/port-list/linkagg for which the counter needs to be cleared.

MIP\_RETCOD clearUdpRelayDhcpSnoopingViolationCounters(int option, int slot\_agg, int port1, int port2)

This API will read the “*DhcpSnoopingPortEntry*” of the port-list/slot/linkagg specified by the user and will reset the below counter values.

portEntPtr->dhcpServerViolation = 0;

portEntPtr->macAddrViolation = 0;

portEntPtr->relayAgentViolation = 0;

portEntPtr->option82Violation = 0;

portEntPtr->bindingViolation = 0;

The below CLI wrapper function is added to invoke this API.

void WRPMiscClearDhcpViolationCounters(CliSubparserGlobals \*globals, int displayId, int option, int slot\_agg, int port1, int port2)

* + 1. **CLI Show command to display and clear the DHCP Snooping/Relay global counters.**

A new API is added to display the DHCP snooping Global counter values. The counter values displayed are software counters.

MIP\_RETCOD view\_udp\_relay\_dhcp\_snooping\_global\_counters()

The below CLI wrapper function is added to invoke this API.

void WRPMiscShowDhcpGlobalCounters(CliSubparserGlobals \*globals, int displayId)

Below counter variables are used in CMM to collect the respective values. The counters for the DHCP message types like discover offer etc. will be updated only if the packet reaches CMM. These counters are not synced to secondary CMM.

DHCP Discover : dhcpDiscoverCnt;

DHCP Offer : dhcpOfferCnt;

DHCP Request : dhcpRequestCnt;

DHCP Decline : dhcpDeclineCnt;

DHCP ACK : dhcpAckCnt;

DHCP NACK : dhcpNackCnt;

DHCP Release : dhcpReleaseCnt;

DHCP Inform : dhcpInformCnt;

DHCP Unknown Packets : dhcpUnknownPktDropped;

DHCP Malformed : dhcpSnoopingCorrUDPHdrLen;

DHCP packet with large size : dhcpPktSizeTooLarge;

DHCP binding failure due to TCAM resource failure: dhcpBindingFail\_TCAMresources;

In the show command, “Unknown/Malformed Packets Dropped” will be the sum of dhcpUnknownPktDropped, dhcpSnoopingCorrUDPHdrLen and dhcpPktSizeTooLarge.

Below counter variables are used in CMM to count the respective values.

Packet Tx b/w CMM and NI **:** dhcpPktFromCMMtoNI;

Packet Rx b/w CMM and NI **:** dhcpPktFromNItoCMM;

Below counter variables are used in NI to count the respective values.

Packet Tx b/w NI and CMM **:** dhcpPktFromNItoCMM;

Packet Rx b/w CMM and NI : dhcpPktsFromCMMToNI;

A global structure cmm\_dhcp\_ni\_global\_counters is created in CMM to store the counter values from each NI. Every time a dhcp packet come to CMM these counters will be increment. This data structure is not synced across CMM. Please note that the NI counters will be piggy backed with the existing IPC between IPNI and UDP Relay CMM – no additional polling or IPC is needed.

typedef struct dhcp\_ni\_global\_counters

{

uint32 cmm\_dhcpPktsFromCMMToNI;

uint32 cmm\_dhcpPktsFromNIToCMM;

uint32 ni\_dhcpPktsFromCMMToNI;

uint32 ni\_dhcpPktsFromNIToCMM;

}DHCP\_NI\_GLOBAL\_COUNTERS;

Number of packet processed but resulting to an error (i.e. not transmitted) for each of the internal events. This includes the internal errors in the switch like buffer unavailability, IPC/message failure etc. This count will not include drop due to packet errors, violations etc.

The below wrapper is added to clear the counter values.

void WRPMiscClearDhcpGlobalCounters(CliSubparserGlobals \*globals, int displayId)

This wrapper function will invoke the below function to clear the global counter values.

MIP\_RETCOD clear\_dhcp\_snooping\_global\_counters();

* + 1. **Swlog and SNMP trap for ISF drop.**

Switch log will be populated with appID “UdpRelay” with debug level “DEBUG2”.

SwLog Example:

-> show log swlog

Displaying file contents for '/flash/swlog1.log'

FILEID: fileName[/flash/swlog1.log],

Time Stamp Application Level Log Message

------------------------+------------+-------+-------------------------------------------

TUE DEC 19 19:22:54 2016 UdpRelay debug2 ISF: 300 packets dropped between 12-28-2016 18:22:54 to 12-28-2016 19:22:54

The below new API is added to generate the trap message. A timer will invoke this API if there is an ISF drop observed in that time interval. This API takes 3 arguments, the start time and the end time of the interval for which ISF drop was monitored and the number of drop observer in that time frame.

int UDPRelay\_alaDhcpIsfDropTrap\_Generate(char \*startTime, char \*stopTime, int32 dropCount)

Below APIs are added to implement the timer. The reload and expiry period will be one hour.

void dhcpSnoopingStartIsfTrapTimerCB();

void dhcpSnoopingAddIsfTrapTimer();

void dhcpSnoopingRemoveIsfTrapTimer();

The below API is added to invoke the trap message and add the swlog message, if an ISF drop occur in the one-hour time period. Every one hour the timer will get triggered and this API will be invoked. This

API will check whether there is any ISF drop occurred in this time interval. If there is an ISF drop a switch log will be generated and the API to generate the trap message will be invoked.

void dhcpSnoopingIsfTrapTimerHandler();

* + 1. **Swlog and SNMP trap for binding entry creation failure due to TCAM resources fail.**

A swlog entry and an SNMP trap message is added for binding entry creation failure due to TCAM resources fail. Code changes to invoke the trap message and swlog is done in the UDPRELAY CMM code. The log will contain the details, which are available from the client at that point of code flow.

Switch log will be populated with appID “udpRelay” with debug level “DEBUG2”.

SwLog Example.

-> show log swlog

Displaying file contents for '/flash/swlog1.log'

FILEID: fileName[/flash/swlog1.log],

Time Stamp Application Level Log Message

------------------------+------------+-------+-------------------------------------------

TUE DEC 19 21:03:54 2000 UdpRelay debug2 In DhcpSnoopingBindingTableSet:2853: Binding Failed: In-sufficient TCAM

TUE DEC 19 21:03:54 2000 UdpRelay debug2 [Count.]Resource. Slot: 1 Port: 10 MAC: 0-0-0-0-0-ff VLAN: 100

TUE DEC 19 21:03:54 2000 UdpRelay debug2 In dhcpSnoopingPostNiReadyConfigISF:6980: Binding Failed: In-sufficient TCAM

TUE DEC 19 21:03:54 2000 UdpRelay debug2 [Count.]Resource. Slot: 1 Port: 10

The below new API is added to generate the trap message. The below API fills the octet string variable in the trap message with a string that specifies TCAM failure.

int UDPRelay\_alaDhcpBindingTcamFailTrap\_Generate()

Message in the trap: char\* msg = "DHCP snooping binding failed due to TCAM resource failure.";

* + 1. **DHCP Snooping on demand debugging for a given port, linkagg or client MAC-address.**

This will log the DHCP events and transactions for the specified client or all clients on the specified port on a switch into a separate log buffer. There will not be any separate debug levels for this logs. This log will be stored in a buffer and will not be stored in files. So this logs will not be persistent across takeover and reload.

User needs to enable this feature globally and needs to specify the client MAC address or port or linkagg for which debug needs to be enabled. At a time, the debug will be done only for a MAC or a port or a linkagg and a combination of this will not be used. The port/MAC/linkagg specified at last is taken for logging. A user can specify whether the DHCP packet also needs to be dumped into the log buffer. This utility is for debugging switch with DHCP Snooping enabled. If DHCP Snooping is not enabled, then the DHCP packets may not be reaching CMM and may not get logged into the buffer.

The below API and wrapper is added to get the input from the user.

MIP\_RETCOD enable\_udp\_relay\_dhcp\_snooping\_debug\_log(MIP\_OIDC \*inIndex)

void WRPMiscEnableDhcpSnoopingDebugLog(CliSubparserGlobals \*globals, int displayId, int option1, int option2, int slot\_agg, int port, char\* macAddress, int vlan)

The debug configurations will not be updated in configuration snapshot. It will be stored in a run time data structure. These configurations will be displayed along with the logs by “debug show dhcp” command. These configurations will not be persistent across reload/takeover. A new data structure is added to store the configurations for DHCP Snooping on demand debugging.

typedef struct dhcpClientDebug

{

int port;

uint8 mac[6];

uint8 flag;

int debug\_enable;

int dump\_packet\_enable;

int linkAgg;

int vlan;

}DHCP\_CLIENT\_DEBUG;

port: Holds the interface ID of port on which debugging needs to be enabled.

linkAgg: Holds the linkagg id on which debugging needs to be enabled.

mac: MAC address of the client that needs to be debugged.

vlan: VLAN on which the specified client will be searched.

flag: Flag denotes whether debugging needs to be done on port/MAC/linkagg.

#define DHCP\_SNOOPING\_DEBUG\_MAC 1

#define DHCP\_SNOOPING\_DEBUG\_PORT 2

#define DHCP\_SNOOPING\_DEBUG\_LINKAGG 3

debug\_enable: Specifies whether on demand debugging is enabled globally.

dump\_packet\_enable: Specifies whether dump packet option is enabled.

A new buffer will be added to log the DHCP message. Once the buffer gets filled up, the older logs will get overwritten. This buffer will store maximum 600 lines of log each line can be of maximum 160 character in length.

#define DHCP\_LOG\_LINESIZE 160

#define DHCP\_LOG\_MAXLINES 600

dhcplog = (char \*)malloc(DHCP\_LOG\_LINESIZE \* DHCP\_LOG\_MAXLINES);

The below APIs are added to create the infra for logging.

void dhcpLogInit()

void dhcpLogClear()

void dhcpLogInternal(int time\_stamp, char \*format, ...)

void packet\_dump\_dhcp\_debug(int32 num\_bytes, int8 \*signed\_addr)

Below macro is defined to call the DHCP Snooping logging API.

#define dhcpLog(time\_stamp, format, args...) \

do { \

dhcpLogInternal(time\_stamp, format, ## args); \

} while(0)

Below API is added to check whether debugging is enabled on the given MAC/interface(slot/linkagg).

We use the client hardware address present in the DHCP header to check the clients.

int dhcpDebugEnabled(uint8 \*mac, int vlan)

In case of linkAgg and port we use the UDPRelay database to find the clients connected to that port/linkagg.

CLI show/clear APIs and wrapper functions are added to display and clear the log buffer.

void WRPMiscShowDhcpSnoopingDebugLog(CliSubparserGlobals \*globals, int displayId)

void WRPMiscClearDhcpSnoopingDebugLog(CliSubparserGlobals \*globals, int displayId)

MIP\_RETCOD view\_udp\_relay\_dhcp\_snooping\_debug\_log()

MIP\_RETCOD clear\_udp\_relay\_dhcp\_snooping\_debug\_log()

* + 1. **ISF Statistics.**

ISF statistics command will show the packets dropped per vlan/port. There is a counter attached to the drop rule in TCAM. These counters will be fetched to CMM from TCAM Manger.

* + - 1. **Clear command for DHCP snooping ISF statistics.**

Statistics will be cleared by invoking the below API.

MIP\_RETCOD clear\_udp\_relay\_dhcp\_snooping\_isf\_statistics(int option, int slot\_agg, int port, int vlan)

The below CLI wrapper function is added to invoke this API.

void WRPMiscClearDhcpIsfStatistics(CliSubparserGlobals \*globals, int displayId, int option, int slot\_agg, int port, int vlan)

* + - 1. **CLI Show command to display ISF statistics.**

A new API is added to display the DHCP snooping ISF statistics. The counters value from NI will be fetched by CMM every 60 seconds when the ISF is enabled

MIP\_RETCOD view\_udp\_relay\_dhcp\_snooping\_isf\_statistics()

The below CLI wrapper function is added to invoke this API.

* + - 1. void WRPMiscShowDhcpISFStatistics(CliSubparserGlobals \*globals, int displayId

# Appendix M: RTR-5323 DHCP Relay Over IP Interface For Service

## INTRODUCTION

### Purpose

The purpose of this appendix is to present complete details on RTR 5323, DHCP Relay over Interfaces for service domain (Phase-1) in 8.5.R3.

### Scope

The scope of this appendix is to present complete functional description of RTR 5323.

With RTR 5323, the DHCP Relay feature will be available to configure over IP interfaces for service domain (SPB) on platform 9900.

### Intended Audience

This document is intended for the following audience.

* Engineering Design Team.
* Product Testing Team.
* System Testing Team.

### Referenced Documents

#### Internal Documents

* Please Refer 3.2.10.1 section for DHCP Relay
* Services SFS
* Inline routing SFS

#### External Documents

<https://tools.ietf.org/html/rfc3046>

## FUNCTIONAL DESCRIPTION

### Basic Overview

The RTR 5323 allows user to configure DHCP relay over IP interface for service domain, meaning, user has option to specify IP interface name, from which DHCP packet will be relayed.

Note: The DHCP Snooping operations are not supported on services.

## FUNCTIONAL REQUIREMENTs.

Note that, the user specifies incoming IP interface name while configuring server address. Please note that this should be the primary IP interface.

## PLATFORM SUPPORTED

As of now this is supported only on 9900.

## MANAGEMENT INTERFACE

### Command Line Interface (CLI)

Following are the commands introduced for service domain DHCP Relay in 8.5. R03.

1. Command to enable Relay per interface mode.
2. Command to configure DHCP Relay for service interface
3. Command to show DHCP relay statistics
4. Command to clear DHCP relay statistics
5. Command to show DHCP relay configuration snap-shot

1. Command to enable Relay per interface mode. By default, mode will be global.

-> [no] ip dhcp relay per-interface-mode

2. Command to configure DHCP Relay for service interface

ip dhcp relay admin-state enable/disable

 Global admin state for enabling/disabling IP DHCP relay

Default State: Feature Disable

If this is disabled. No matter whatever relevant configuration you have, they will not take effect.

[no]ip dhcp relay per-interface-mode

 For enabling IP DHCP Relay per interface mode.

Default Mode: Global

[no]ip dhcp relay interface <name> destination <IPv4 address>

 For configuring destination address.

ip dhcp relay interface <name> admin-state enable/disable

 Disabling/enabling relay on an interface.

Default State: Enable (Once you configure the per interface address this command will get enabled automatically.)

[no]ip dhcp relay destination <IPv4 address>

 To be used when per-interface-mode is off

NOTE: If the IP interface is on VLAN, the above configuration will not be supported in 8.5. R3 and errored out with below message.

“*This command is not supported on VLAN IP interface. Please use ip helper command instead.*”

3. Command to show DHCP relay statistics

-> show ip dhcp relay statistics

Global Statistics :

Reception From Client :

Total Count = 2, Delta = 0

Forw Delay Violation :

Total Count = 0, Delta = 0

Max Hops Violation :

Total Count = 0, Delta = 0

Agent Info Violation :

Total Count = 0, Delta = 0

Invalid Gateway IP :

Total Count = 0, Delta = 0

Server Specific Statistics :

From any Vlan to Server 3.3.0.2

Tx Server :

Total Count = 2, Delta = 0

InvAgentInfoFromServer:

Total Count = 0, Delta = 0

4. Command to clear DHCP relay statistics

-> ip dhcp relay clear statistics

Global Statistics :

Reception From Client :

Total Count = 0, Delta = 0

Forw Delay Violation :

Total Count = 0, Delta = 0

Max Hops Violation :

Total Count = 0, Delta = 0

Agent Info Violation :

Total Count = 0, Delta = 0

Invalid Gateway IP :

Total Count = 0, Delta = 0

Server Specific Statistics :

From any IP interface to Server 3.3.0.2

Tx Server :

Total Count = 0, Delta = 0

InvAgentInfoFromServer:

Total Count = 0, Delta = 0

5. Command to show DHCP relay configuration snap-shot

-> show configuration snapshot ip-dhcp-relay

! DHCP Relay:

ip dhcp relay interface client-traffic address 3.3.0.2

## Scalability and Limitations

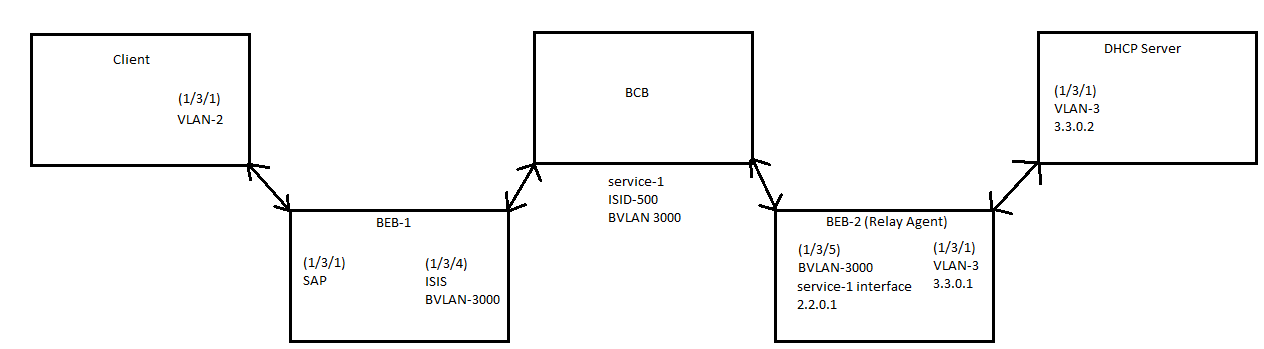
Maximum number of next hop IP address or next hop instances is 1536.

(This number including both services and VLAN).

## Testability.

Below is the typical-topology for DHCP relay over IP interfaces.

The (BEB-2) DHCP relay agent will be the DUT.



BEB-1

spb bvlan 3000 admin-state enable

spb isis bvlan 3000 ect-id 1

spb isis control-bvlan 3000

spb isis interface port 1/3/4

spb isis admin-state enable

service 1 spb isid 500 bvlan 3000

service access port 1/3/1

service 1 sap port 1/3/1:0 admin-state enable

BEB Agent

spb bvlan 3000 admin-state enable

spb isis bvlan 3000 ect-id 1

spb isis control-bvlan 3000

spb isis interface port 1/3/5

spb isis admin-state enable

service 1 spb isid 500 bvlan 3000

ip interface "client\_traffic" address 2.2.0.1 mask 255.255.255.0 service 1

DHCP Relay command on BEB Agent

-> ip dhcp relay interface-name client-traffic address 3.3.0.2

Logs

Below debug logs can be enabled on both client and agent.

-> swlog enable

-> swlog appid udprelay subapp all level debug3

-> swlog appid udprelay subapp all enable

-> swlog appid ipni subapp all level debug3

-> swlog appid ipni subapp all enable

## WebView Support.

A separate Web page should be provided for service domain for the same.

## SNMP Support.

A separate MIB table has been added for the commands mentioned in CLI section.

-- -----------------------------------------------------------------------------

-- DHCP RELAYTABLE CONFIGURATION

-- -----------------------------------------------------------------------------

alaDhcpRelayInterfaceTable OBJECT-TYPE

SYNTAX SEQUENCE OF AlaDhcpRelayInterfaceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An IP DHCP Relay interface table."

::= { alaDhcpRelayMIB 2 }

alaDhcpRelayInterfaceEntry OBJECT-TYPE

SYNTAX AlaDhcpRelayInterfaceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An IP DHCP interface entry."

INDEX { alaDhcpRelayInterfaceName , alaDhcpRelayInterfaceIpAddressType, alaDhcpRelayInterfaceIpAddress}

::= { alaDhcpRelayInterfaceTable 1 }

AlaDhcpRelayInterfaceEntry ::= SEQUENCE {

alaDhcpRelayInterfaceName SnmpAdminString,

alaDhcpRelayInterfaceIpAddressType InetAddressType,

alaDhcpRelayInterfaceIpAddress InetAddress,

alaDhcpRelayInterfaceStatus RowStatus

}

alaDhcpRelayInterfaceName OBJECT-TYPE

SYNTAX SnmpAdminString(SIZE (1..63))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"IP interface name for which DHCP Relay is applicable."

::= { alaDhcpRelayInterfaceEntry 1 }

::= { alaDhcpRelayInterfaceEntry 1 }

alaDhcpRelayInterfaceIpAddressType OBJECT-TYPE

SYNTAX InetAddressType { ipv4(1), ipv6(2) }

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Server IP address for DHCP Relay type."

::= { alaDhcpRelayInterfaceEntry 2 }

alaDhcpRelayInterfaceIpAddress OBJECT-TYPE

SYNTAX InetAddress (SIZE(4|16))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Server IP address for DHCP Relay."

::= { alaDhcpRelayInterfaceEntry 3 }

alaDhcpRelayInterfaceStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object specifies the status of the entry."

::= { alaDhcpRelayInterfaceEntry 4 }

--

-- Alcatel DHCP Relay Server Destination Table

--

alaDhcpRelayServerDestinationTable OBJECT-TYPE

SYNTAX SEQUENCE OF AlaDhcpRelayServerDestinationEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Table allowing the creation and removal of DHCP

relay server destinations."

::= { alaDhcpRelayMIB 3 }

alaDhcpRelayServerDestinationEntry OBJECT-TYPE

SYNTAX AlaDhcpRelayServerDestinationEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A DHCP relay server destination entry."

INDEX {

alaDhcpRelayServerDestinationAddressType,

alaDhcpRelayServerDestinationAddress

}

::= { alaDhcpRelayServerDestinationTable 1 }

AlaDhcpRelayServerDestinationEntry ::= SEQUENCE {

alaDhcpRelayServerDestinationAddressType InetAddressType,

alaDhcpRelayServerDestinationAddress InetAddress,

alaDhcpRelayServerDestinationRowStatus RowStatus

}

alaDhcpRelayServerDestinationAddressType OBJECT-TYPE

SYNTAX InetAddressType {ipv4(1)}

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The address type of a DHCP relay server destination."

::= { alaDhcpRelayServerDestinationEntry 1 }

alaDhcpRelayServerDestinationAddress OBJECT-TYPE

SYNTAX InetAddress (SIZE(4))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The address of a DHCP relay server destination."

::= { alaDhcpRelayServerDestinationEntry 2 }

alaDhcpRelayServerDestinationRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Used to control the addition and removal of DHCP

relay server destinations."

::= { alaDhcpRelayServerDestinationEntry 3 }

--

-- Alcatel DHCP Relay Interface Admin state table

--

alaDhcpRelayInterfaceAdminStateTable OBJECT-TYPE

SYNTAX SEQUENCE OF AlaDhcpRelayInterfaceAdminStateEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An IP DHCP relay interface status table."

::= { alaDhcpRelayMIB 4 }

alaDhcpRelayInterfaceAdminStateEntry OBJECT-TYPE

SYNTAX AlaDhcpRelayInterfaceAdminStateEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An IP DHCP interface status entry."

INDEX { alaDhcpRelayInterfaceName }

::= { alaDhcpRelayInterfaceAdminStateTable 1 }

AlaDhcpRelayInterfaceAdminStateEntry ::= SEQUENCE {

alaDhcpRelayInterfaceAdminStatus INTEGER

}

alaDhcpRelayInterfaceAdminStatus OBJECT-TYPE

SYNTAX INTEGER {

enable(1),

disable(2)

}

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"DHCP relay administrative status on the interface."

::= { alaDhcpRelayInterfaceAdminStateEntry 1 }

--

-- Alcatel DHCP Relay Clear Interface Statistics Table

--

alaDhcpRelayClearStatisticsTable OBJECT-TYPE

SYNTAX SEQUENCE OF AlaDhcpRelayClearStatisticsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An IP DHCP relay clear interface statistics table."

::= { alaDhcpRelayMIB 5 }

alaDhcpRelayClearStatisticsEntry OBJECT-TYPE

SYNTAX AlaDhcpRelayClearStatisticsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An IP DHCP clear interface statistics entry."

INDEX {

alaDhcpRelayClearStatisticsInterfaceName,

alaDhcpRelayClearStatisticsIpAddressType,

alaDhcpRelayClearStatisticsIpAddress

}

::= { alaDhcpRelayClearStatisticsTable 1 }

AlaDhcpRelayClearStatisticsEntry ::= SEQUENCE {

alaDhcpRelayClearStatisticsInterfaceName SnmpAdminString,

alaDhcpRelayClearStatisticsIpAddressType InetAddressType,

alaDhcpRelayClearStatisticsIpAddress InetAddress

}

alaDhcpRelayClearStatisticsInterfaceName OBJECT-TYPE

SYNTAX SnmpAdminString(SIZE (0..63))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"IP interface statistics name for which DHCP relay is applicable."

::= { alaDhcpRelayClearStatisticsEntry 1 }

alaDhcpRelayClearStatisticsIpAddressType OBJECT-TYPE

SYNTAX InetAddressType { ipv4(1) }

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Server IP address for DHCP relay type."

::= { alaDhcpRelayClearStatisticsEntry 2 }

alaDhcpRelayClearStatisticsIpAddress OBJECT-TYPE

SYNTAX InetAddress (SIZE(4))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Server IP address for DHCP relay."

::= { alaDhcpRelayClearStatisticsEntry 3 }

alaDhcpRelayGlobalConfig OBJECT IDENTIFIER ::= { alaDhcpRelayMIB 1 }

alaDhcpRelayAdminStatus OBJECT-TYPE

SYNTAX INTEGER {

enable(1),

disable(2)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Global DHCP administrative status. This object provides a

means to globally enable or disable."

::= { alaDhcpRelayGlobalConfig 2 }

alaDhcpRelayPerInterfaceMode OBJECT-TYPE

SYNTAX INTEGER {

enable(1),

disable(2)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object provides a means to globally enable or disable

the DHCP per interface mode."

::= { alaDhcpRelayGlobalConfig 8 }

## SYSTEM DESIGN (APIs and Data Structures Added/Modified).

The configuration will be on IP interface and this will be stored in the node of the rbtree “dhcpRelayAgentRequest” for DHCP Relay as follows. We will now listen to IP interface events and handle them for DHCP Relay. Under “dhcpRelayAgentRequest” function, based on incoming VLAN ID following packet-processing will be done.

When the packet comes in, it will be mapped to the node and it will be looked up to determine the server IP. In case of current helper behavior, the primary address is selected as gateway(giaddr) IP address. In the same way, in case, if there are multiple interfaces on a VLAN, the primary interface IP address will be selected as gateway IP address. So, user has to enable relay on primary interface.

Existing “*udpRelayNextHop*” data structure will be used for updating interface names.

*typedef struct udpRelayNextHop*

*{*

*RBT\_NODE node;*

*uint32 vlan\_number;*

*uint32 ipAddr; /\* no need to use struct in\_addr \*/*

*int txToServerCnt;*

*int invalidAgentInfoFrmSvrCnt;*

*int prevTxToServerCnt;*

*int prevInvalidAgentInfoFrmSvrCnt;*

*} DHCP\_RELAY\_NEXT\_HOP;*

On deletion of interface, which is configured for DHCP Relay over IP interface, the corresponding relay configuration will be removed.

Note: All the global helper configuration commands, mentioned below, also apply to DHCP Relay over Services also. This behavior is only for 8.5R03 release.

ip helper forward-delay

ip helper maximum-hops

ip helper agent-information

ip helper agent-information policy

ip helper option-82 format

APPENDIX N: RTR 5241 - DHCP Snooping with CVLAN option for Option 82 ASCII support

In AOS, if the Agent information or DHCP Snooping is enabled in the switch then while processing the DHCP Bootp request by the Relay Agent, option82 is inserted in the Bootp packet before sending it to the server. On receiving DHCP-Bootp reply packets, the option82 field in the packet will be stripped before sending the packet to the client. We will be adding support for CVLAN as well, in option 82 as a part of RTR 5241 in 85R4.

# System Requirement Specifications

## System Requirements

### Hardware Requirements

There is no hardware requirement for this feature.

### Software Requirements

**SFS\_UR\_101:** cvlan info needs to be added as available field for option-82 ASCII format.

### Platforms suppoprted

TOR (OS6900), Shasta (OS6860) and Everest (OS6865).

## Management Requirements

### MIB

For this feature, the following MIB needs to be changed.

AlcatelIND1UDPRelay.mib

For detailed MIB changes please see section [1.2.4](#_SNMP)

### CLI

There will be four new CLI additions for this feature.

The existing CLI with option “field” having the existing values as below will be enhanced to include cvlan:

base-mac, system-name, user-string <string>, interface, interface-alias, vlan.

Along with all these fields, new fields, as part of this RTR 5241, will be cvlan.

#### CLI to configure Remote ID

|  |  |
| --- | --- |
| [no] dhcp-snooping option-82 format ascii remote-id **[cvlan]** | |
| **Description:** | This CLI will be used to configure the type of information that is inserted in Remote ID suboption fields of the Option-82 field. for each field, one format has to be selected from the following options:  {base-mac, system-name, user-string string, interface-alias, interface, vlan, cvlan } |
| **Syntax Definitions:** | |
| Base-mac: | The base MAC address of the switch |
| system-name: | The system name of the switch. |
| String: | A user-defined text string up to 64 characters. |
| Interface-alias: | The configured interface-alias for the port on which the DHCP request is received (already in string format) |
| Interface: | The configured interface-alias for the port on which the DHCP request is received (already in string format) |
| Vlan: | The vlan number as a string format (i.e. 4 characters) |
| **Cvlan** | The cvlan number as a string format (i.e. 4 characters) |
| **Platforms Supported:** | OS6860,OS6865 & OS6900 |
| **Usage Guidelines:** | When two or more fields are configured, the delimiter is required. The data specified with this command is added to the Remote ID fields only when DHCP Option-82 data insertion is enabled for the switch. |
| **Examples:** | dhcp-snooping option-82 format ascii remote-id cvlan  dhcp-snooping option-82 format ascii remote-id base-mac vlan cvlan delimiter / |

#### CLI to remove Circuit ID

|  |  |
| --- | --- |
| [no] dhcp-snooping option-82 format ascii circuit-id **cvlan** | |
| **Description:** | This CLI will be used to unconfigure circuit-id fields set. |
| **Syntax Definitions:** | |
|  |  |
| **Platforms Supported:** | OS6860, OS6865 & OS6900. |
| **Cvlan** | The cvlan number as a string format (i.e. 4 characters) |
| **Examples:** | no dhcp-snooping option-82 format ascii circuit-id cvlan |

#### Modification of “show dhcp-snooping”

|  |
| --- |
| **Modified CLI Output:** |
| -> show dhcp-snooping    DHCP Snooping Status               =  Switch-Level Enabled,        Option 82 Data Insertion Per Switch  =  Enabled,        MAC Address Verification Per Switch  =  Enabled,    DHCP Snooping Bypass Opt82-Check   =  Disabled,    DHCP Snooping Opt82 ASCII Circuit ID Field1  =  Base MAC,    DHCP Snooping Opt82 ASCII Circuit ID Field1 String  =  00:e0:b1:91:45:d0,    DHCP Snooping Opt82 ASCII Circuit ID Field2  =  Cvlan,    DHCP Snooping Opt82 ASCII Circuit ID Field2 String  =   - ,    DHCP Snooping Opt82 ASCII Circuit ID Field3  =  Interface,    DHCP Snooping Opt82 ASCII Circuit ID Field3 String  =   - ,    DHCP Snooping Opt82 ASCII Circuit ID Field4  =  Interface Alias,    DHCP Snooping Opt82 ASCII Circuit ID Field4 String  =   - ,    DHCP Snooping Opt82 ASCII Circuit ID Field5  =  System Name,    DHCP Snooping Opt82 ASCII Circuit ID Field5 String  =  vxTarget,    DHCP Snooping Opt82 ASCII Circuit ID Delimiter  =  "/",    DHCP Snooping Opt82 ASCII Remote ID Field1  =  Vlan,    DHCP Snooping Opt82 ASCII Remote ID Field1 String  =   - ,    DHCP Snooping Opt82 ASCII Remote ID Field2  =  Cvlan,    DHCP Snooping Opt82 ASCII Remote ID Field2 String  =   - ,    DHCP Snooping Opt82 ASCII Remote ID Field3  =  User String,    DHCP Snooping Opt82 ASCII Remote ID Field3 String  =  biswajit,    DHCP Snooping Opt82 ASCII Remote ID Delimiter  =  " ",    DHCP Snooping Binding DB Status    =  Enabled,        Database Sync Timeout          =  300,        Database Last Sync Time        =  ,        Binding Persistency Status   =  Disabled,    PXE support                        =  Disabled,    Forward option     = standard        Vlan Number NA    Bootup Option Disable        Forwarding Address :          20.0.0.151    UDP Relay on Default VRF = Enabled  -> |

#### Modification of “show configuration snapshot dhcp-snooping”

|  |  |
| --- | --- |
| show configuration snapshot dhcp-snooping | |
| **Description:** | This CLI displays all the dhcp-snooping related configurations done in the switch. The lines in blue color are the new additions to the existing snapshot CLI by this modification. |
| **CLI Output:** | |
| -> show configuration snapshot dhcp-snooping  dhcp-snooping enable  dhcp-snooping binding enable  dhcp-snooping option-82 format base-mac  dhcp-snooping port 1/4 trust  dhcp-snooping option-82 format ascii circuit-id vlan base-mac delimiter “/”  dhcp-snooping option-82 format ascii remote-id cvlan  -> | |

### SNMP

The MIB changes will be in AlcatelIND1UDPRelay.mib

A new enum cvlan with value 7 will be added in the TEXTUAL-CONVENTION

IphelpereOption82ASCIIFieldType.

IphelpereOption82ASCIIFieldType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Ip helper Option 82 Format ASCII field type"

SYNTAX INTEGER {

none(0),

macAddress(1),

systemName(2),

userString(3),

interfaceAlias(4),

vlan(5),

interface(6),

cvlan(7)

}

A new SNMP table will be added to have the Circuit ID and Remote ID specific configuration details.

### WebView

The new option will be enable for the CVLAN in the DHCP Snooping Ciruit/Remote ID Configuration.

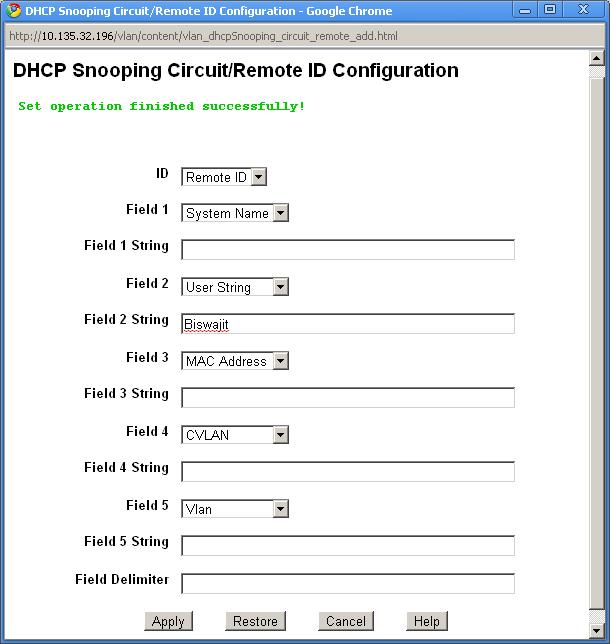


Fig: Configuration Page for Setting Circuit ID and Remote ID Explicitly

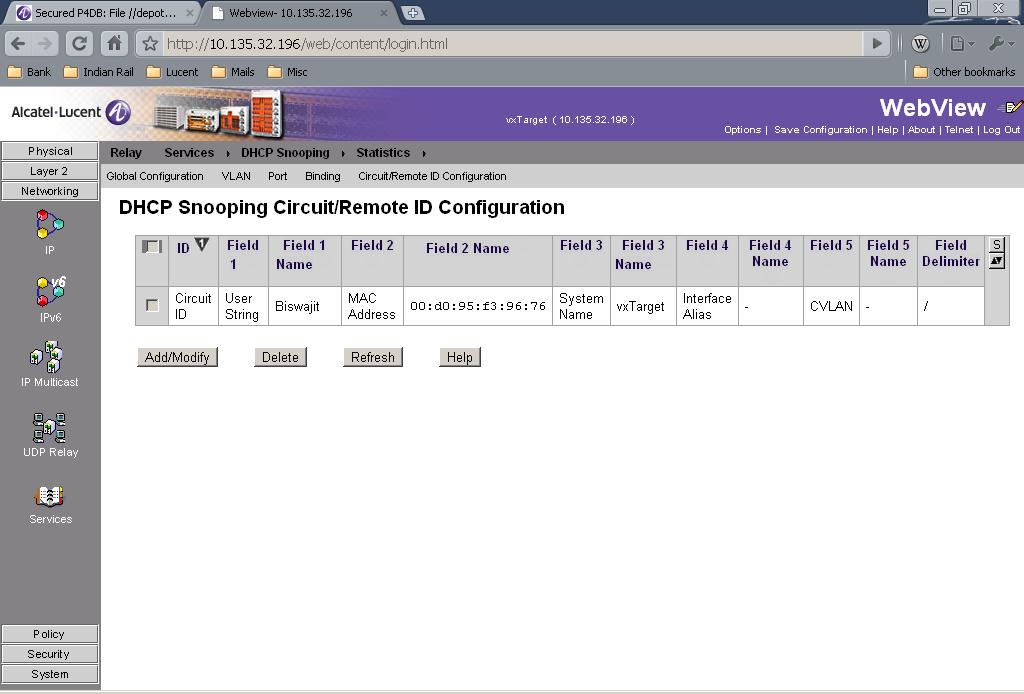


Fig: Explicit Circuit ID and Remote ID Display Page

## Impacted AOS Modules

**UDP Relay**

Changes in UDP Relay Module CMM will be done while processing DHCP packets.

## Use Cases

OmniSwitch (OS6860/OS6560) are deployed as CPE

Network core will be connected to services (IPTV, healthcare, educational …) from many different providers and use Open Access to users.

Case 1: One house 3 different carrier customers.

Carrier 1 has customer in ports 1-4

Carrier 2 has customer in ports 5-12

Carrier 3 has customer in ports 12-20

In this case there is only one carrier fiber coming to house and stacking VLAN is used.

Each carrier is coming to switch using their own **VLAN**. Delimiter in all cases is for example /:

|  |  |  |
| --- | --- | --- |
| Carrier | CID | RID |
| Carrier 1 | System name / port | "client-hostname" or "client-mac" |
| Carrier 2 | Port alias | System name / port |
| Carrier 3 | Port alias (ASCII) | CVLAN |

Case 2: one house 3 different carrier customers.

Carrier 1 has customer in ports 1-4

Carrier 2 has customer in ports 5-12

Carrier 3 has customer in ports 12-20

In this case there are 3 different carrier fibers coming to house (in middle of the town).

Each carrier is coming to switch using their own **PORT**. Delimiter in all cases is for example /:

|  |  |  |
| --- | --- | --- |
| Carrier | CID | RID |
| Carrier 1 | System name / port | "client-hostname" or "client-mac" |
| Carrier 2 | Port alias | System name / port |
| Carrier 3 | Port alias (ASCII) | CVLAN |

# FUNCTIONAL SPECIFICATIONS

## Implementation Details

1. PACKET CAPTURES

DHCP Client Hostname is set as My\_Laptop.

Case1:

====

Explicit ASCII Circuit ID - Set

Explicit ASCII Remote ID - Set

Global Configuration - Non-ASCII

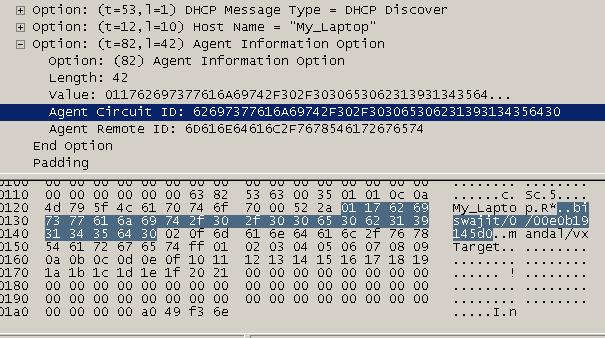
CLI Configuration Done:

==================

dhcp-snooping option-82 format ascii circuit-id user-string biswajit cvlan base-mac delimiter /

dhcp-snooping option-82 format ascii remote-id user-string mandal system-name delimiter /

dhcp-snooping option-82 format base-mac



In the example above, the offset value of Circuit-ID will be at 300 and Remote-ID will be at offset 325.

Case 2:

=====

Explicit ASCII Circuit ID - Set

Explicit ASCII Remote ID - Not Set

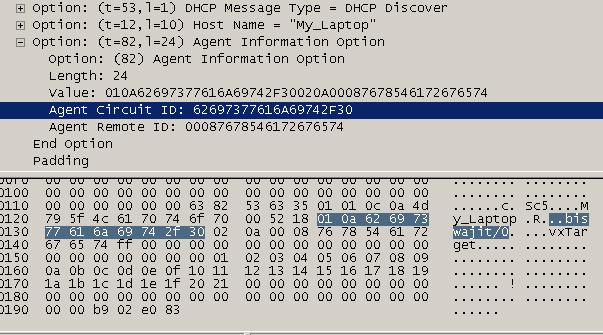
Global Configuration - Non-ASCII

CLI Configuration Done:

==================

dhcp-snooping option-82 format ascii circuit-id user-string biswajit cvlan delimiter /

dhcp-snooping option-82 format system-name



In the example above, the offset value of Circuit-ID will be at 299 and Remote-ID will be at offset 311.

Case 3:

=====

Explicit ASCII Circuit ID - Not Set

Explicit ASCII Remote ID - Set

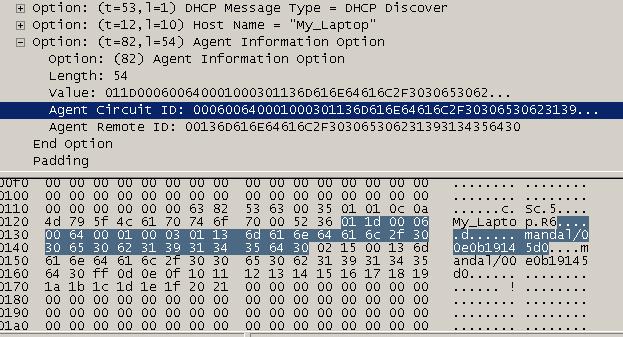
Global Configuration - Non-ASCII

CLI Configuration Done:

==================

-> dhcp-snooping option-82 format ascii remote-id user-string mandal base-mac delimiter /

-> dhcp-snooping option-82 format interface-alias



In the example above, the offset value of Circuit-ID will be at 300 and Remote-ID will be at offset 331.

Case 4:

=====

Explicit ASCII Circuit ID - Not Set

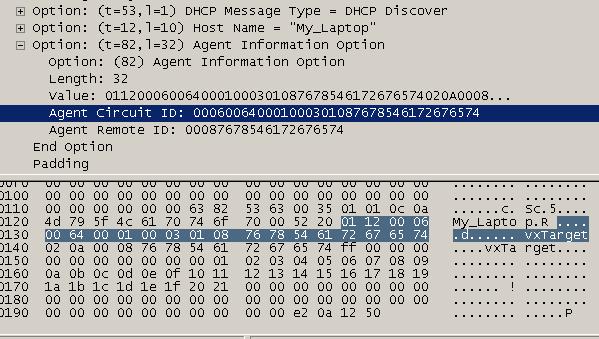
Explicit ASCII Remote ID - Not Set

Global Configuration - Non-ASCII

CLI Configuration Done:

==================

-> dhcp-snooping option-82 format system-name



In the example above, the offset value of Circuit-ID will be at 300 and Remote-ID will be at offset 320.

1. MANAGEMENT INTERFACE CROSS-REFERENCE
2. LIMITATION

The explicit remote-id or circuit-id can not be configured on per port, per vlan or per port/vlan

basis. This feature is only globally configurable.

APPENDIX O: RTR 5181 ISF scalability enhancement on 6560, RTR 5247 DHCP snooping binding table for ISF for 6465

The below features will be available on OS6560/ OS6465 as part of 85R04.

* Maximum number of vlans enabled on 6560 with ip-source-filtering are 32 as earlier.
* Maximum number of vlans enabled on 6465 with ip-source-filtering are 16.

The Maximum number of binding entries for ISF per NI are as per the table mentioned below.

|  |  |  |
| --- | --- | --- |
| Platforms | Vlan Level ISF | Port Level ISF |
| 6560 with 24 port | 32 Vlans with 223 clients | 254 clients |
| 6560 with 48 Port | 32 Vlans with 223 clients | 254 clients |
| 6465 | 16 Vlans with 31 Clients | 46 clients |

* Please note that with 6465 there will be one rule consumed per linkagg member port, if the binding entry client port is a linkagg with multiple member ports on the same NI.
* 6560 VLAN based ISF scalability matrix (Maximum 32 vlans can be enabled with ISF)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | VC of 1 | VC of 2 | VC of 3 | VC of 4 |
| 4 vlans | 251 | (251\*2) 502 | (251\*3) 753 | (251\*4)1004 |
| 8 vlans | 247 | (247\*2)          494 | (247\*3)741 | (247\*4)             988 |
| 16 vlans | 239 | (239\*2)             478 | (239\*3)             717 | (239\*4)956 |
| 32 vlans | 223 | (223\*2)           446 | (223\*3)669 | (223\*4)892 |

* 6465 VLAN based ISF scalability matrix (Maximum 16 vlans can be enabled with ISF)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | VC of 1 | VC of 2 | VC of 3 | VC of 4 |
| 4 vlans | 43 | (43\*2)86 | (43\*3)126 | (43\*4)172 |
| 8 vlans | 39 | (39\*2)78 | (39\*3)117 | (39\*4)156 |
| 16 vlans | 31 | (31\*2)62 | (31\*3)93 | (31\*4)124 |

APPENDIX P: RTR 5400 Apple netboot support with DHCP Snooping, Relay

**Description:**

A NetBoot 2.0 client uses the Boot Server Discovery Protocol (BSDP) to dynamically acquire resources that enable it to boot a suitable operating system. The client uses DHCP to acquire its IP address and BSDP to acquire boot image resources. The protocols are initiated by the client at boot time.

BSDP uses a two-packet exchange mechanism modeled after DHCP. The first packet sent by the client generates a response from a BSDP server. The response contains a list of possible boot images that the client may load. The second packet sent by the client selects a particular server and boot image. The successful response from the selected server confirms the selection.

BSDP uses DHCP INFORM and ACK packets for the communication between the client and server. The Vendor Class Identifier option is set to a value that identifies it as a BSDP packet. The Vendor Specific Information option includes a BSDP Message Type option that specifies one of LIST, SELECT, or FAILED.



Figure : Initial Boot Sequence

The role of AOS DHCP Snooping or Relay is to support BSDP in the network, when we are an intermediate switch. For a single INFORM, there will be multiple ACKs sent from the various BSDP servers, and the DUT has to forward them downstream. Please note that there will be no additional configuration required for Netboot support in DHCP Snooping/ Relay. Please note that the support will POC in 85R3 and it will be officially supported across platforms in 86R1.

APPENDIX Q: ISF Scalability NUMBERS for 6860, 6865, 6900

Following are the scalability numbers for ISF for the above platforms from 85R4 :-

6860/6865/6900

* Port Level ISF supports 253 Clients Per NI.
* VLAN based ISF scalability matrix (Maximum 32 vlans can be enabled with ISF) is below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | VC of 1 | VC of 2 | VC of 3 | VC of 4 |
| 4 vlans | 244 | (244\*2) 488 | (244\*3) 732 | (244\*4) 976 |
| 8 vlans | 232 | (232\*2)  464 | (232\*3) 696 | (232\*4)  928 |
| 16 Vans | 208 | (208\*2)   416 | (208\*3)           624 | (208\*4) 832 |
| 32 vlans | 160 | (160\*2)   320 | (160\*3) 480 | (160\*4) 640 |

9900

* Port Level ISF supports 254 Clients Per NI.
* VLAN based ISF scalability matrix (Maximum 32 vlans can be enabled with ISF) is below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Single NI | 2 Nis | 3 NIs | 4 Nis |
| 4 vlans | 251 | (251\*2) 502 | (251\*3)753 | (251\*4) 1004 |
| 8 vlans | 247 | (247\*2) 494 | (247\*3)741 | (247\*4)             988 |
| 16 vlans | 239 | (239\*2) 478 | (239\*3)717 | (239\*4) 956 |
| 32 vlans | 223 | (223\*2) 446 | (223\*3)669 | (223\*4) 892 |

APPENDIX R: RTR 5372 – GeNERIC UDP Relay Over services

This section briefly describes the feature to be implemented as part POC in 85R4 (RTR-5434) and officially supported via RTR 5372 in 86R1. The supported platforms are OS9900 (Medora).

The following will be supported in 85R04 on OS9900 UDP Relay over:   
1. SPB using inline routing (OS9900)  
2. SPB using loopback cable / traditional routing (OS6900-V72, OS6900-C32, OS9900)

The existing AOS implementation redirects the broadcasted UDP packets to a destination VLAN(s) or a specific address.

This enhancement provides an additional feature of relaying the UDP packets to the configured service ID.

We will also support UDP relay for packets coming on IP interfaces over SPB.

# System Requirement Specifications

## System Requirements

### Hardware Requirements

This enhancement is restricted to AOS 9900 which supports inline routing for services.

### Software Requirements

TBD

### Platforms supported

OS9900

## Management Requirements

### MIB

alaGenericUdpServiceSvcGroup OBJECT-GROUP

OBJECTS

{

alaGenericUdpServiceStatTxToSvc,

alaGenericUdpServiceSvcRowStatus

}

STATUS current

DESCRIPTION

" Objects for IP DHCP service SVC."

### CLI

Below command is the new CLI to support generic UDP relay over services for egress traffic.

-> ip udp relay service {tftp | tacacs | ntp | nbns | nbdd | dns} | port <num> [description <string>] [no] {vlan <vid>[-<vid2>]} | svc <svcId>[-<svcId2>]}

### SNMP

To be supported.

### WebView

To be supported.

## Impacted AOS Modules

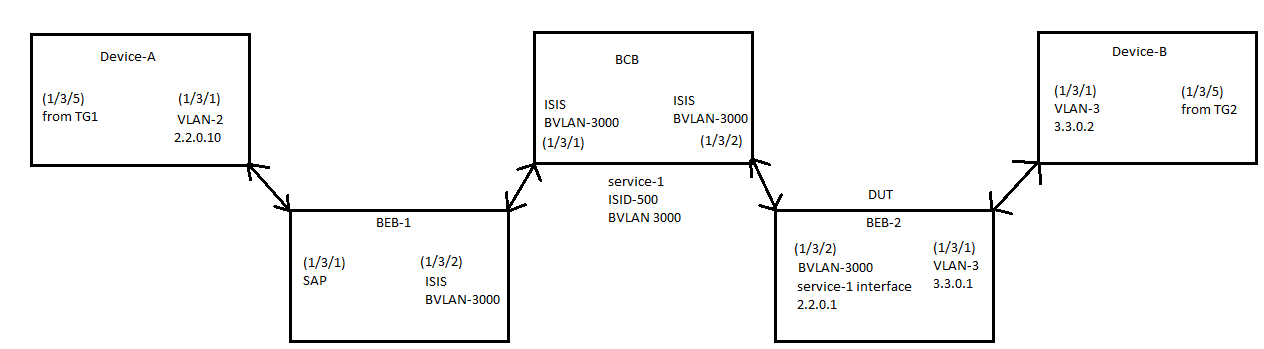
Changes in UDP Relay Module CMM will be done with generic relay perspective.

(Ingress Or Egress traffic being on IP interface over SPB)

# FUNCTIONAL SPECIFICATIONS

The function *genericHandleRequest()* has been updated to check the eVLAN and then allow further processing in relay.

## Topology



Example:

* TFTP service is assumed.
* Service instance with ID 1 is assumed.

Possible commands based on ingress and egress traffic.

|  |  |  |
| --- | --- | --- |
| Ingress | Egress | Command |
| Over Non-Service | Over Non-Service | ->ip udp relay service tftp vlan 3  ->ip udp relay service tftp address 3.3.0.2 |
| Over Non-Service | Over Service | ->ip udp relay service tftp svc 1  ->ip udp relay service ttp address 2.2.0.10 |
| Over Service | Over Non-Service | ->ip udp relay service tftp vlan 40  ->ip udp relay service tftp address 3.3.0.2 |
| Over Service | Over Service | ->ip udp relay service tftp svc 1 |

APPENDIX S: RTR-5335 New DHCP Snooping Binding Table CLI OPTIONS and Behavior

**Description:**

For specific reasons of debug, there is a need to display the entries in the increasing order of the associated port. Also there is a requirement to add keywords to filter the binding entries based on the port and IP address. These changes are implemented in the part of RTR-5335 from 86R01 onwards, across platforms.

**Requirements:**

The Existing display will be changed to display in the increasing order of the port.

New command is required to display the binding table with option to filter port.

New command is required to display the binding table with option to filter ip-address.

**Command Line Interface (CLI):**

**show dhcp snooping binding [port <**chassis/slot/port> | **linkagg** <id> **| ip-address <**ipaddr>**]**

Example:

-> **show dhcp snooping binding**

**Total Number of Binding Entries: 10**

**MAC IP Lease VLAN Binding**

**Address Port Address Time ID Type**

**-----------------+-----------+-----------------+---------------+--------+-----------**

**02:00:00:00:0a:00 1/1/5 100.100.100.11 30 100 Dynamic**

**02:00:00:00:06:00 1/1/5 100.100.100.20 30 100 Static**

**02:00:00:00:02:00 1/1/6 100.100.100.10 30 100 Dynamic**

**02:00:00:00:09:00 1/1/6 100.100.100.18 30 100 Dynamic**

**02:00:00:00:08:00 1/1/6 100.100.100.6 30 100 Dynamic**

**02:00:00:00:05:00 1/1/6 100.100.100.8 30 100 Dynamic**

**02:00:00:00:03:00 1/1/7 100.100.100.3 30 100 Dynamic**

**02:00:00:00:01:00 1/1/7 100.100.100.17 30 100 Dynamic**

**02:00:00:00:07:00 0/1 100.100.100.13 30 100 Dynamic**

**02:00:00:00:04:00 0/1 100.100.100.15 30 100 Dynamic**

-> **show dhcp snooping binding port 1/1/5**

**Total Number of Binding Entries: 2**

**MAC IP Lease VLAN Binding**

**Address Port Address Time ID Type**

**-----------------+-----------+-----------------+---------------+--------+-----------**

**02:00:00:00:0a:00 1/1/5 100.100.100.11 30 100 Dynamic**

**02:00:00:00:06:00 1/1/5 100.100.100.20 30 100 Static**

-> **show dhcp snooping binding linkagg 1**

**Total Number of Binding Entries: 2**

**MAC IP Lease VLAN Binding**

**Address Port Address Time ID Type**

**-----------------+-----------+-----------------+---------------+--------+-----------**

**02:00:00:00:07:00 0/1 100.100.100.13 30 100 Dynamic**

**02:00:00:00:04:00 0/1 100.100.100.15 30 100 Dynamic**

-> **show dhcp snooping binding ip-address 100.100.100.11**

**Total Number of Binding Entries: 1**

**MAC IP Lease VLAN Binding**

**Address Port Address Time ID Type**

**-----------------+-----------+-----------------+---------------+--------+-----------**

**02:00:00:00:0a:00 1/1/5 100.100.100.11 30 100 Dynamic**

APPENDIX T: RTR-5371 DHCP-Snooping Global Admin Disable

DHCP snooping and ISF will be decoupled and removal of configuration for Snooping/ disabling Snooping will not affect ISF functionality/ ISF config. ISF would work independently as per user configuration with static binding entries configured.

**Overview**

1. A 'no' command for DHCP snooping shall be added to remove the snooping configuration and flush the dynamic binding entries.
2. dhcp-snooping admin-state disable will disable DHCP-snooping but retain user configuration in that scope. The dynamic binding entries would be flushed on snooping being disabled.
3. A new 'dhcp-snooping ip-source-filter enable/ disable' will be introduced. Globally ISF disable will disable the ISF functionality - remove drop rules/ binding entries in hardware without removing the user configuration. They will be reprogrammed on global ISF enable.
4. Other Snooping/ ISF behavior remains as is.

**Management Interface**

**Command Line Interface:**

**Changes will be as follows:-**

**[no]** dhcp-snooping [vlan <vlan-id>] admin-state {enable| disable}

**dhcp-snooping ip-source-filter admin-state {enable| disable}**

APPENDIX U: RTR 5372 – IPv4 relay commands migration

This section explains on renaming and migration of existing IPv4 commands for:

1. UDP Relay
2. DHCP Relay

Note: “*ip helper …*” commands are migrated to to “*ip dhcp relay ..*”

This is targeted for 86R01 release.

Note: For detailed information on enhanced commands related to IP UDP Relay Over Service, please refer appendix-M RTR-5732. Whenever user executes any deprecated commands mentioned above, proper deprecation message will be given along with new command notification.

# System Requirement Specifications

## System Requirements

All platforms are supported.

## Management Requirements

### MIB

Please refer SNMP section.

### CLI

This section explains on new format of commands. Please note that the following are applicable in 86R1 regardless of whether the ingress IP interface is on a VLAN or a service.

|  |  |  |
| --- | --- | --- |
| **New command format** | **Description** | **Old / Deprecated command** |
| ip dhcp relay admin-state {enable | disable} | Global admin state for IPv4 DHCP Relay. | This is totally new command introduced in this RTR. |
| [no] ip dhcp relay destination <server\_ipv4\_address> | Add or delete global DHCP server IPv4 address to be used for DHCP relay. | ip helper address |
| [no] ip dhcp relay interface <ifName> [destination <server\_ipv4\_address>] [admin-state {enable | disable}] | DHCP relay state on IPv4 interface. Indicates the incoming IP interface to relay received DHCP packets. Destination (i.e. DHCP server) address is applicable only if operating in per-interface mode (i.e. global admin state is disabled). | ip helper vlan address |
| ip dhcp relay forward-delay <seconds> | Configure forward delay time value for IPv4 DHCP relay. Global setting | ip helper forward-delay |
| [no] ip dhcp relay per-interface-mode | Toggles between per-interface-mode and global mode | ip helper standard  ip helper per-vlan-only |
| **New command format** | **Description** | **Old / Deprecated command** |
| [no] ip dhcp insert-agent-information [policy {drop|keep|repalce}] | Enable/Disable insertion and handling of DHCP relay agent information (Option 82).  This can be configured only if DHCP snooping option-82 is disabled. | ip helper agent-information policy  Note: Below command has been completely removed.  ip helper agent-information {enable|disable} |
| [no] ip dhcp relay pxe-support | Enable relay agent support for Preboot Execution Environment (PXE) devices | ip helper pxe-support |
| [no] ip dhcp relay insert-agent-information format [base-mac | system-name | user-string string | interface-alias | autointerface-  alias | ascii [{ remote-id | circuit-id} {base-mac | cvlan | interface | interface-alias | systemname  | user-string string | vlan} {delimiter string}]] | Configures the type of information that is inserted into both the Circuit ID and Remote ID sub option  fields of the Option-82 field. Please note that the highlighted options are introduced as a part of this RTR. Also, the option82 configuration is common and therefore mutually exclusive between DHCP Snooping & Relay as earlier.  This can be configured only if DHCP snooping option-82 is disabled. | ip helper option-82 format |
| ip dhcp relay clear statistics [global-only] [destination <ipv4\_address>] [interface <ifName> destination <ipv4\_address>] | Clear statistics counters (Global or per-interface) | no ip helper statistics |
| ip dhcp relay clear insert-agent-information error-count [port chassis/slot/port | interface <name> | port chassis/slot/port interface <name>] | Clears the Option 82 related error statistics on a per port and per vlan basis. | no ip helper option-82 error-count [port chassis/slot/port | vlan vlan id | port chassis/slot/port vlan  vlan id] |
| show ip dhcp relay insert-agent-information error-count [port chassis/slot/port | interface <name> | port chassis/slot/port interface <name>] | Displays the Option 82 related error statistics on a per port and per interface basis. | show ip helper option-82 error-count [port chassis/slot/port | vlan vlan id | port chassis/slot/port vlan  vlan id] |
| show ip dhcp relay [interface] | Display IPv4 DHCP relay (and agent) information | show ip helper |
| show ip dhcp relay statistics | Display IPv4 DHCP relay statistics | show ip helper statistics |

Below is the enhancement in IP UDP Relay command.

|  |  |
| --- | --- |
| **IPv4 UDP Relay Commands (svc is new parameter for service)** | **Description** |
| ip udp relay service {tftp | tacacs | ntp | nbns | nbdd | dns} | port <*num*> [description <*string*>] [no] {vlan <*vid*>[-<*vid2*>]} | svc <*svcId*>[-<*svcId2*>]} | Specify a VLAN or Service (svc) domain on which IPv4 traffic to a UDP port is forwarded |
| ip udp relay clear statistics  Note: “ip udp relay no statistics” is deprecated. | Reset all the generic IPv4 UDP relay service related statistics |
| show ip udp relay [{service {tftp | tacacs | ntp | nbns | nbdd | dns}} | {port <num>}] | Display the VLAN/Service domain assignments on which IPv4 traffic received on the UDP service ports is forwarded |
| show ip udp relay statistics [{service {tftp | tacacs | ntp | nbns | nbdd | dns}} | {port <num>}] | Display the current statistics for each IPv4 UDP port relay service |

### SNMP

IPv4 DHCP MIBS

alaDhcpRelayInterfaceGroup OBJECT-GROUP

OBJECTS

{

alaDhcpRelayInterfaceStatus

}

STATUS current

DESCRIPTION

" Objects for IP DHCP interface table."

::= { alcatelIND1UDPRelayMIBGroups 13}

alaDhcpRelayGlobalConfigGroup OBJECT-GROUP

OBJECTS

{

alaDhcpRelayStatisticsClear,

alaDhcpRelayAdminStatus,

alaDhcpRelayForwardDelay,

alaDhcpRelayMaximumHops,

alaDhcpRelayPxeSupport,

alaDhcpRelayInsertAgentInformation,

alaDhcpRelayInsertAgentInformationPolicy,

alaDhcpRelayPerInterfaceMode

}

STATUS current

DESCRIPTION

" Objects for IP DHCP statistics."

::= { alcatelIND1UDPRelayMIBGroups 14}

alaDhcpRelayServerDestinationGroup OBJECT-GROUP

OBJECTS

{

alaDhcpRelayServerDestinationRowStatus

}

STATUS current

DESCRIPTION

" Objects for IP DHCP relay server destination."

::= { alcatelIND1UDPRelayMIBGroups 15}

alaDhcpRelayInterfaceAdminStateGroup OBJECT-GROUP

OBJECTS

{

alaDhcpRelayInterfaceAdminStatus

}

STATUS current

DESCRIPTION

" Objects for IP DHCP relay interface admin status ."

::= { alcatelIND1UDPRelayMIBGroups 16}

-- -----------------------------------------------------------------------------

-- DHCP RELAY GLOBAL CONFIGURATION

-- -----------------------------------------------------------------------------

alaDhcpRelayGlobalConfig OBJECT IDENTIFIER ::= { alaDhcpRelayMIB 1 }

alaDhcpRelayStatisticsClear OBJECT-TYPE

SYNTAX INTEGER {

default(1),

reset(2)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object defines DHCP Relay statistics information clear or reset."

::= { alaDhcpRelayGlobalConfig 1 }

alaDhcpRelayAdminStatus OBJECT-TYPE

SYNTAX INTEGER {

enable(1),

disable(2)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Global DHCP administrative status. This object provides a

means to globally enable or disable."

::= { alaDhcpRelayGlobalConfig 2 }

alaDhcpRelayForwardDelay OBJECT-TYPE

SYNTAX Unsigned32 (0 .. 65535)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"DHCP relay forward delay value in seconds."

::= { alaDhcpRelayGlobalConfig 3 }

alaDhcpRelayMaximumHops OBJECT-TYPE

SYNTAX Unsigned32 (1..16)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"DHCP relay maximum hops."

::= { alaDhcpRelayGlobalConfig 4 }

alaDhcpRelayPxeSupport OBJECT-TYPE

SYNTAX INTEGER {

enable(1),

disable(2)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object provides a means to globally enable or disable

the DHCP pxe-support."

::= { alaDhcpRelayGlobalConfig 5 }

alaDhcpRelayInsertAgentInformation OBJECT-TYPE

SYNTAX INTEGER {

enable(1),

disable(2)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Global DHCP insert-agent-information state. This object provides a

means to globally enable or disable the DHCP insert-agent-information."

::= { alaDhcpRelayGlobalConfig 6 }

alaDhcpRelayInsertAgentInformationPolicy OBJECT-TYPE

SYNTAX INTEGER {

drop(1),

keep(2),

replace(3)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The policy of DHCP relay insert agent information."

::= { alaDhcpRelayGlobalConfig 7 }

alaDhcpRelayPerInterfaceMode OBJECT-TYPE

SYNTAX INTEGER {

enable(1),

disable(2)

}

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object provides a means to globally enable or disable

the DHCP per interface mode."

::= { alaDhcpRelayGlobalConfig 8 }

-- -----------------------------------------------------------------------------

-- DHCP RELAYTABLE CONFIGURATION

-- -----------------------------------------------------------------------------

alaDhcpRelayInterfaceTable OBJECT-TYPE

SYNTAX SEQUENCE OF AlaDhcpRelayInterfaceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An IP DHCP Relay interface table."

::= { alaDhcpRelayMIB 2 }

alaDhcpRelayInterfaceEntry OBJECT-TYPE

SYNTAX AlaDhcpRelayInterfaceEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An IP DHCP interface entry."

INDEX { alaDhcpRelayInterfaceName , alaDhcpRelayInterfaceIpAddressType, alaDhcpRelayInterfaceIpAddress}

::= { alaDhcpRelayInterfaceTable 1 }

AlaDhcpRelayInterfaceEntry ::= SEQUENCE {

alaDhcpRelayInterfaceName SnmpAdminString,

alaDhcpRelayInterfaceIpAddressType InetAddressType,

alaDhcpRelayInterfaceIpAddress InetAddress,

alaDhcpRelayInterfaceStatus RowStatus

}

alaDhcpRelayInterfaceName OBJECT-TYPE

SYNTAX SnmpAdminString(SIZE (1..63))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"IP interface name for which DHCP Relay is applicable."

::= { alaDhcpRelayInterfaceEntry 1 }

alaDhcpRelayInterfaceIpAddressType OBJECT-TYPE

SYNTAX InetAddressType { ipv4(1), ipv6(2) }

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Server IP address for DHCP Relay type."

::= { alaDhcpRelayInterfaceEntry 2 }

alaDhcpRelayInterfaceIpAddress OBJECT-TYPE

SYNTAX InetAddress (SIZE(4|16))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Server IP address for DHCP Relay."

::= { alaDhcpRelayInterfaceEntry 3 }

alaDhcpRelayInterfaceStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object specifies the status of the entry."

::= { alaDhcpRelayInterfaceEntry 4 }

--

-- Alcatel DHCP Relay Server Destination Table

--

alaDhcpRelayServerDestinationTable OBJECT-TYPE

SYNTAX SEQUENCE OF AlaDhcpRelayServerDestinationEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Table allowing the creation and removal of DHCP

relay server destinations."

::= { alaDhcpRelayMIB 3 }

alaDhcpRelayServerDestinationEntry OBJECT-TYPE

SYNTAX AlaDhcpRelayServerDestinationEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A DHCP relay server destination entry."

INDEX {

alaDhcpRelayServerDestinationAddressType,

alaDhcpRelayServerDestinationAddress

}

::= { alaDhcpRelayServerDestinationTable 1 }

AlaDhcpRelayServerDestinationEntry ::= SEQUENCE {

alaDhcpRelayServerDestinationAddressType InetAddressType,

alaDhcpRelayServerDestinationAddress InetAddress,

alaDhcpRelayServerDestinationRowStatus RowStatus

}

alaDhcpRelayServerDestinationAddressType OBJECT-TYPE

SYNTAX InetAddressType {ipv4(1)}

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The address type of a DHCP relay server destination."

::= { alaDhcpRelayServerDestinationEntry 1 }

alaDhcpRelayServerDestinationAddress OBJECT-TYPE

SYNTAX InetAddress (SIZE(4))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The address of a DHCP relay server destination."

::= { alaDhcpRelayServerDestinationEntry 2 }

alaDhcpRelayServerDestinationRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Used to control the addition and removal of DHCP

relay server destinations."

::= { alaDhcpRelayServerDestinationEntry 3 }

--

-- Alcatel DHCP Relay Interface Admin state table

--

alaDhcpRelayInterfaceAdminStateTable OBJECT-TYPE

SYNTAX SEQUENCE OF AlaDhcpRelayInterfaceAdminStateEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An IP DHCP relay interface status table."

::= { alaDhcpRelayMIB 4 }

alaDhcpRelayInterfaceAdminStateEntry OBJECT-TYPE

SYNTAX AlaDhcpRelayInterfaceAdminStateEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An IP DHCP interface status entry."

INDEX { alaDhcpRelayInterfaceName }

::= { alaDhcpRelayInterfaceAdminStateTable 1 }

AlaDhcpRelayInterfaceAdminStateEntry ::= SEQUENCE {

alaDhcpRelayInterfaceAdminStatus INTEGER

}

alaDhcpRelayInterfaceAdminStatus OBJECT-TYPE

SYNTAX INTEGER {

enable(1),

disable(2)

}

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"DHCP relay administrative status on the interface."

::= { alaDhcpRelayInterfaceAdminStateEntry 1 }

--

-- Alcatel DHCP Relay Clear Interface Statistics Table

--

alaDhcpRelayClearStatisticsTable OBJECT-TYPE

SYNTAX SEQUENCE OF AlaDhcpRelayClearStatisticsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An IP DHCP relay clear interface statistics table."

::= { alaDhcpRelayMIB 5 }

alaDhcpRelayClearStatisticsEntry OBJECT-TYPE

SYNTAX AlaDhcpRelayClearStatisticsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An IP DHCP clear interface statistics entry."

INDEX {

alaDhcpRelayClearStatisticsInterfaceName,

alaDhcpRelayClearStatisticsIpAddressType,

alaDhcpRelayClearStatisticsIpAddress

}

::= { alaDhcpRelayClearStatisticsTable 1 }

AlaDhcpRelayClearStatisticsEntry ::= SEQUENCE {

alaDhcpRelayClearStatisticsInterfaceName SnmpAdminString,

alaDhcpRelayClearStatisticsIpAddressType InetAddressType,

alaDhcpRelayClearStatisticsIpAddress InetAddress

}

alaDhcpRelayClearStatisticsInterfaceName OBJECT-TYPE

SYNTAX SnmpAdminString(SIZE (0..63))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"IP interface statistics name for which DHCP relay is applicable."

::= { alaDhcpRelayClearStatisticsEntry 1 }

alaDhcpRelayClearStatisticsIpAddressType OBJECT-TYPE

SYNTAX InetAddressType { ipv4(1) }

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Server IP address for DHCP relay type."

::= { alaDhcpRelayClearStatisticsEntry 2 }

alaDhcpRelayClearStatisticsIpAddress OBJECT-TYPE

SYNTAX InetAddress (SIZE(4))

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Server IP address for DHCP relay."

::= { alaDhcpRelayClearStatisticsEntry 3 }

IPv4 UDP Relay Service MIB

alaGenericUdpServiceSvcGroup OBJECT-GROUP

OBJECTS

{

alaGenericUdpServiceStatTxToSvc,

alaGenericUdpServiceSvcRowStatus

}

STATUS current

DESCRIPTION

" Objects for IP DHCP service SVC."

### WebView

Should be updated accordingly.

# 19.3 FUNCTIONAL SPECIFICATIONS

UDP Relay module must be updated for command migration accordingly.

All the MIP functions must be written for new tables and need to take care of backward compatibility accordingly.