



ALL-IN-ONE

CCIE ROUTING AND SWITCHING V5.1

WRITTEN EXAM 400-101 CERT GUIDE
FOR CCNP & CCNA PROFESSIONALS

*Learn, Practice and Ace the New CCIE Written Exams with
Test Prep Resources from CCIEin8Weeks™*

PAUL ADAM, CCIE®

CCIEin8Weeks.com

2ND EDITION

**All-in-One CCIE Routing and Switching V5.1
400-101 Written Exam Cert Guide**

for CCNP and CCNA Professionals

2nd Edition

FULL GUIDE HAS 400+ PAGES

Contents at a Glance

Part 1 Network Principles

Chapter 1: Network Theory

Chapter 2: Network Implementation and operation

Chapter 3: Network Troubleshooting

Part 2 Layer 2 Technologies

Chapter 4: LAN Switching Technologies

Chapter 5: Layer 2 Multicast

Chapter 6: Layer 2 WAN Circuit Technologies

Part 3 Layer 3 Technologies

Chapter 7: Addressing Technologies

Chapter 8: Layer 2 Multicast

Chapter 9: Fundamental Routing Concepts

Chapter 10: RIPv2 (IPv4/IPv6)

Chapter 11: EIGRP (IPv4/IPv6)

Chapter 12: OSPF (v2, v3)

Chapter 13: BGP

Chapter 14: ISIS (IPv4/IPv6)

Part 4 VPN Technologies

Chapter 15: Tunneling

Chapter 16: Encryption

Part 5 Infrastructure Security

Chapter 17: Device Security

Chapter 18: Network Security

Part 6 Infrastructure Services

Chapter 19: System Management

Chapter 20: Quality of Service

Chapter 21: Network Services

Chapter 22: Network Optimization

Part 7 Evolving Technologies V1.1

Chapter 23 Cloud

Chapter 24 Network Programmability

Chapter 25 Internet of Things (IoT)

SAMPLE GUIDE

Table of Contents

| | |
|---|-----------|
| <i>Preface</i> | 24 |
| <i>What this Exam Cert Guide covers</i> | 24 |
| <i>How to use this Exam Cert Guide</i> | 25 |
| <i>What's available on the CCIEin8Weeks website</i> | 25 |
| Part 1 Network Principles | 26 |
| <i>Chapter 1: Network Theory</i> | 28 |
| <i>Describe basic software architecture differences between IOS and IOS XE</i> | 28 |
| <i>Table 1-1, shows functions of Cisco IOS XE Software Subpackages</i> | 29 |
| <i>Control plane and Forwarding plane</i> | 30 |
| <i>Table 1-2, compares classic IOS ("IOS) and IOS XE architectures</i> | 31 |
| <i>Impact to troubleshooting and performances</i> | 31 |
| <i>Table 1-3, shows comparison of troubleshooting differences between classic IOS ("IOS) and IOS XE</i> | 32 |
| <i>Identify Cisco express forwarding concepts</i> | 32 |
| <i>RIB, FIB, LFIB, Adjacency table</i> | 34 |
| <i>Routing Information Base (RIB)</i> | 34 |
| <i>Forwarding Information Base (FIB)</i> | 34 |
| <i>Label Information Base (LIB)</i> | 34 |
| <i>Adjacency Tables</i> | 34 |
| <i>Load balancing Hash</i> | 35 |
| <i>Per-Destination load balancing</i> | 35 |
| <i>Per-Packet load balancing</i> | 36 |
| <i>Polarization concept and avoidance</i> | 37 |
| <i>Explain general network challenges</i> | 37 |
| <i>Unicast flooding</i> | 37 |
| <i>Asymmetric Routing</i> | 38 |
| <i>Spanning-Tree Protocol Topology Changes</i> | 38 |
| <i>Forwarding Table Overflow</i> | 39 |
| <i>Out of order packets</i> | 39 |
| <i>Impact of micro burst</i> | 39 |
| <i>Explain IP operations</i> | 40 |
| <i>ICMP unreachable, redirect</i> | 40 |
| <i>IPv4 options, IPv6 extension headers</i> | 40 |
| <i>Table 1-5, shows IP header options and their description</i> | 40 |
| <i>Table 1-6, IPv6 Extension Headers and their Recommended Order in a Packet</i> | 41 |
| <i>IPv4 and IPv6 fragmentation</i> | 41 |
| <i>TTL</i> | 42 |

| | |
|--|----|
| <i>IP MTU</i> | 42 |
| <i>Explain TCP operations</i> | 43 |
| <i>IPv4 and IPv6 PMTU</i> | 43 |
| <i>Latency</i> | 44 |
| <i>Windowing</i> | 44 |
| <i>Bandwidth delay product</i> | 44 |
| <i>Global synchronization</i> | 45 |
| <i>Options</i> | 45 |
| <i>Options have up to three fields:</i> | 45 |
| <i>Explain UDP operations</i> | 45 |
| <i>Starvation</i> | 46 |
| <i>Latency</i> | 46 |
| <i>RTP/RTCP concepts</i> | 46 |
| <i>Exam Essentials</i> | 46 |
| <i>Chapter 2: Network Implementation and Operation</i> | 50 |
| <i>Evaluate proposed changes to a network</i> | 50 |
| <i>Changes to routing protocol parameters</i> | 50 |
| <i>Migrate parts of a network to IPv6</i> | 50 |
| <i>Routing protocol migration</i> | 50 |
| <i>Further Reading</i> | 51 |
| <i>Adding multicast support</i> | 51 |
| <i>Further Reading</i> | 52 |
| <i>Migrate spanning tree protocol</i> | 52 |
| <i>PVST+ to MST Migration</i> | 53 |
| <i>STP to RSTP (802.1w) or MSTP (802.1s)</i> | 53 |
| <i>Configuration Steps:</i> | 53 |
| <i>Further Reading</i> | 54 |
| <i>Evaluate impact of new traffic on existing QoS design</i> | 54 |
| <i>Exam Essentials</i> | 54 |
| <i>Chapter 3: Network Troubleshooting</i> | 57 |
| <i>Use IOS troubleshooting tools</i> | 57 |
| <i>Further Reading</i> | 57 |
| <i>Debug, conditional debug</i> | 57 |
| <i>Ping, traceroute with extended options</i> | 57 |
| <i>Further Reading</i> | 58 |
| <i>Embedded packet capture</i> | 58 |
| <i>Further Reading</i> | 58 |
| <i>Performance monitor</i> | 58 |
| <i>Further Reading</i> | 59 |
| <i>Apply troubleshooting methodologies</i> | 59 |
| <i>Further Reading</i> | 59 |
| <i>Interpret packet capture</i> | 59 |

| | |
|---|-----------|
| <i>Using Wireshark trace analyzer</i> | 59 |
| <i>Further Reading</i> | 60 |
| <i>Using IOS Embedded Packet Capture (EPC)</i> | 60 |
| <i>Basic EPC Configuration</i> | 60 |
| <i>Further Reading</i> | 61 |
| <i>Exam Essentials</i> | 61 |
| Part 2 Layer 2 Technologies | 62 |
| <i>Chapter 4: LAN Switching Technologies</i> | 64 |
| <i>Implement and troubleshoot switch administration</i> | 64 |
| <i>Managing MAC address table</i> | 64 |
| <i>Further Reading</i> | 64 |
| <i>Errdisable recovery</i> | 64 |
| <i>Further Reading</i> | 66 |
| <i>L2 MTU</i> | 66 |
| <i>Implement and troubleshoot layer 2 protocols</i> | 66 |
| <i>CDP, LLDP</i> | 66 |
| <i>Further Reading</i> | 66 |
| <i>UDLD</i> | 67 |
| <i>Further Reading</i> | 67 |
| <i>Implement and troubleshoot VLAN</i> | 67 |
| <i>Access ports</i> | 67 |
| <i>VLAN database</i> | 67 |
| <i>Normal, extended VLAN, voice VLAN</i> | 68 |
| <i>Table 4-1, shows various default VLANs and the respective L2 protocols</i> | 68 |
| <i>Implement and troubleshoot trunking</i> | 69 |
| <i>VTPv1, VTPv2, VTPv3, VTP pruning</i> | 70 |
| <i>Table 4-2, summaries different VTP versions and their limitations</i> | 70 |
| <i>Dot1Q</i> | 70 |
| <i>Native VLAN</i> | 71 |
| <i>Manual pruning</i> | 71 |
| <i>Implement and troubleshoot EtherChannel</i> | 71 |
| <i>Further Reading</i> | 72 |
| <i>LACP, PAgP, manual</i> | 72 |
| <i>Further Reading</i> | 73 |
| <i>Layer 2, layer 3, Load-balancing</i> | 73 |
| <i>Table 4-3, shows various platforms and the load balancing options that are available</i> | 73 |
| <i>Further Reading</i> | 74 |
| <i>Etherchannel misconfiguration guard</i> | 74 |
| <i>Implement and troubleshoot spanning-tree</i> | 74 |
| <i>Further Reading</i> | 75 |
| <i>PVST+/RPVST+/MST</i> | 75 |

| | |
|--|-----------|
| <i>Table 4-4, summarizes different STP versions and their limitations.....</i> | <i>76</i> |
| <i>Further Reading</i> | <i>76</i> |
| <i>Switch priority, port priority, path cost, STP timers.....</i> | <i>76</i> |
| <i>Further Reading</i> | <i>78</i> |
| <i>Port Fast, BPDUguard, BPDUfilter.....</i> | <i>78</i> |
| <i>Loop Guard, Root Guard</i> | <i>79</i> |
| <i>Further Reading</i> | <i>79</i> |
| <i>Implement and troubleshoot other LAN switching technologies.....</i> | <i>79</i> |
| <i>SPAN, RSPAN, ERSPAN.....</i> | <i>79</i> |
| <i>Further Reading</i> | <i>80</i> |
| <i>Describe chassis virtualization and aggregation technologies</i> | <i>80</i> |
| <i>Multi-chassis</i> | <i>80</i> |
| <i>Further Reading</i> | <i>81</i> |
| <i>VSS concepts</i> | <i>81</i> |
| <i>Alternative to STP.....</i> | <i>81</i> |
| <i>Further Reading</i> | <i>82</i> |
| <i>StackWise</i> | <i>82</i> |
| <i>Table 4-5, shows rules and their respective priority order.....</i> | <i>82</i> |
| <i>Excluding specific platform implementation.....</i> | <i>83</i> |
| <i>Describe spanning-tree concepts.....</i> | <i>83</i> |
| <i>Further Reading</i> | <i>84</i> |
| <i>Compatibility between MST and RSTP.....</i> | <i>84</i> |
| <i>Further Reading</i> | <i>84</i> |
| <i>STP dispute, STP bridge assurance.....</i> | <i>84</i> |
| <i>Further Reading</i> | <i>85</i> |
| <i>Exam Essentials.....</i> | <i>85</i> |
| <i>Chapter 5: Layer 2 Multicast.....</i> | <i>88</i> |
| <i>Implement and troubleshoot IGMP</i> | <i>88</i> |
| <i>Further Reading</i> | <i>88</i> |
| <i>IGMPv1, IGMPv2, IGMPv3.....</i> | <i>88</i> |
| <i>Table 5-1, shows IGMPv2 intervals and their default values.....</i> | <i>90</i> |
| <i>Further Reading</i> | <i>91</i> |
| <i>IGMP Snooping.....</i> | <i>91</i> |
| <i>IGMP Querier.....</i> | <i>92</i> |
| <i>Further Reading</i> | <i>92</i> |
| <i>IGMP Filter</i> | <i>92</i> |
| <i>Further Reading</i> | <i>93</i> |
| <i>IGMP proxy.....</i> | <i>93</i> |
| <i>Further Reading</i> | <i>93</i> |
| <i>Explain MLD.....</i> | <i>93</i> |
| <i>MLD Versions</i> | <i>94</i> |
| <i>Explain PIM Snooping.....</i> | <i>94</i> |

| | |
|--|------------|
| <i>PIM Snooping Configuration Guidelines and Restrictions</i> | 95 |
| <i>Further Reading</i> | 96 |
| <i>Exam Essentials</i> | 96 |
| <i>Chapter 6: Layer 2 WAN Circuit Technologies</i> | 98 |
| <i>Implement and troubleshoot HDLC</i> | 98 |
| <i>Implement and troubleshoot PPP</i> | 98 |
| <i>Authentication (PAP, CHAP)</i> | 98 |
| <i>PPPoE</i> | 98 |
| <i>MLPPP</i> | 99 |
| <i>Multilink PPP Bundles and PPP Links</i> | 100 |
| <i>Describe WAN rate-based Ethernet circuits</i> | 101 |
| <i>Further Reading</i> | 101 |
| <i>Metro and WAN Ethernet topologies</i> | 101 |
| <i>Table 6-1, shows breakdown of metro ethernet services into port or VLAN based categories</i> | 102 |
| <i>Use of rate-limited WAN Ethernet services</i> | 103 |
| <i>Ethernet Private Line (EPL)</i> | 103 |
| <i>Ethernet Virtual Private Line (EVPL)</i> | 103 |
| <i>Further Reading</i> | 103 |
| <i>Exam Essentials</i> | 103 |
| Part 3 Layer 3 Technologies | 105 |
| <i>Chapter 7: Addressing Technologies</i> | 107 |
| <i>Address types, VLSM</i> | 107 |
| <i>Further Reading</i> | 107 |
| <i>ARP</i> | 107 |
| <i>Further Reading</i> | 108 |
| <i>Identify, implement and troubleshoot IPv6 addressing and subnetting</i> | 108 |
| <i>Unicast, multicast</i> | 108 |
| <i>Table 7-1, shows various IPv6 address types and respective formats</i> | 109 |
| <i>Further Reading</i> | 109 |
| <i>EUI-64</i> | 109 |
| <i>ND, RS/RA</i> | 110 |
| <i>Router Solicitation</i> | 110 |
| <i>Further Reading</i> | 111 |
| <i>Autoconfig/SLAAC, temporary addresses (RFC 4941)</i> | 111 |
| <i>Global prefix configuration feature</i> | 112 |
| <i>Further Reading</i> | 112 |
| <i>DHCP protocol operations</i> | 112 |
| <i>DHCP Server Function</i> | 112 |
| <i>Table 7-2, shows various DHCP messages and their intended use</i> | 113 |
| <i>Client Function</i> | 114 |

| | |
|---|-----|
| <i>Further Reading</i> | 114 |
| <i>SLAAC/DHCPv6 interaction</i> | 114 |
| <i>Stateful, Stateless DHCPv6</i> | 115 |
| <i>DHCPv6 prefix delegation</i> | 115 |
| <i>Exam Essentials</i> | 116 |
| <i>Chapter 8: Layer 3 Multicast</i> | 118 |
| <i>Troubleshoot reverse path forwarding</i> | 118 |
| <i>RPF failure</i> | 118 |
| <i>RPF failure with tunnel interface</i> | 118 |
| <i>Further Reading</i> | 118 |
| <i>Implement and troubleshoot IPv4 protocol independent multicast</i> | 118 |
| <i>PIM dense mode, sparse mode, sparse-dense mode</i> | 119 |
| <i>Static RP, auto-RP, BSR</i> | 120 |
| <i>Further Reading</i> | 120 |
| <i>Bidirectional PIM</i> | 121 |
| <i>Further Reading</i> | 121 |
| <i>Source-specific multicast</i> | 121 |
| <i>Further Reading</i> | 122 |
| <i>Group to RP mapping</i> | 122 |
| <i>Table 8-1, shows various mechanisms for disseminating RP information</i> | 122 |
| <i>Further Reading</i> | 122 |
| <i>Multicast boundary</i> | 123 |
| <i>Further Reading</i> | 123 |
| <i>Implement and troubleshoot multicast source discovery protocol</i> | 123 |
| <i>Intra-domain MSDP (anycast RP)</i> | 123 |
| <i>SA filter</i> | 124 |
| <i>Further Reading</i> | 124 |
| <i>Describe IPv6 multicast</i> | 124 |
| <i>IPv6 multicast addresses</i> | 124 |
| <i>Table 8-2, shows IPv6 multicast address format</i> | 125 |
| <i>PIMv6</i> | 125 |
| <i>Exam Essentials</i> | 125 |
| <i>Chapter 9: Fundamental Routing Concepts</i> | 127 |
| <i>Implement and troubleshoot static routing</i> | 127 |
| <i>Implement and troubleshoot default routing</i> | 127 |
| <i>Compare routing protocol types</i> | 128 |
| <i>Distance vector</i> | 128 |
| <i>Further Reading</i> | 128 |
| <i>Link state</i> | 128 |
| <i>Further Reading</i> | 128 |
| <i>Path vector</i> | 128 |
| <i>Implement, optimize and troubleshoot administrative distance</i> | 129 |

| | |
|---|-----|
| <i>Implement and troubleshoot passive interface</i> | 129 |
| <i>Implement and troubleshoot VRF lite</i> | 129 |
| <i>Implement, optimize and troubleshoot filtering with any routing protocol</i> | 130 |
| <i>Implement, optimize and troubleshoot redistribution between any routing protocol</i> | 131 |
| <i>Distance Vector Protocols</i> | 131 |
| <i>Link State Protocols</i> | 133 |
| <i>Further Reading</i> | 133 |
| <i>Implement, optimize and troubleshoot manual and auto summarization with any routing protocol</i> | 133 |
| <i>Implement, optimize and troubleshoot policy-based routing</i> | 134 |
| <i>Further Reading</i> | 134 |
| <i>Identify and troubleshoot sub-optimal routing</i> | 134 |
| <i>Implement and troubleshoot bidirectional forwarding detection</i> | 135 |
| <i>Implement and troubleshoot loop prevention mechanisms</i> | 135 |
| <i>Route tagging, filtering</i> | 136 |
| <i>Implement and troubleshoot routing protocol authentication</i> | 137 |
| <i>MD5</i> | 137 |
| <i>OSPF Authentication</i> | 137 |
| <i>RIP and EIGRP</i> | 137 |
| <i>Further Reading</i> | 138 |
| <i>Key-chain</i> | 138 |
| <i>EIGRP HMAC SHA2-256 bit</i> | 138 |
| <i>Configuration Steps</i> | 139 |
| <i>Further Reading</i> | 139 |
| <i>OSPFv2 SHA1-196bit</i> | 139 |
| <i>OSPFv3 IPsec authentication</i> | 140 |
| <i>Further Reading</i> | 141 |
| <i>Exam Essentials</i> | 141 |
| <i>Chapter 10: RIPv2 (IPv4/IPv6)</i> | 144 |
| <i>Implement and troubleshoot RIPv2</i> | 144 |
| <i>Further Reading</i> | 144 |
| <i>Describe RIPv6 (RIPng)</i> | 144 |
| <i>Further Reading</i> | 144 |
| <i>Exam Essentials</i> | 144 |
| <i>Chapter 11: EIGRP (IPv4/IPv6)</i> | 147 |
| <i>Describe packet types</i> | 147 |
| <i>Packet types (hello, query, update, and such)</i> | 147 |
| <i>Further Reading</i> | 148 |
| <i>Route types (internal, external)</i> | 148 |
| <i>Implement and troubleshoot neighbor relationship</i> | 149 |
| <i>Multicast, unicast EIGRP peering</i> | 149 |
| <i>Further Reading</i> | 149 |

| | |
|--|-----|
| <i>OTP point-to-point peering</i> | 149 |
| <i>OTP route-reflector peering</i> | 150 |
| <i>OTP multiple service providers scenario</i> | 150 |
| <i>Further Reading</i> | 151 |
| <i>Implement and troubleshoot loop free path selection</i> | 151 |
| <i>RD, FD, FC, successor, feasible successor</i> | 151 |
| <i>Further Reading</i> | 152 |
| <i>Classic metric</i> | 152 |
| <i>Wide metric</i> | 152 |
| <i>Implement and troubleshoot operations</i> | 152 |
| <i>Topology table, update, query, active, passive</i> | 153 |
| <i>Further Reading</i> | 154 |
| <i>Stuck in active</i> | 154 |
| <i>Graceful shutdown</i> | 155 |
| <i>Implement and troubleshoot EIGRP stub</i> | 155 |
| <i>Stub</i> | 155 |
| <i>Leak-map</i> | 156 |
| <i>Further Reading</i> | 156 |
| <i>Implement and troubleshoot load-balancing</i> | 156 |
| <i>Equal-cost</i> | 156 |
| <i>Unequal-cost</i> | 156 |
| <i>Add-path</i> | 156 |
| <i>Implement EIGRP (multi-address) named mode</i> | 157 |
| <i>Types of families</i> | 157 |
| <i>IPv4 address-family</i> | 157 |
| <i>IPv6 address-family</i> | 157 |
| <i>Implement, troubleshoot and optimize EIGRP convergence and scalability</i> | 157 |
| <i>Describe fast convergence requirements</i> | 157 |
| <i>Further Reading</i> | 158 |
| <i>Control query boundaries</i> | 158 |
| <i>IP FRR/fast reroute (single hop)</i> | 159 |
| <i>Summary leak-map and metric</i> | 159 |
| <i>Exam Essentials</i> | 159 |
| <i>Chapter 12: OSPF (v2 and v3)</i> | 163 |
| <i>Describe packet types</i> | 163 |
| <i>LSA types (1, 2, 3, 4, 5, 7, 9, 10)</i> | 163 |
| <i>Table 12-1 summarizes various LSA types and their description</i> | 163 |
| <i>Table 12-2, shows various OSPF network types and traffic that are allowed</i> | 164 |
| <i>Route types (N1, N2, E1, E2)</i> | 165 |
| <i>Implement and troubleshoot neighbor relationship</i> | 165 |
| <i>Further Reading</i> | 167 |
| <i>Implement and troubleshoot OSPFv3 address-family support</i> | 168 |

| | |
|---|-----|
| Configuration Steps | 168 |
| Verification Steps | 168 |
| Further Reading | 169 |
| IPv4/v6 address-family..... | 169 |
| Implement and troubleshoot network types, area types and router types..... | 171 |
| Point-to-point, multipoint, broadcast, non-broadcast..... | 171 |
| Point-to-Point Sub-interfaces..... | 171 |
| Point-to-Multipoint Interfaces | 171 |
| Broadcast Interfaces..... | 171 |
| Table 12-3, shows the various OSPF network types and their associated default set of timers | 171 |
| LSA types, area type: backbone, normal, transit, stub, NSSA, totally stub..... | 172 |
| Table 12-4, shows the differences between the types of the OSPF areas..... | 172 |
| Internal router, ABR, ASBR..... | 172 |
| Virtual link..... | 173 |
| Implement and troubleshoot path preference..... | 173 |
| Further Reading | 174 |
| Implement and troubleshoot operations..... | 174 |
| General operations | 174 |
| Further Reading | 174 |
| Graceful shutdown..... | 174 |
| Generic TTL Security Mechanism (GTSM)..... | 174 |
| Further Reading | 175 |
| Implement, troubleshoot and optimize OSPF convergence and scalability | 175 |
| Metrics | 175 |
| LSA throttling, SPF tuning, fast hello..... | 176 |
| LSA propagation control (area types, ISPF)..... | 178 |
| IP FRR/fast reroute (single and multi hop)..... | 179 |
| Further Reading | 179 |
| OSPFv3 prefix suppression..... | 179 |
| Further Reading | 180 |
| Exam Essentials..... | 180 |
| Chapter 13: BGP..... | 183 |
| Describe, implement and troubleshoot peer relationships | 183 |
| Peer-group, template..... | 183 |
| Further Reading | 184 |
| Active, passive..... | 184 |
| States, timers..... | 184 |
| Dynamic neighbors..... | 186 |
| Implement and troubleshoot IBGP and EBGP..... | 187 |
| EBGP, IBGP | 187 |
| 4-bytes AS number..... | 187 |

| | |
|--|-----|
| <i>Private AS</i> | 188 |
| <i>Explain attributes and best-path selection</i> | 189 |
| <i>Further Reading</i> | 189 |
| <i>Implement, optimize and troubleshoot routing policies</i> | 189 |
| <i>Attribute manipulation</i> | 190 |
| <i>BGP Path Attributes</i> | 190 |
| <i>Next-Hop Attribute</i> | 190 |
| <i>Local Preference Attribute</i> | 190 |
| <i>Origin Attribute</i> | 190 |
| <i>AS_Path Attribute</i> | 191 |
| <i>MED Attribute</i> | 191 |
| <i>Community Attribute</i> | 191 |
| <i>Atomic Aggregate and Aggregator Attributes</i> | 191 |
| <i>Table 13-1, shows BGP attributes and the category they belong to</i> | 192 |
| <i>Conditional advertisement</i> | 192 |
| <i>Outbound route filtering</i> | 193 |
| <i>Communities, extended communities</i> | 193 |
| <i>Multi-homing</i> | 193 |
| <i>Implement and troubleshoot scalability</i> | 194 |
| <i>Route-reflector, cluster</i> | 194 |
| <i>Confederations</i> | 195 |
| <i>Further Reading</i> | 195 |
| <i>Aggregation, AS set</i> | 195 |
| <i>Impact of the Use of suppress-map with Other Configuration Commands</i> | 196 |
| <i>Further Reading</i> | 196 |
| <i>Implement and troubleshoot multiprotocol BGP</i> | 196 |
| <i>IPv4, IPv6, VPN address-family</i> | 197 |
| <i>Further Reading</i> | 198 |
| <i>Implement and troubleshoot AS path manipulations</i> | 198 |
| <i>Local AS, allow AS in, remove private AS</i> | 198 |
| <i>Prepend</i> | 198 |
| <i>Regexp</i> | 198 |
| <i>Table 13-2, shows various regular expressions and their description</i> | 198 |
| <i>Further Reading</i> | 199 |
| <i>Implement and troubleshoot other features</i> | 199 |
| <i>Multipath</i> | 199 |
| <i>BGP synchronization</i> | 199 |
| <i>Soft reconfiguration, route refresh</i> | 200 |
| <i>Describe BGP fast convergence features</i> | 200 |
| <i>Prefix independent convergence (PIC)</i> | 200 |
| <i>Add-path</i> | 201 |
| <i>Next-hop address tracking</i> | 201 |

| | |
|---|------------|
| <i>Exam Essentials</i> | 202 |
| <i>Chapter 14: ISIS (IPv4/IPv6)</i> | 205 |
| <i>Describe basic ISIS network</i> | 205 |
| <i>Single area, Single topology</i> | 206 |
| <i>Describe neighbor relationship</i> | 207 |
| <i>Table 14-1, describes the configuration steps to enable ISIS</i> | 207 |
| <i>Further Reading</i> | 208 |
| <i>Describe network types, levels and router types</i> | 208 |
| <i>Network Service Access Point (NSAP) addressing</i> | 208 |
| <i>Further Reading</i> | 209 |
| <i>Point-to-point, broadcast</i> | 209 |
| <i>Describe operations</i> | 209 |
| <i>Describe optimization features</i> | 209 |
| <i>Metrics, wide metric</i> | 210 |
| <i>Exam Essentials</i> | 210 |
| Part 4 VPN Technologies | 212 |
| Chapter 15: Tunneling | 214 |
| <i>Implement and troubleshoot MPLS operations</i> | 214 |
| <i>Label stack, LSR, LSP</i> | 214 |
| <i>Further Reading</i> | 215 |
| <i>LDP</i> | 215 |
| <i>Further Reading</i> | 216 |
| <i>MPLS ping, MPLS traceroute</i> | 216 |
| <i>Further Reading</i> | 217 |
| <i>Implement and troubleshoot basic MPLS L3VPN</i> | 217 |
| <i>L3VPN, CE, PE, P</i> | 217 |
| <i>Extranet (route leaking)</i> | 218 |
| <i>Further Reading</i> | 218 |
| <i>Implement and troubleshoot encapsulation</i> | 218 |
| <i>GRE</i> | 218 |
| <i>Dynamic GRE</i> | 218 |
| <i>LISP encapsulation principles supporting EIGRP OTP</i> | 219 |
| <i>Control Plane</i> | 220 |
| <i>Data Plane</i> | 220 |
| <i>How EIGRP Over the Top Works</i> | 221 |
| <i>Further Reading</i> | 221 |
| <i>Implement and troubleshoot DMVPN (single hub)</i> | 221 |
| <i>NHRP</i> | 221 |
| <i>Further Reading</i> | 222 |
| <i>DMVPN with IPsec using pre-shared key</i> | 222 |
| <i>QoS profile</i> | 228 |

| | |
|---|------------|
| <i>Pre-classify</i> | 230 |
| <i>Further Reading</i> | 230 |
| <i>Describe IPv6 tunneling techniques</i> | 231 |
| <i>6in4, 6to4</i> | 231 |
| <i>Further Reading</i> | 231 |
| <i>ISATAP</i> | 231 |
| <i>6RD</i> | 232 |
| <i>6rd Basic Configuration Guidelines</i> | 232 |
| <i>Further Reading</i> | 234 |
| <i>6VPE</i> | 234 |
| <i>Further Reading</i> | 235 |
| <i>Describe basic layer 2 VPN — wireline</i> | 235 |
| <i>L2TPv3 general principles</i> | 235 |
| <i>Further Reading</i> | 235 |
| <i>ATOM general principles</i> | 235 |
| <i>Further Reading</i> | 236 |
| <i>Describe basic L2VPN — LAN services</i> | 236 |
| <i>MPLS-VPLS general principles</i> | 236 |
| <i>Further Reading</i> | 237 |
| <i>OTV general principles</i> | 237 |
| <i>Table 15-1 shows the OTV entities/roles and their description</i> | 238 |
| <i>Further Reading</i> | 239 |
| <i>Exam Essentials</i> | 239 |
| <i>Chapter 16: Encryption</i> | 242 |
| <i>Implement and troubleshoot IPsec with pre-shared key</i> | 242 |
| <i>IPv4 site to IPv4 site</i> | 242 |
| <i>Further Reading</i> | 243 |
| <i>IPv6 in IPv4 tunnels</i> | 243 |
| <i>Further Reading</i> | 244 |
| <i>Virtual tunneling Interface (VTI)</i> | 244 |
| <i>Further Reading</i> | 246 |
| <i>Describe GET VPN</i> | 246 |
| <i>Group Member</i> | 246 |
| <i>Key Server</i> | 246 |
| <i>Communication Flow Between Key Servers and Group Members to Update IPsec SAs</i> | 247 |
| <i>IPsec and ISAKMP Timers</i> | 248 |
| <i>Further Reading</i> | 249 |
| <i>Exam Essentials</i> | 249 |
| Part 5 Infrastructure Security | 251 |
| <i>Chapter 17: Device Security</i> | 253 |
| <i>Implement and troubleshoot IOS AAA using local database</i> | 253 |

| | |
|---|-----|
| <i>Further Reading</i> | 253 |
| <i>Implement and troubleshoot device access control</i> | 253 |
| <i>Lines (VTY, AUX, Console)</i> | 253 |
| <i>Further Reading</i> | 255 |
| <i>SNMP</i> | 255 |
| <i>Further Reading</i> | 256 |
| <i>Management plane protection</i> | 256 |
| <i>Management Plane</i> | 256 |
| <i>Management Plane Protection Feature</i> | 257 |
| <i>Benefits of the Management Plane Protection Feature</i> | 258 |
| <i>Configuring a Device for Management Plane Protection</i> | 258 |
| <i>Prerequisites</i> | 258 |
| <i>Configuration Steps</i> | 258 |
| <i>Further Reading</i> | 259 |
| <i>Password encryption</i> | 259 |
| <i>Further Reading</i> | 260 |
| <i>Implement and troubleshoot control plane policing</i> | 260 |
| <i>Further Reading</i> | 261 |
| <i>Describe device security using IOS AAA with TACACS+ and RADIUS</i> | 261 |
| <i>AAA with TACACS+ and RADIUS</i> | 261 |
| <i>Local privilege authorization fallback</i> | 262 |
| <i>Exam Essentials</i> | 263 |
| <i>Chapter 18: Network Security</i> | 265 |
| <i>Implement and troubleshoot switch security features</i> | 265 |
| <i>VACL, PACL</i> | 265 |
| <i>Further Reading</i> | 266 |
| <i>Storm Control</i> | 266 |
| <i>DHCP snooping</i> | 266 |
| <i>IP source-guard</i> | 267 |
| <i>Further Reading</i> | 267 |
| <i>Dynamic ARP inspection</i> | 267 |
| <i>Port-security</i> | 268 |
| <i>Private VLAN</i> | 268 |
| <i>Implement and troubleshoot router security features</i> | 269 |
| <i>IPv4 access control lists (standard, extended, time-based)</i> | 269 |
| <i>Further Reading</i> | 272 |
| <i>IPv6 traffic filter</i> | 272 |
| <i>Further Reading</i> | 272 |
| <i>Unicast reverse path forwarding</i> | 272 |
| <i>Implement and troubleshoot IPv6 first hop security</i> | 273 |
| <i>RA guard</i> | 273 |
| <i>Further Reading</i> | 273 |

| | |
|--|------------|
| DHCP guard..... | 273 |
| Binding table | 274 |
| Device tracking..... | 274 |
| ND inspection/snooping..... | 274 |
| Source guard | 275 |
| PACL..... | 275 |
| Describe 802.1x | 276 |
| 802.1x, EAP, RADIUS..... | 276 |
| Further Reading | 277 |
| MAC authentication bypass..... | 277 |
| Further Reading | 278 |
| Exam Essentials..... | 278 |
| Part 6 Infrastructure Services | 280 |
| Chapter 19: System Management..... | 282 |
| Implement and troubleshoot device management..... | 282 |
| Console and VTY..... | 282 |
| Telnet, HTTP, HTTPS, SSH, SCP | 282 |
| FTP, TFTP..... | 282 |
| Implement and troubleshoot SNMP..... | 283 |
| SNMP v2c, v3..... | 283 |
| Implement and troubleshoot logging..... | 284 |
| Local logging, syslog, debug, conditional debug..... | 284 |
| Further Reading | 285 |
| Timestamp..... | 285 |
| Exam Essentials..... | 285 |
| Chapter 20: Quality of Service..... | 288 |
| Implement and troubleshoot end-to-end QoS..... | 288 |
| CoS and DSCP mapping..... | 288 |
| Further Reading | 289 |
| Implement, optimize and troubleshoot QoS using MQC | 289 |
| Classification..... | 289 |
| Network based application recognition (NBAR)..... | 290 |
| Policing, shaping..... | 291 |
| Further Reading | 292 |
| Congestion management (queuing)..... | 292 |
| HQoS, sub-rate ethernet link..... | 293 |
| Congestion avoidance (WRED)..... | 294 |
| Further Reading | 295 |
| Describe layer 2 QoS | 295 |
| Further Reading | 295 |
| Queuing, scheduling..... | 295 |

| | |
|--|-----|
| <i>Exam Essentials</i> | 296 |
| <i>Chapter 21: Network Services</i> | 298 |
| <i>Implement and troubleshoot first-hop redundancy protocols</i> | 298 |
| <i>HSRP, GLBP, VRRP</i> | 298 |
| <i>HSRP Addressing</i> | 299 |
| <i>HSRP Features</i> | 299 |
| <i>Preemption</i> | 299 |
| <i>Preempt Delay</i> | 300 |
| <i>Interface Tracking</i> | 300 |
| <i>Use Burned-In Address</i> | 301 |
| <i>Multiple HSRP Groups</i> | 301 |
| <i>Configurable MAC Address</i> | 301 |
| <i>Authentication</i> | 302 |
| <i>IP Redundancy</i> | 302 |
| <i>Restrictions for VRRP</i> | 302 |
| <i>VRRP Operation</i> | 303 |
| <i>VRRP Benefits</i> | 303 |
| <i>Multiple Virtual Router Support</i> | 305 |
| <i>VRRP Router Priority and Preemption</i> | 305 |
| <i>VRRP Advertisements</i> | 306 |
| <i>VRRP Object Tracking</i> | 306 |
| <i>How VRRP Object Tracking Affects the Priority of a Device</i> | 307 |
| <i>GLBP Active Virtual Gateway</i> | 308 |
| <i>GLBP Virtual MAC Address Assignment</i> | 308 |
| <i>GLBP Virtual Gateway Redundancy</i> | 308 |
| <i>GLBP Virtual Forwarder Redundancy</i> | 308 |
| <i>GLBP Gateway Priority</i> | 309 |
| <i>GLBP Gateway Weighting and Tracking</i> | 309 |
| <i>GLBP Benefits</i> | 310 |
| <i>Further Reading</i> | 310 |
| <i>Redundancy using IPv6 RS/RA</i> | 310 |
| <i>Further Reading</i> | 311 |
| <i>Implement and troubleshoot network time protocol</i> | 311 |
| <i>Further Reading</i> | 312 |
| <i>NTP Authentication</i> | 312 |
| <i>Implement and troubleshoot IPv4 and IPv6 DHCP</i> | 313 |
| <i>DHCP client, IOS DHCP server, DHCP relay</i> | 313 |
| <i>Further Reading</i> | 314 |
| <i>DHCP options</i> | 315 |
| <i>DHCP protocol operations</i> | 315 |
| <i>Further Reading</i> | 315 |
| <i>SLAAC/DHCPv6 interaction</i> | 315 |

| | |
|---|-----|
| <i>DHCPv6 prefix delegation</i> | 316 |
| <i>Further Reading</i> | 317 |
| <i>Implement and troubleshoot IPv4 network address translation</i> | 317 |
| <i>Static NAT, dynamic NAT, policy-based NAT, PAT</i> | 317 |
| <i>Further Reading</i> | 318 |
| <i>NAT ALG</i> | 318 |
| <i>Further Reading</i> | 318 |
| <i>Describe IPv6 network address translation</i> | 318 |
| <i>NAT64</i> | 318 |
| <i>Further Reading</i> | 319 |
| <i>NPTv6</i> | 319 |
| <i>Further Reading</i> | 319 |
| <i>Exam Essentials</i> | 319 |
| <i>Chapter 22: Network Optimization</i> | 322 |
| <i>Implement and troubleshoot IP SLA</i> | 322 |
| <i>ICMP, UDP, Jitter, VoIP</i> | 322 |
| <i>Further Reading</i> | 323 |
| <i>Implement and troubleshoot tracking object</i> | 323 |
| <i>Tracking object, tracking list</i> | 323 |
| <i>Further Reading</i> | 323 |
| <i>Tracking different entities (e.g. interfaces, routes, IPSLA, and such)</i> | 323 |
| <i>Implement and troubleshoot Netflow</i> | 324 |
| <i>Netflow v5, v9</i> | 324 |
| <i>Table 22-1, describes various NetFlow versions and their description</i> | 325 |
| <i>Further Reading</i> | 325 |
| <i>Export (configuration only)</i> | 325 |
| <i>Further Reading</i> | 326 |
| <i>Implement and troubleshoot embedded event manager</i> | 326 |
| <i>EEM policy using applet</i> | 326 |
| <i>Further Reading</i> | 327 |
| <i>Identify performance routing (PfR)</i> | 327 |
| <i>Further Reading</i> | 328 |
| <i>Basic load balancing</i> | 328 |
| <i>Further Reading</i> | 329 |
| <i>Voice optimization</i> | 329 |
| <i>Delay</i> | 329 |
| <i>Jitter</i> | 330 |
| <i>Packet Loss</i> | 330 |
| <i>Mean Opinion Score (MOS)</i> | 330 |
| <i>Further Reading</i> | 331 |
| <i>Exam Essentials</i> | 331 |

| | |
|---|------------|
| Part 7 Evolving Technologies V1.1..... | 333 |
| Chapter 23: Cloud | 335 |
| <i>Compare and Contrast Public, Private, Hybrid, and Multi-cloud Design Considerations.....</i> | <i>335</i> |
| <i>Public Cloud.....</i> | <i>336</i> |
| <i>Private Cloud.....</i> | <i>336</i> |
| <i>Virtual Private Cloud (VPC).....</i> | <i>337</i> |
| <i>Hybrid Cloud.....</i> | <i>337</i> |
| <i>Multi-cloud.....</i> | <i>338</i> |
| <i>Infrastructure as a service (IaaS).....</i> | <i>339</i> |
| <i>Platform as a service (PaaS).....</i> | <i>340</i> |
| <i>Software as a Service (SaaS).....</i> | <i>341</i> |
| <i>Consolidation.....</i> | <i>342</i> |
| <i>Virtualization.....</i> | <i>342</i> |
| <i>Automation.....</i> | <i>343</i> |
| <i>Performance, Scalability, and High Availability.....</i> | <i>343</i> |
| <i>Performance.....</i> | <i>343</i> |
| <i>Scalability and High Availability.....</i> | <i>345</i> |
| <i>Security Implications, Compliance, and Policy.....</i> | <i>345</i> |
| <i>Workload Migration.....</i> | <i>347</i> |
| Describe Cloud Infrastructure and Operations..... | 348 |
| <i>Compute Virtualization (Containers and Virtual Machines).....</i> | <i>348</i> |
| <i>Installing Docker.....</i> | <i>351</i> |
| <i>Using Docker Commands.....</i> | <i>351</i> |
| <i>Connectivity (Virtual Switches, SD-WAN and SD-Access).....</i> | <i>352</i> |
| <i>Virtual Switches.....</i> | <i>353</i> |
| <i>Virtual Machine Device Queues (VMDq).....</i> | <i>354</i> |
| <i>Single Root IO Virtualization (SR-IOV).....</i> | <i>355</i> |
| <i>SD-WAN and SD-Access.....</i> | <i>355</i> |
| <i>Cisco SD-WAN Solution (formerly Viptela).....</i> | <i>356</i> |
| <i>Software-Defined Access (or SD-Access).....</i> | <i>360</i> |
| <i>Virtualization Functions (NFVI, VNF, and L4/L1).....</i> | <i>363</i> |
| <i>NFVI and VNFs.....</i> | <i>363</i> |
| <i>Virtual Topology Forwarder (VTF).....</i> | <i>364</i> |
| <i>Automation and Orchestration Tools (CloudCenter, DNA-center, and Kubernetes).....</i> | <i>365</i> |
| <i>Cisco CloudCenter Manager and Orchestrator.....</i> | <i>365</i> |
| <i>Cisco DNA Center.....</i> | <i>367</i> |
| <i>Kubernetes.....</i> | <i>368</i> |
| <i>Exam Essentials.....</i> | <i>379</i> |
| <i>Further Reading.....</i> | <i>380</i> |
| Chapter 24: Network Programmability | 382 |
| <i>Describe Architectural and Operational Considerations for a Programmable Network.....</i> | <i>382</i> |
| <i>Data Models and Structures (YANG, JSON and XML).....</i> | <i>382</i> |

| | |
|---|-----|
| YANG..... | 383 |
| YAML | 386 |
| JSON..... | 387 |
| JSON Example..... | 387 |
| XML | 388 |
| XML Example | 388 |
| Device Programmability (gRPC, NETCONF and RESTCONF) | 389 |
| gRPC..... | 389 |
| NETCONF..... | 391 |
| NETCONF Example | 392 |
| RESTCONF | 392 |
| Using RESTCONF to Retrieve Full Running Configuration | 393 |
| Using RESTCONF to Retrieve Interface Specific Attributes..... | 393 |
| Controller Based Network Design (Policy Driven Configuration and Northbound/Southbound APIs)..... | 393 |
| Policy-driven Configuration..... | 393 |
| Northbound and Southbound APIs | 394 |
| Configuration Management Tools (Agent and Agentless) and Version Control Systems (Git and SVN)..... | 396 |
| Configuration Management Tools (Agent and Agentless)..... | 396 |
| Version Control Systems (Git and SVN)..... | 397 |
| Creating Configuration Change | 400 |
| Building New Configuration | 400 |
| Testing New Configuration | 401 |
| Deploying New Configuration..... | 401 |
| Exam Essentials..... | 406 |
| Further Reading | 407 |
| Chapter 25: Internet of Things..... | 410 |
| Describe Architectural Framework and Deployment Considerations for Internet of Things (IoT) | 410 |
| IoT Technology Stack (IoT Network Hierarchy, Data Acquisition and Flow)..... | 413 |
| Embedded Systems Layer | 413 |
| Multi-Service Edge (or Access) Layer | 413 |
| Core Network Layer..... | 414 |
| Data Center Cloud Layer | 414 |
| Data Acquisition and Flow | 414 |
| IoT Standards and Protocols (characteristics within IT and OT environment)..... | 416 |
| IoT Security (network segmentation, device profiling, and secure remote access) | 418 |
| IoT Edge and Fog Computing (data aggregation and edge intelligence) | 419 |
| Data Aggregation | 419 |
| Edge Intelligence..... | 420 |
| Exam Essentials..... | 421 |

Further Reading422

SAMPLE GUIDE

Preface

Congratulations! You have taken your first step towards passing the CCIE R&S 400-101 (V5.1) written exam. This written exam cert guide is specifically geared for CCNP/CCNA certified individuals who want to step up and prepare for the CCIE written exam.

This book is dedicated to *all those souls who will never settle for less than they can be, do, share, and give!*

What this Exam Cert Guide covers

As you may already have noticed on the "Contents at a Glance" page that this guide has been formatted around the Cisco's official CCIE 400-101 (V5.1) exam topics or [curriculum](#). So as you read through parts and chapters, you know exactly where you're within your learning journey. All contents are carefully covered in enough details however still trimmed to exactly what's necessary to pass the exam. Each exam topic has "Further Reading" section, which I highly recommend you to refer to for more in-depth details to the source material.

It is also worth noting that CCNP/CCNA official exam topics nicely align with the CCIE written blueprint which makes CCIE written a perfect step up from Associate/Professional tracks.

| CCIE 400-101 R&S V5.1 Exam Topics | CCNP 300-101 Exam Topics | CCNA 200-101 Exam Topics |
|--|-------------------------------------|--|
| Network Principles | Network Principles | <ul style="list-style-type: none"> • Operations of IP Data Networks • Troubleshooting |
| Layer 2 Technologies | Layer 2 Technologies | <ul style="list-style-type: none"> • LAN Switching Technologies • WAN Technologies |
| Layer 3 Technologies | Layer 3 Technologies | <ul style="list-style-type: none"> • IP Addressing (IPv4/IPv6) • IP Routing Technologies |

FULL GUIDE HAS 400+ PAGES

| | | |
|-------------------------|-------------------------|-------------------------|
| VPN Technologies | VPN Technologies | N/A |
| Infrastructure Security | Infrastructure Security | Network Device Security |
| Infrastructure Services | Infrastructure Services | IP Services |
| Evolving Technologies | N/A | N/A |

How to use this Exam Cert Guide

This guide is written as a bridging study material for CCNP/CCNA professionals who are studying for CCIE R&S written 400-101 (V5.1) exam. I strongly suggest to take a methodical approach for exam preparation, i.e. start with a target date when you would like to sit for the actual exam and then work backwards to see what kind of study plan would work for you. I believe you can cover the entire contents within a couple months including the practice questions (refer to website). We strongly suggest you to also use other study resources in addition to bringing your networking experience to bear in order to successfully pass the exam.

What's available on the CCIEin8Weeks website

CCIEin8Weeks.com carries the extras that go hand in hand with this exam cert guide to further ensure your exam success!

Website includes:

- Seven practice quizzes (one for each section as per official curriculum), one final exam
- Four study plans, to help you track your progress or help you come up with your own plan
- CCIE Exam community forum, to help you interact with others, share knowledge, find study partners etc.
- Last but not least, this exam cert guide in printable PDF format

Part 1 Network Principles

Chapter 1: Network Theory

Chapter 2: Network Implementation and Operation

Chapter 3: Network Troubleshooting

SAMPLE GUIDE

Chapter 1: Network Theory

This chapter covers the following exam topics from Cisco's official 400-101 (v5.1) written exam curriculum.

- Basic software architecture differences between classic IOS and IOS XE
- Cisco Express Forwarding (CEF)
- General network challenges (such as unicast flooding, out of order packets, asymmetric routing and impact of microbursts)
- IP operations (such as ICMP unreachable/redirect, IPv4 options, IPv6 extension headers, IPv4/v6 fragmentation, TTL, IP MTU)
- TCP operations (IPv4/IPv6 PMTU, Maximum Segment Size-MSS, Latency, Windowing, BW delay product, global synchronization)
- UDP operations (TCP starvation / UDP dominance, latency, RTP/RTCP concepts)

SAMPLE GUIDE

Chapter 1: Network Theory

Describe basic software architecture differences between IOS and IOS XE

Cisco classic IOS has always had a monolithic software architecture, which means that it is both downloaded and run as a single binary image where all processes share the same memory address space. Monolithic and non-modular architecture leads to no memory protection between processes, as a result software defects in classic IOS code can potentially corrupt data used by other processes. It also has a run to completion scheduler, which means that the kernel does not preempt a running process — the process must make a kernel call before other processes can be scheduled and get a chance to run. Since, historically, IOS has served as an Operating System as well as providing the key Routing Infrastructure, there has always been an aspect of Platform Dependent (PD) and Platform Independent (PI) code within IOS.

In all variations of classic Cisco IOS, packet routing and forwarding (switching) are distinct functions. Routing and other protocols run as IOS processes and contribute to the formation of Routing Information Base (RIB). This is processed to generate the final IP forwarding table (FIB, Forwarding Information Base), which is used by the forwarding function of the router. On router platforms with software-based forwarding (e.g., Cisco 7200 or Cisco ISR G2) most traffic handling is done at interrupt level using Cisco Express Forwarding (CEF). This helps avoid process context switching that would need to be done otherwise to forward packets. Routing functions such as OSPF or BGP run at the process level. In routers with hardware-based forwarding, such as the Cisco ASR1000 (which runs IOS XE), ASR9000 or CRS-1 or NCS series (which run IOS XR), IOS computes the FIB in software running on route processor (RP) hardware (typically x86 CPUs) and loads it into the forwarding hardware (such as an ASIC or a network processor), which performs the actual packet forwarding function. IOS XE allows the platform dependent code to be abstracted from a single monolithic image. By moving drivers outside of IOS, IOS XE enables a more purely PI-focused IOS process. This provides a more efficient software delivery model for both the core IOS team, as well as platform developers, since the software can be developed, packaged and released independently.

The IOS XE is a POSIX based environment along with various open source software for the common drivers, tools and utilities needed to manage the system. In addition to the standard set

of off-the-shelf drivers, IOS XE also includes a set of Cisco specific drivers and associated chassis/platform management modules.

On top of the base operating system (Linux) and drivers, IOS XE provides a comprehensive set of infrastructure modules which define how software is installed, how processes are started and sequenced, how high-availability (HA) and software upgrades are performed. The core application that runs on top of this new infrastructure is the IOS feature set in the form of IOS daemon (IOSd). By running Cisco IOS, products reap the benefits of an extensive feature set for routing and switching platforms that has been built into IOS over last two decades. Finally, the evolved IOS architecture is specifically designed to accommodate other applications outside of IOS blob or IOSd. These applications can be upgraded or restarted independently of IOSd. If an application does require services from IOS, it can integrate with IOS through a set of client libraries called service points. These service points generically extend IOS information and services to outside applications such that these services are not replicated or managed separately. IOS XE is not a new network “OS” per se, it is rather an incarnation of classic IOS (“IOS”) where role of classic IOS is reduced to an application running on top of a Linux kernel. This approach also allows building routing/switching platforms that use a variety of data plane hardware (ASICs or network processors such as Cisco’s QFP or CPP) by way of the abstraction provided between control and data planes.

IOS XE permits the integration of non-IOS applications through the following mechanisms:

- Standard Linux-based environment for hosting applications;
- Extending IOS functionality into peripheral applications through well-defined APIs exported via Linux-shared client libraries;
- Provide a robust management infrastructure called Common Management Enabling Technology (COMET) that allows for CLI, XML, SNMP, and HTTP-based management of integrated applications.

Each Cisco IOS XE Software subpackage provides specific functions for the Cisco ASR 1000 Series router.

Table 1-1, shows functions of Cisco IOS XE Software Subpackages

| Package | Function |
|---------|----------|
|---------|----------|

| | |
|-----------|---|
| RPBase | Provides the operating system software for the route processor |
| RPControl | Controls the control plane processes that interface between Cisco IOS Software and the rest of the platform |
| RPAccess | Provides software required for router access <ul style="list-style-type: none"> • The non-K9 version of this subpackage is included only in consolidated packages that do not have cryptographic support. • The K9 version of this sub-package includes restricted components (Secure Sockets Layer [SSL] and Secure Shell [SSH]); consolidated packages with this subpackage are subject to export controls. |
| RPIOS | Provides the Cisco IOS Software kernel, which is where Cisco IOS Software features are stored and run; each consolidated package has a different RPIOS subpackage |
| ESPBase | Provides the ESP operating system and control processes and the ESP software |
| SIPSPA | Provides the shared port adaptor (SPA) driver and associated field-programmable device (FPD) images |
| SIPBase | Controls the Session Initiation Protocol (SIP) carrier card operating system and control processes |

Control plane and Forwarding plane

IOS XE allows development of data plane ASICs outside the IOS instance and have them program to a set of standard APIs which in turn enforces Control Plane and Data Plane processing separation. It achieves Control Plane / Data Plane separation through the introduction of the Forwarding and Feature Manager (FFM) and its standard interface to the

Forwarding Engine Driver (FED). FFM provides a set of APIs to control plane processes. FFM programs the data plane via the FED and maintains forwarding state for the entire system. The FED is the instantiation of the hardware driver for the data plane.

Table 1-2, compares classic IOS (“IOS”) and IOS XE architectures

| Software Architecture | Classic IOS (“IOS”) | IOS XE |
|--|---|---|
| Monolithic | Yes | No |
| Control and Data Plane separation (Software) | No | Yes |
| Control and Data Plane separation (Hardware) | Platforms that run classic IOS do not have clear separation of control and data plane in hardware | Platforms that run IOS XE do have clear separation of control and data plane hardware |
| Feature parity | All IOS versions contain a singular set of IOS features (E.g. all platforms in T train contain same features such as Cisco ISR G2 and 7200) | All IOS XE versions contain software that is specific to one or a set of platforms E.g., IOS XE running on Cisco ASR 900 contains different feature set than one running on Cisco ASR1000 |
| Example Platforms | All software based platform Cisco ISRs (3900, 2900, 1900, 8XX), 7200 (end of life) | Cisco ISR 4K Cisco ASR1000 series (1006, 1004, 1002, 1001) |

Impact to troubleshooting and performances

Significant differences in software architecture lead to significant differences in how you troubleshoot the platforms that run classic IOS versus IOS XE. We can break down those differences in few larger areas.

Table 1-3, shows comparison of troubleshooting differences between classic IOS (“IOS”) and IOS XE

| Troubleshooting Area | Classic IOS (“IOS”) | IOS XE |
|--|---|--|
| Methodology | All software based platforms use single set of troubleshooting approach | Different set of show/debug platform CLIs for control and data planes |
| Command Line Interface (CLI) - show and debug commands | All software based platforms use single set of CLIs | Show/Debug CLIs can be specific to a given platform (e.g. ASR1000 versus ISR 4451) |
| Files and locations | All software based platforms use single set of rules (E.g., crash dump go to internal bootflash by default) Classic IOS image naming conventions | It can be specific to a given platform (e.g. ASR1000 versus ISR 4451) ASR1000 dumps SPA driver cashdump on harddisk: by default Newer software packages naming conventions |
| Level of difficulty | Straightforward, upside of software-based platforms | Complex, downside or cost of modularity i.e. clear separation of control and data plane software and hardware leads to more involved troubleshooting |

Identify Cisco express forwarding concepts

Cisco Express Forwarding (CEF) is advanced, layer 3 IP forwarding technology. CEF optimizes network performance and scalability where networks have large and dynamic traffic patterns, such as the Internet itself.

CEF offers the following benefits:

- Improved performance—CEF is less CPU-intensive than older fast switching. As a result, more CPU processing power can be dedicated to other layer 3 services such as quality of service (QoS) and encryption.
- Scalability—CEF offers full switching capacity at each line card or blade when distributed CEF (dCEF) mode is active.
- Resilience—CEF offers switching consistency and stability in large dynamic networks. In dynamic networks, fast-switched cache entries go through high level of churn and are frequently invalidated due to routing changes. These changes can cause traffic to be process switched using the routing table, rather than fast switched using the route cache. With CEF, Forwarding Information Base (FIB) lookup table contains all known routes that exist in the routing table, it eliminates route cache maintenance and as a result avoids sub optimal forwarding scenarios that takes place with the fast-switch or process switching.

CEF uses a FIB to make IP destination prefix-based switching decisions. The FIB is conceptually similar to a routing table or information base. It maintains a mirror image of the forwarding information contained in the IP routing table. When routing or topology changes occur in the network, the IP routing table is updated, and those changes are reflected in the FIB. The FIB maintains next hop address information based on the information in the IP routing table.

Hardware based switching platforms use Content Addressable Memory (CAM) for storing the CEF related information. These tables are finite and can fill up to exhaustion, which would cause forwarding to fall back to software. Catalyst 4500, as an example, can carry up to 128K entries in Supervisor IV/V CAMs. Once those entries are filled up, it switches to software forwarding with an error message “C4K_L3HWFORWARDING-2-FWDCAMFULL”. You can verify CAM table usage by show platform hardware ip route summary command.

“show mls cef exception status” can be used on Catalyst 6500 to check on FIB TCAM usage. A switch with FIB TCAM full will repeatedly throw an error message like below:

```
%MLSCEF-DFC4-7-FIB_EXCEPTION: FIB TCAM exception, Some
entries will be software switched
```

As a result of this TCAM exception condition, connectivity is affected and might result in elevated CPU usage due to software switching.

RIB, FIB, LFIB, Adjacency table

Routing Information Base (RIB)

RIBs (Routing Information Base) maintain the network topologies and routing tables for each protocol. This would include many routes going to the same destination prefix. It is built on per routing protocol basis, so RIP and OSPF have their own copy of RIBs.

Forwarding Information Base (FIB)

FIBs are the best routes from possibly many routing protocols in the RIBs pushed down to fast forwarding lookup memory (or just DRAM for software-based platforms) for the best path(s). This is what you see in show ip route command output. There is one copy of FIB per system for centralized forwarding platforms, or one for each line card in case of distributed systems.

Label Information Base (LIB)

LIB (Label Information Base) is the software table maintained by IP/MPLS capable routers to store the details of port and the corresponding MPLS router label to be popped or pushed on incoming or outgoing MPLS packets respectively. LIB entries are populated from label-distribution protocols. LIB functions in the control plane of Cisco routers. It is used by the label distribution protocol for mapping the next hop labels.

Label forwarding information base (LFIB) is a data structure and way of managing forwarding in which destinations and incoming labels are associated with outgoing interfaces and labels. The forwarding paradigm employed by MPLS is based on the notion of label swapping. When a packet with a label is received by an Label Switching Router (LSR), the switch uses the label as an index in its LFIB to determine the outgoing interface.

Adjacency Tables

Routers or Switches in a network are considered adjacent if they can reach each other with a single hop across a link layer. In addition to the FIB, CEF uses adjacency tables to prepend

Layer 2 addressing information such as MAC addresses. The adjacency table maintains Layer 2 next-hop addresses for all FIB entries.

Load balancing Hash

In a router, act of distributing packets across multiple links based on layer 3 routing information is known as load balancing. If a router discovers multiple paths to a destination, the routing table is updated with multiple entries for that destination.

```
Router> show ip route
```

```
[...]
```

```
I      192.168.25.0/24 [115/10] via 192.168.24.6
```

```
                [115/10] via 192.168.24.10
```

```
                [115/10] via 192.168.24.14
```

```
[...]
```

Usually the paths have the same metric, however there are routing protocols that allow unequal cost (or metric) load balancing. A router learns about the existence of parallel paths through the routing protocols and builds its routing table accordingly.

The number of paths used is limited by the number of entries the given IP routing protocol puts in the routing table, the default in IOS is 4 entries for most routing protocols with the exception of BGP, where it is one entry (only the best path). The maximum number of paths that can be configured are 6. Cisco IOS supports two primary modes of load balancing, i.e. per-destination and per-packet basis.

Per-Destination load balancing

In per-destination mode all packets for a given destination are forwarded along the same path. This preserves packet order however may lead to unequal usage of the links. If one host receives the majority of the traffic all packets will use one link, leaving bandwidth on other links

unused. This is the default load balancing mode in IOS using universal algorithm. Original per-destination algorithm creates a 4-bit hash of the source and destination IP address and load balances based on this 16 (2^4) value hash.

Per-Packet load balancing

Per-packet load balancing guarantees equal load across all links however packets may arrive out-of-order at the destination as differential delay may exist for each link used. In particular, per-packet load balancing can result in unsatisfactory data transmission for video and voice streaming.

Router(config)#ip cef load-sharing algorithm ?

include-ports Algorithm that includes Layer 4 ports

original Original algorithm

tunnel Algorithm for use in tunnel only environments

universal Algorithm for use in most environments

The following load-balancing algorithms are provided for use with Cisco Express Forwarding traffic.

- Original algorithm—The original Cisco Express Forwarding load-balancing algorithm produces distortions in load sharing across multiple routers because the same algorithm was used on every router. Depending on your network environment, you should select either the universal algorithm (default) or the tunnel algorithm instead.
- Universal algorithm—The universal load-balancing algorithm allows each router on the network to make a different load sharing decision for each source-destination address pair, which resolves load-sharing imbalances. The router is set to perform universal load sharing by default.
- Tunnel algorithm—The tunnel algorithm is designed to balance the per-packet load when only a few source and destination pairs are involved.
- Include-ports algorithm—The include-ports algorithm allows you to use the Layer 4 source and destination ports as part of the load-balancing decision. This method benefits traffic streams running over equal cost paths that are not load shared because the majority of the traffic is between peer addresses that use different port numbers, such as Real-Time Protocol (RTP) streams.

Polarization concept and avoidance

CEF polarization occurs when traffic uses per destination load balancing and the same algorithm, which is default, is used throughout the network which causes traffic to not be load balanced after the first distribution.

As an example think of a layer 3 network with multiple layers or levels each with a possible path to the right or left. If 100Mbps of traffic was coming into a router, it would be load balanced 50/50, with 50Mbps to Router-right and 50Mbps to Router-left, but as Router-level-1-right & Router-level1-left will use the same algorithm to determine which path the traffic will take, but as the algorithm is identical it will be a 100/0 split, with 50Mbps going to Router-level2-right and Router-level2-left and no data going to other paths. Whenever there is an even number of ECMP available, traffic will not be distributed evenly. Level 1 and 2 represent top to down router topology.

To counter this issue, a newer algorithm called the universal algorithm was developed where a 32-bit value is added to the hashing algorithm, this value can be manually set but defaults to the highest loopback IP on the router. This is based on the concept called unique-ID/universal-ID. Hash function is known as universal-ID, a randomly generated value at the time of the router or layer 3 switch boot up that can be manually controlled. This seeds the hash function on each router with a unique ID, which ensures that the same source/destination pair hash into a different value on different routers along the path within the network. This process provides a better network-wide load-sharing and avoids the polarization issue. In order to configure a custom ID, you can use the following CLI:

```
Router(config)#ip cef load-sharing algorithm universal <id>
```

Another way to avoid polarization would be to use alternate between default (Source IP and Destination IP) and full (Source IP + Destination IP + Layer 4 ports) hashing inputs configuration at each layer of the network. Of course, this is not practical if we're talking about a large network with many layers some possibly outside the control of the given network administrator.

Explain general network challenges

Unicast flooding

Unicast flood is the unintentional behavior of a switch treating a unicast packet as a broadcast packet; a packet destined for one host is flooded or transmitted out of all the ports of a switch. The underlying cause of flooding is that destination MAC address of the packet is not in the L2 forwarding table (there is one for each VLAN) of the switch. The primary reasons for unicast flooding behavior include asymmetric routing, STP topology changes (i.e. repeated TCNs), and MAC forwarding table overflow.

Asymmetric Routing

When communication between two hosts (or endpoints of any type) take different paths on their way out and another on their way in, it is called asymmetric routing. It can also cause packets to arrive out of order if packets that are part of a given flow take different paths. Large amounts of flooded traffic might saturate low-bandwidth links causing network performance issues or complete connectivity outage to devices connected across such low-bandwidth links. An example of such situation could be a topology where there are two switches (ports in two VLANs, say A and B), two routers (doing inter-VLAN routing between A and B) and two hosts one in VLAN A and one in VLAN B. Now since the routers will proxy ARP for respective hosts as they are default gateways, switches will never be able to learn actual end hosts MAC addresses (router will rewrite them every single time to their own). Switch A and B will continue to flood traffic since they are unaware of the actual host A and host B MAC addresses. The solution approach is normally to bring the router's ARP timeout and the switch's forwarding table-aging time close to each other. This will cause the ARP packets to be broadcast, relearning must occur before the L2 forwarding table entry ages out.

Spanning-Tree Protocol Topology Changes

TCNs are triggered by a port that is transitioning to or from the forwarding state. After the TCN, even if the particular destination MAC address has aged out, flooding should not happen for long in most cases since the address will be relearned. The issue might arise when TCNs are occurring repeatedly within short period of time. The switches will constantly be fast-aging their forwarding tables so flooding will be nearly constant. Typically, a TCN is rare occurrence in a well-configured network. As said before, when the port on a switch goes up or down, there is eventually a TCN once the STP state of the port is changing to or from forwarding. However, when a port is flapping, repetitive TCNs and flooding occurs.

The solution approach would be to configure ports with the STP portfast feature and avoid TCNs when going to or from the forwarding state. Configuration of portfast on all end-device ports (such as printers, PCs, servers, and so on) should limit TCNs to a low amount.

Forwarding Table Overflow

As mentioned before, another possible but not so common cause of flooding can be overflow of the switch forwarding table. In this case, new addresses cannot be learned and packets destined to such addresses are flooded until some space becomes available in the forwarding table. New addresses will then be learned. Since most modern switches have large enough forwarding tables to accommodate MAC addresses for most designs, L2 table overflows are uncommon.

Out of order packets

Using per-packet load balancing to share the traffic load across available paths to a given destination can lead to out-of-order packets for a given data flow.

Impact of micro burst

Micro-bursting is a phenomenon where rapid bursts of data packets are sent in quick succession, leading to periods of full line-rate transmission that can overflow packet buffers of the network stack, both in network endpoints and routers and switches inside the network. Symptoms of micro bursts will manifest in the form of ignores and/or overruns (also shown as accumulated in “input error” counter within show interface output). This is indicative of receive ring and corresponding packet buffer being overwhelmed due to data bursts coming in over extremely short period of time (microseconds). You will never see a sustained data traffic within show interface’s “input rate” counter as they are averaging bits per second (bps) over 5 minutes by default (way too long to account for microbursts). You can understand microbursts from a scenario where a 3-lane highway merging into a single lane at rush hour – the capacity burst cannot exceed the total available bandwidth (i.e. single lane), but it can saturate it for a period of time.

In order to troubleshoot microbursts, you need a packet sniffer that can capture traffic over a long period of time and allow you to analyze it in the form of a graph which displays the saturation points (packet rate during microbursts versus total available bandwidth). You can

eventually trace it to the source causing the bursts (e.g. stock trading applications). You can implement large packet buffers to avoid or mitigate microbursts.

Explain IP operations

ICMP unreachable, redirect

If a router or a layer 3 switch receives a non-broadcast packet destined for itself that uses an unknown protocol, it sends an ICMP protocol unreachable message back to the source. Similarly, if the software receives a packet that it is unable to deliver to the ultimate destination because it knows of no route to the destination address, it will send an ICMP host unreachable message to the source. This feature is enabled by default.

You can disable it by using the following CLI:

```
Router(config-if)# no ip unreachable
```

IPv4 options, IPv6 extension headers

The possible options that can be put in the IPv4 header are as follows:

Table 1-5, shows IP header options and their description

| Field | Size (bits) | Description |
|---------------|-------------|--|
| Copied | 1 | Set to 1 if the options need to be copied into all fragments of a fragmented packet. |
| Option Class | 2 | A general options category. 0 is for "control" options, and 2 is for "debugging and measurement". 1, and 3 are reserved. |
| Option Number | 5 | Specifies an option |
| Option Length | 8 | Indicates the size of the entire option (including this field). This field may not exist for simple options |
| Option Data | Variable | Option-specific data. This field may not exist for simple options |

IPv6 uses two distinct types of headers:

- Main/Regular IPv6 Header

- IPv6 Extension Headers

The main IPv6 header is equivalent to the basic IPv4 one despite some field differences that are the result of lessons learned from operating IPv4.

Table 1-6, IPv6 Extension Headers and their Recommended Order in a Packet

| Order | Header Type | Next Header Code |
|-------------|--|------------------|
| 1 | Basic IPv6 Header | - |
| 2 | Hop-by-Hop Options | 0 |
| 3 | Destination Options (with Routing Options) | 60 |
| 4 | Routing Header | 43 |
| 5 | Fragment Header | 44 |
| 6 | Authentication Header | 51 |
| 7 | Encapsulation Security Payload Header | 50 |
| 8 | Destination Options | 60 |
| 9 | Mobility Header | 135 |
| | No next header | 59 |
| Upper Layer | TCP | 6 |
| Upper Layer | UDP | 17 |
| Upper Layer | ICMPv6 | 58 |

Extension headers are an intrinsic part of the IPv6 protocol and they support some basic functions and certain services. The following is a list of situations where EHs are commonly used:

- Hop-by-Hop EH is used for the support of Jumbo-grams or, with the Router Alert option, it is an integral part in the operation of Multicast Listener Discovery (MLD). Router Alert is an integral part in the operations of IPv6 Multicast through MLD) and RSVP for IPv6.
- Destination EH is used in IPv6 Mobility as well as support of certain applications.
- Routing EH is used in IPv6 Mobility and in Source Routing. It may be necessary to disable ipv6 source routing using ipv6 source-route command on routers to protect against DDoS.
- Fragmentation EH is critical in support of communication using fragmented packets (in IPv6, the traffic source must do fragmentation-routers do not perform fragmentation of the packets they forward)
- Mobility EH is used in support of Mobile IPv6 service
- Authentication EH is similar in format and use to the IPv4 authentication header
- Encapsulating Security Payload EH is similar in format and use to the IPv4 ESP header. All information following the Encapsulating Security Header (ESH) is encrypted and obfuscated and for that reason, it is invisible to intermediary network devices.

SAMPLE GUIDE