Overview

This chapter provides an overview of the Cisco NX-OS software.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the “New and Changed Information” section or the "Feature History" table.

Software Compatibility

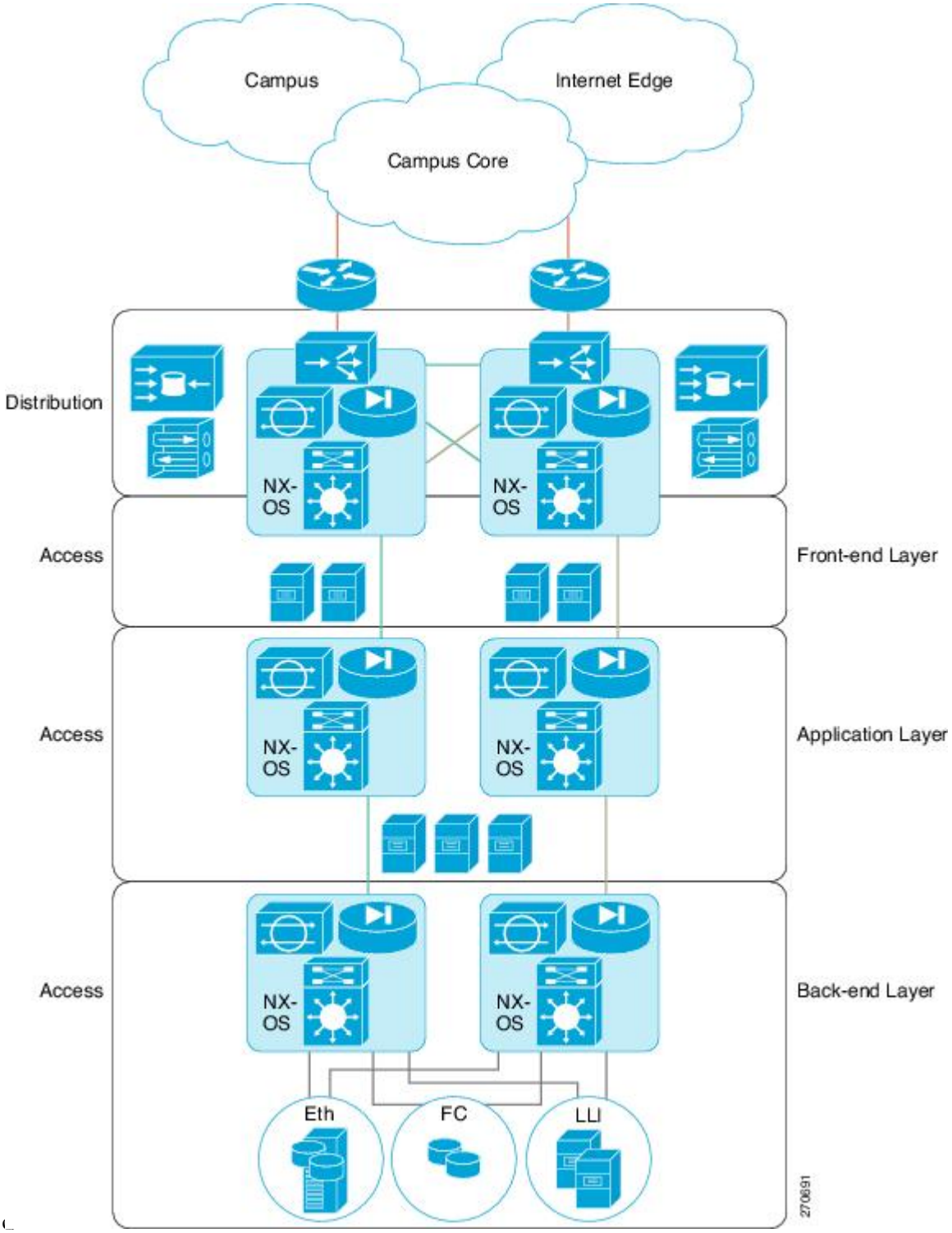
The Cisco NX-OS software interoperates with Cisco products that run any variant of the Cisco IOS software. The Cisco NX-OS software also interoperates with any networking operating system that conforms to the IEEE and RFC compliance standards.

Common Software Throughout the Data Center

The Cisco NX-OS software provides a unified operating system that is designed to run all areas of the data center network including the LAN and Layer 4 through Layer 7 network services.

Figure 1: Cisco NX-OS in a Data Center

This figure shows an overview of the Cisco NX-OS software in the data



Modular Software Design

The Cisco NX-OS software supports distributed multithreaded processing on symmetric multiprocessors (SMPs), multi-core CPUs, and distributed data module processors. The Cisco NX-OS software offloads computationally intensive tasks, such as hardware table programming, to dedicated processors distributed across the data modules. The modular processes are created on demand, each in a separate protected memory space. Processes are started and system resources are allocated only when you enable a feature. A real-time preemptive scheduler helps to ensure the timely processing of critical functions.

Virtual Device Contexts

The Cisco NX-OS software can segment system and hardware resources into virtual contexts that emulate virtual devices. Each virtual device context (VDC) has its own software processes, dedicated hardware resources (interfaces), and an independent management environment. With VDCs, you can consolidate separate networks onto a common infrastructure, which maintains the administrative boundary separation and fault isolation characteristics of physically separate networks, and provides many of the operational cost benefits of a single infrastructure. For more information, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide*.

Serviceability

The Cisco NX-OS software has serviceability functions that allow the device to respond to network trends and events. These features help you with network planning and improving response times.

Switched Port Analyzer

The Switched Port Analyzer (SPAN) feature allows you to analyze all traffic between ports (called the SPAN source ports) by nonintrusively directing the SPAN session traffic to a SPAN destination port that has an external analyzer attached to it. For more information about SPAN, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide*.

Ethanalyzer

Ethanalyzer is a Cisco NX-OS protocol analyzer tool based on the Wireshark (formerly Ethereal) open source code. Ethanalyzer is a command-line version of Wireshark for capturing and decoding packets. You can use Ethanalyzer to troubleshoot your network and analyze the control-plane traffic. For more information about Ethanalyzer, see the *Cisco Nexus 7000 Series NX-OS Troubleshooting Guide*.

Call Home

The Call Home feature continuously monitors hardware and software components to provide e-mail-based notification of critical system events. A versatile range of message formats is available for optimal compatibility with pager services, standard e-mail, and XML-based automated parsing applications. It offers alert grouping capabilities and customizable destination profiles.You can use this feature, for example, to directly page a network support engineer, send an e-mail message to a network operations center (NOC), and employ Cisco AutoNotify services to directly generate a case with the Cisco Technical Assistance Center (TAC). For more

information about Call Home, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 6.x*.

Online Diagnostics

Cisco generic online diagnostics (GOLD) verify that hardware and internal data paths are operating as designed. Boot-time diagnostics, continuous monitoring, and on-demand and scheduled tests are part of the Cisco GOLD feature set. GOLD allows rapid fault isolation and continuous system monitoring. For information about configuring GOLD, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 6.x*.

Embedded Event Manager

Cisco Embedded Event Manager (EEM) is a device and system management feature that helps you to customize behavior based on network events as they happen. For information about configuring EEM, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 6.x*.

NetFlow

The Cisco NX-OS NetFlow implementation supports version 5 and version 9 exports. It also supports the Flexible NetFlow configuration model and hardware-based Sampled NetFlow for enhanced scalability. For more information about NetFlow, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 6.x*.

Consistency Checker

Consistency Checker — Cisco NX-OS Release 8.2(1)

This section describes how to use the Consistency Checker CLIs to collect information on various table states within the software and the hardware for Cisco NX-OS Release 8.2(1).

Consistency checker compares the software state of the supervisor, with the hardware state of supported I/O modules. If there is any inconsistency, it flags the issue immediately. This helps to reduce increased troubleshooting time at a later period. Consistency checker supplements basic troubleshooting, and helps to identify scenarios where inconsistent state between software and hardware tables are causing issues in the network, thereby reducing the mean time to resolve the issue.

Consistency Checker is a serviceability tool that performs the following functions:

- Checks for consistency between software and hardware tables.
- Alerts administrators upon finding any inconsistencies.
- Helps to speed up fault isolation.

The Consistency Checker feature verifies the consistency between the software and the hardware for the following parameters in Cisco NX-OS Release 8.2(1). Except for Persistent Storage Service (PSS) consistency checker, all other features are supported since Cisco NX-OS Release 8.0(1) and are enhanced in Cisco NX-OS Release 8.2(1). Consistency checker is supported on M3 and F3 modules. Users can execute the **show consistency-checker all** command to perform consistency check for all components/features.

The following consistency checker components are supported in Cisco NX-OS Release 8.2(1):

- FabricPath
- Interface-properties
- Layer 2 Unicast
- Layer 2 Multicast
- L3-Interface Tables
- Link-state
- Proxy Forwarding
- Spanning-Tree
- Persistent Storage Service (PSS)

FabricPath

The FabricPath Consistency Checker verifies the programming consistency for the following FabricPath parameters:

- FTAG-state
- GPC-membership (Gateway Port-Channel, which is used internally for FabricPath forwarding, and this does not refer to the user-configured port-channels).

Interface-properties

The Interface-properties Consistency Checker verifies the programming consistency between software and hardware for EthPM tables (Ethernet Port Manager) including the following parameters:

- Link state
- Interface MTU
- Flow control
- FEX fabric port
- Native VLAN

Layer 2 Unicast

The Layer 2 Unicast Consistency Checker verifies the programming consistency between software and hardware tables for classical Ethernet (CE) Layer 2 unicast mac address entries.

Layer 2 Multicast

The Layer 2 Multicast Consistency Checker verifies the programming consistency between software and hardware tables for Layer 2 IGMP snooping entries in classical Ethernet (CE) topologies.

L3-Interface Tables

The L3-Interface Consistency Checker verifies the programming consistency between software and hardware for Layer 3-interface ingress and egress forwarding tables.

L3-interace consistency checker is supported only on the M3 and F3 VDCs in Cisco NX-OS Release 8.2(1). It is not supported on the VDC combination that contains a module other than M3 or F3.

Link-state

The Link-state Consistency Checker verifies the programming consistency between software and hardware for the link-state status of the interfaces.

Spanning-Tree

The Spanning-Tree Consistency Checker verifies the programming consistency between software and hardware tables for the Spanning-Tree state.

Persistent Storage Service (PSS)

The PSS Consistency Checker verifies the consistency between run-time data and data stored in PSS for the following parameters:

- Spanning-Tree
- Various ingress and egress forwarding parameters for interfaces (ELTM)
- Interface state (ETHPM)
- VLAN information (Vlan-manager)
- vPC state (vPC manager)

PSS Consistency Checker checks the system state before and after system triggers (switch over, reload, and ISSU). Invoke PSS consistency checker in steady state to avoid false alarms.

Guidelines and Limitations

- Consistency checkers are supported only on M3 and F3 Modules. Only F3 modules are supported in Cisco NX-OS Release 8.0(x), and Cisco NX-OS Release 8.1(x) releases.
- If there is a configuration change or a table state change in the environment while a consistency checker is running, it is possible to trigger false positives. In cases where false positives may be a concern, it is recommended to run multiple iterations of that consistency checker.
- L3-interface consistency checker supports only L3 standalone, L3 port channel IPv4 and IPv6 interfaces, and L3 FEX HIF interfaces. Logical interfaces such as OTV, NVE, and tunnel are not supported.
- Layer 2 multicast consistency checker supports only CE (classical Ethernet) IGMP Snooping entries. VxLAN, OTV, and Fabricpath entries for example, are not supported. Layer 2 multicast consistency checker cannot be used when unsupported features such as Fabricpath/ EVPN) is enabled on a VDC.

Using the Consistency Checker CLIs

To verify the consistency between the hardware and software for the Consistency Checker parameter for Cisco NX-OS Release 8.2(1) uses the following CLIs:

Command	Purpose
show consistency-checker link-state	Verifies the programming consistency between software and hardware for the link-state status of the interfaces.
show consistency-checker interface-properties module *[module number]*	Verifies the interface properties for all modules. Use the *[module]* keyword to verify the properties for a specific module.
show consistency-checker stp-state	Verifies the programming consistency between software and hardware tables for the Spanning-Tree state.
show consistency-checker l2mcast { *vlan ID* } { *group address* | *source address* } [all] [detail]	Verifies the layer-2 multicast consistency for L2 IGMP Snooping entries between supervisor and I/O modules

show consistency-checker l3-interface { *if index* | **bdi** | **ethernet** | **port-channel** }	Verifies the programming consistency between software and hardware for L3-interface ingress and egress forwarding tables
show consistency-checker fabricpath {ftag-state | gpc-membership}	Verifies the ftag CBL state in the software and the hardware and the FabricPath gateway port-channel membership.
show consistency-checker proxy rpc membership	Verifies the proxy router port-channel membership.
show consistency-checker l2unicast *module number*	Verifies consistency for L2 mac address table between supervisor software and I/O module hardware
show consistency-checker pss	Verifies the consistency between run-time data and data stored in PSS for STP, ELTM, ETHPM, VLAN manager, and vPC manager.
show consistency-checker all	Performs all available consistency checkers.

Consistency Checker — Cisco NX-OS Release 8.0(1)

The following sections are applicable for Cisco NX-OS Release 8.0(1).

Consistency Checker is a serviceability tool that performs the following functions:

- Checks for system consistency
- Helps perform root cause analysis and fault isolation
- Checks for consistency between software and hardware tables
- Performs on-demand trigger through CLI or NX-API

Consistency Checker consists of the following components:

- **Ethernet Port Manager (EthPM)**—Provides software values for the following parameters:
  - Link state—Provides software support on Ethernet interfaces, Fabric Extender (FEX) interfaces, and breakout interfaces.
  - Flow control—Provides software support on Ethernet interfaces, FEX interfaces, breakout interfaces, and port-channel interfaces.
  - FEX fabric port or any other port—Provides software support on FEX fabric port or any other port.
  - Native VLAN—Provides software support on L2 Ethernet interfaces, L2 FEX interfaces, L2 breakout interfaces, and L2 port-channel interfaces.
- **Spanning Tree Protocol (STP)**—Checks logical port-state consistency, either port or VLAN. Consistency is checked against STP and PIXM components.

Note Currently, consistency is checked only against the STP internal database based on the software port state and from the response provided by the PIXM on any port-state request.

- **PIXM**—Establishes relationship between the following parameters:

- Port-channel membership between PIXM and port channel
- Gateway port channel (GPC) membership between Private Internet Exchange Manager (PIXM) and Multi Channel Manager (MCM)
- RPC membership between PIXM and MCM
- VLAN CBL membership between STP, PIXM, and HW
- FTAG CBL membership between PIXM and HW

- **L2MCAST**—Verifies Layer 2 multicast (L2MCAST) route consistency across Internet Group Management Protocol (IGMP), Multicast Layer 2 RIB (M2RIB), Multicast FIB (MFIB) Distribution (MFDM), PIXM, and L2MCAST.

Note Currently, L2MCAST supports only Classical Ethernet (CE) mode and not FabricPath.

- **L3 interface properties**—Checks consistency between the contents of various forwarding hardware tables (LDB, ILM, ELM, PVV, and so on) used in L3 interfaces and their expected contents that are stored in ELTM or IFTMC. Consistency is checked on L3 interfaces, L3 port channels, L3 FEX ports, L3 HIF port channels, and L3 interface VLANs.

Output Examples for Consistency Checker Components

Output Examples for Consistency Checker Components – Cisco NX-OS Release 8.2(1)

Example: Show Consistency Checker All Output

```
switch# show consistency-checker all

------------------------------------------------------------
Consistency checker started at 2017 Sep 29 20:54:09 .
Please run 'show consistency-checker all status' to see the status.
------------------------------------------------------------
switch# show consistency-checker all status
------------------------------------------------------------
Consistency checker was started at 2017 Sep 29 20:54:09 .
Consistency checker in progress !
------------------------------------------------------------
switch# show consistency-checker all output
Consistency-checker result:
(VDC: 1 ,TIME: 2017 Sep 29 20:54:09)
------------------------------------------------------------
Consistency Checker Result for Ftag CBL: SUCCESS
------------------------------------------------------------
------------------------------------------------------------
Consistency Checker Result for GPC:  SUCCESS
------------------------------------------------------------
Interface properties checks (Module 2):
NATIVE_VLAN: PASSED
FEX_STATUS: PASSED
SPEED: PASSED
FLOW_CONTROL: PASSED
MTU: PASSED
------------------------------------------------------------
Module 2: PASSED.
```

```
----------------------------------------------------------------
Interface properties checks (Module 4):
NATIVE_VLAN: PASSED
FEX_STATUS: PASSED
SPEED: PASSED
FLOW_CONTROL: PASSED
MTU: PASSED
----------------------------------------------------------------
Module 4: PASSED.
----------------------------------------------------------------
Link State Checks :
----------------------------------------------------------------
Module 2: PASSED
----------------------------------------------------------------
Link State Checks :
----------------------------------------------------------------
Module 4: PASSED
----------------------------------------------------------------
----------------------------------------------------------------
Consistency Checker Result for RPC: SUCCESS
----------------------------------------------------------------
----------------------------------------------------------------
Consistency Checker Result for STP (VLAN CBL): SUCCESS
----------------------------------------------------------------
PSS CONSISTENCY CHECK RESULT FOR IFTMC ON VDC 1 MODULE 2: SUCCESS
================================================================
PSS CONSISTENCY CHECK RESULT FOR IFTMC ON VDC 1 MODULE 4: SUCCESS
================================================================
PSS CONSISTENCY CHECK RESULT FOR ELTM: FAILURE
----------------------------------------------------------------
 ATTRIBUTE NAME     : ELTM INTERFACE PSS
 INCONSISTENT DATA  : intf Vlan4040 (0x9010fc8)
Please collect the tech-support for eltm detail for more details.
================================================================
PSS CONSISTENCY CHECK RESULT FOR ETHPM: SUCCESS
----------------------------------------------------------------
No inconsistency detected in ethpm persistent, runtime and shared data.
================================================================
PSS CONSISTENCY CHECK RESULT FOR STP: SUCCESS
----------------------------------------------------------------
No inconsistency detected in STP CBL data
================================================================
PSS CONSISTENCY CHECK RESULT FOR VLAN_MGR: SUCCESS
----------------------------------------------------------------
No inconsistency detected in vlan_mgr persistent, runtime and shared data.
================================================================
PSS CONSISTENCY CHECK RESULT FOR vPC MGR: SUCCESS
----------------------------------------------------------------
No inconsistency detected in vPC persistent, runtime and shared data.
================================================================

Consistency-checker took 161 secs.
switch#
```

Example: Show Consistency Checker Interface Properties Output

```
switch# show consistency-checker interface-properties

Interface properties checks (Module 4):
NATIVE_VLAN: PASSED
FEX_STATUS: PASSED
SPEED: PASSED
FLOW_CONTROL: PASSED
```

```
MTU: PASSED
------------------------------------------------------------
Module 4: PASSED.
------------------------------------------------------------

switch#
```

Example: Show Consistency Checker Link State Output

```
switch# show consistency-checker link-state


Link State Checks :
------------------------------------------------------------
Module 4: PASSED
------------------------------------------------------------
switch#
```

Example: Show Consistency Checker L2Unicast Output

```
switch# show consistency-checker l2unicast 1
  Consistency Checker Status: Success


switch# show consistency-checker l2unicast 1
Missing entries in the MAC Table
   VLAN     MAC Address      Type      age     Secure NTFY    Ports
---------+-----------------+--------+---------+------+----+------------------
1201     64a0.e741.2bc1    dynamic     ~~~      F    F  Po100
Extra entries in the MAC Table
   VLAN     MAC Address      Type      age     Secure NTFY    Ports
---------+-----------------+--------+---------+------+----+------------------
  1201     64a0.e741.2bc1    dynamic     ~~~      F    F  Po100
  1202     64a0.e741.2bc1    dynamic     ~~~      F    F  Po100
Discrepant entries in the MAC Table
   VLAN     MAC Address      Type      age     Secure NTFY    Ports
---------+-----------------+--------+---------+------+----+------------------
* 2913     0000.3f80.a6e2    static       -       T    T  Eth153/1/17
* 2914     0000.3f80.a6e4    static       -       T    T  Eth153/1/18
* 2915     0000.3f80.a6e6    static       -       T    T  Eth15

Consistency-Checker: Failure
```

Example: Show Consistency Checker L2Multicast Output

```
switch# show consistency-checker l2mcast all

  Module 10 : Success
  Module 1 : Success
  Module 3 : Success
  Module 2 : Success
  Module 4 : Not Supported
  Module 7 : Not Supported
  Module 9 : Success
  Module 8 : Success
  Consistency Checker Status: Success
```

Example: Show Consistency Checker Spanning-Tree Output

```
switch# show consistency-checker stp-state
------------------------------------------------------------
Consistency Checker Result for STP (VLAN CBL): SUCCESS
------------------------------------------------------------
```

```
switch# show consistency-checker stp-state
Consistency Checker Result for STP (VLAN CBL): FAILED
STP/HW VLAN CBL mismatch (port Eth8/3):
INGRESS FORWARDING: (STP) 1-10, (HW) 1-10,30-35
EGRESS FORWARDING: (STP) 1-10, (HW) 1-10,30-35
PIXM/HW VLAN CBL mismatch (port Eth8/3):
INGRESS FORWARDING: (PIXM) 1-10, (HW) 1-10,30-35
EGRESS FORWARDING: (PIXM) 1-10, (HW) 1-10,30-35
```

Example: Show Consistency Checker PSS Output

```
switch# show consistency-checker pss
PSS CONSISTENCY CHECK RESULT FOR IFTMC ON VDC 1 MODULE 2: SUCCESS
============================================================
PSS CONSISTENCY CHECK RESULT FOR IFTMC ON VDC 1 MODULE 4: SUCCESS
============================================================
PSS CONSISTENCY CHECK RESULT FOR ELTM: FAILURE
------------------------------------------------------------
 ATTRIBUTE NAME     : ELTM INTERFACE PSS
 INCONSISTENT DATA  : intf Vlan4040 (0x9010fc8)
 ATTRIBUTE NAME     : ELTM INTERFACE PSS
 INCONSISTENT DATA  : intf port-channel200 (0x160000c7)
Please collect the tech-support for eltm detail for more details.
============================================================
PSS CONSISTENCY CHECK RESULT FOR ETHPM: SUCCESS
------------------------------------------------------------
No inconsistency detected in ethpm persistent, runtime and shared data.
============================================================
PSS CONSISTENCY CHECK RESULT FOR STP: SUCCESS
------------------------------------------------------------
No inconsistency detected in STP CBL data
============================================================
PSS CONSISTENCY CHECK RESULT FOR VLAN_MGR: SUCCESS
------------------------------------------------------------
No inconsistency detected in vlan_mgr persistent, runtime and shared data.
============================================================
PSS CONSISTENCY CHECK RESULT FOR vPC MGR: SUCCESS
------------------------------------------------------------
No inconsistency detected in vPC persistent, runtime and shared data.
============================================================
```

Example: Show Consistency Checker PSS Output

```
switch# show consistency-checker l3-interface port-channel 5
Consistency Checker Result for Interface: port-channel5  :  Success
```

```
switch# show consistency-checker l3-interface port-channel 5
Consistency Checker Result for Interface: port-channel5  :  Failure
Total Errors Found       :  1
Found error on slot 9 Intf: port-channel5 (0x16000004) : SDB error(1)
Errors detected.  Please collect the output of 'show tech-support eltm detail'.
```

Example: Show Consistency Checker FabricPath Output

```
switch# show consistency-checker fabricpath gpc-membership
Consistency Checker Result for GPC:  SUCCESS
```

```
switch# show consistency-checker fabricpath gpc-membership
Consistency Checker Result for GPC : FAILED
    gpc1:1005 not found in PIXM DB
    gpc1:1008 not found in PIXM DB
```

Example: Show Consistency Checker Proxy RPC Output

```
switch# show consistency-checker proxy vl3-membership
Consistency Checker Result for Proxy VL3: SUCCESS
```

```
switch# show consistency-checker proxy vl3-membership
Consistency Checker Result for Proxy VL3: FAILED
MCM VL3 members: Eth1/3  Eth1/4
PIXM VL3 members: Eth1/3
```

Output Examples for Consistency Checker Components – Cisco NX-OS Release 8.0(1)

Example: Link State Output

This example shows a link state output:

```
switch# show consistency-checker link-state
Link State Checks:
Consistency Check: FAILED
Inconsistencies found for following interfaces:
Ethernet1/12 hw_link_state(0) sw_link_state(1)
```

Example: STP Output

This example shows an STP output when the Consistency Checker result for STP passed:

```
switch# show consistency-checker stp-state
Consistency Checker Result for STP (VLAN CBL): SUCCESS
```

This example shows an STP output when the Consistency Checker result for STP failed:

```
switch# show consistency-checker stp-state
Consistency Checker Result for STP (VLAN CBL): FAILED

STP/HW VLAN CBL mismatch (port Eth8/3):
INGRESS FORWARDING: (STP) 1-10, (HW) 1-10,30-35
EGRESS FORWARDING: (STP) 1-10, (HW) 1-10,30-35
PIXM/HW VLAN CBL mismatch (port Eth8/3):
INGRESS FORWARDING: (PIXM) 1-10, (HW) 1-10,30-35
EGRESS FORWARDING: (PIXM) 1-10, (HW) 1-10,30-35

Please collect the output of 'show tech-support spanning-tree'.
```

Example: PIXM (FabricPath) Output

This example shows a PIXM output when the Consistency Checker result for PIXM passed:

```
switch# show consistency-checker fabricpath ftag-state
Consistency Checker Result for Ftag CBL: SUCCESS


switch# show consistency-checker fabricpath gpc-membership
Consistency Checker Result for GPC: SUCCESS
```

These examples show PIXM outputs when the Consistency Checker result for PIXM failed:

```
switch# show consistency-checker fabricpath ftag-state
Consistency Checker Result for Ftag CBL: FAILED
PIXM/HW FTag CBL mismatch (port Eth3/9):
    INGRESS FORWARDING: (PIXM) 1-2, (HW) 1-2,30-35
    EGRESS FORWARDING: (PIXM) 1-2, (HW) 1-2,30-35


switch# show consistency-checker fabricpath gpc-membership
Consistency Checker Result for GPC : FAILED
  gpc3:22
  PIXM members:   Eth2/2
  MCM members:  Eth2/2 Eth2/3
```

```
switch# show consistency-checker proxy rpc-membership
Consistency Checker Result for RPC: FAILED
PIXM vl3 members: Eth4/3
MCM vl3 members:  Eth4/1 Eth4/10 Eth4/17 Eth4/18 Eth4/2 Eth4/25   Eth4/26 Eth4/9 Eth9/1
Eth9/10 Eth9/17 Eth9/18 Eth9/2 Eth9/25 Eth9/26 Eth9/9
```

Example: L2MCAST Output

This example shows a L2MCAST output when the Consistency Checker result for L2MCAST passed:

```
switch(config)# show consistency-checker l2mcast 500 239.2.3.5
Consistency Checker Status: Passed
```

These examples show L2MCAST outputs when the Consistency Checker result for L2MCAST failed:

```
switch(config)# show consistency-checker l2mcast 500 239.2.3.5
Consistency Checker Status: Failed
Inconsistency found in Layer 2 Multicast NextHop
Detailed logs can be found with "show consistency-checker l2mcast vlan group [source]” with
 detail keyword.
```

```
switch(config)# show consistency-checker l2mcast 500 239.2.3.5 detail
Consistency Checker Status: Failed

-------------------------------------------
Route: ('500', '10.120.33.63', '239.2.3.5')
-------------------------------------------
B - Baseline
C - Route and Next-Hop Consistent
I - Next-Hop Inconsistent
M - Missing Route
IGMP: ( B ) set([u'Eth7/9/3'])
M2RIB: ( C ) set([u'Eth7/9/3'])
MFDM: ( C ) 0x7be4
PIXM: ( I ) set(['Eth7/9/3’, ‘Eth7/9/2’])
```

Example: Interface Properties Output

This example shows an interface properties output:

```
switch# show consistency-checker interface-properties
Interface properties checks :
Consistency Check (native_vlan) : PASSED
Consistency Check (fex_status) : PASSED
Consistency Check (speed) : FAILED
Inconsistencies found for following interfaces:
Ethernet1/12 hw_speed(10000) sw_speed(1000)
Consistency Check (flow_control) : PASSED
Please collect the output of 'show tech-support ethpm’
```

Example: L3 Interface Properties Output

This example shows an L3 interface properties output when the Consistency Checker result for L3 interface passed:

```
switch# show consistency-checker l3-interface ethernet 3/6
Consistency Checker Result for Interface:Ethernet3/6 : Success
```

This example shows an L3 interface properties output when the Consistency Checker result for L3 interface failed:

```
switch# show consistency-checker l3-interface ethernet 3/6
Consistency Checker Result for Interface:Ethernet3/6 : Failure
Total Errors Found : 1
Found error on slot 3 Intf:Ethernet3/6 (0x1a105000) : ELM error(19)
Errors detected. Please collect the output of 'show tech-support eltm detail'.
```

Fault Management System

The Fault Management System is used to enhance Cisco NX-OS serviceability by providing an efficient means to capture data that is relevant and adequate to debug the issues being reported at the earliest possible time, without any manual intervention. If all the nodes are down, the packets get routed automatically.

The Fault Management System provides two main benefits in enhancing Cisco NX-OS serviceability:

- **Trigger-based auto capture**—The Fault Management System provides a set of programmable hooks that can be inserted at various predefined (failure) points in such a way that the relevant data is captured automatically whenever a trigger is detected. The data collected by this system includes ASCII tech support, binary tech support, global message and transaction service data, various process-specific details, and specific **show** commands. This system is designed to capture data in the least intrusive way possible.
- **Message and transaction service statistics**—The Fault Management System provides an extension to the message and transaction service infrastructure (mtstrack) library that collects per-process and global message and transaction service statistics. The statistical results can be displayed and analyzed, as required. Message and transaction service statistics (mtstrack) feature is incorporated with the Auto Capture feature to work as an Auto Capture trigger. Using the Auto Capture trigger, any message and transaction service leak in the system can be detected and the **show tech-support** command output can be captured automatically. As with the message and transaction service statistics Auto Capture trigger, trigger points can be identified on other infra components and auto triggers can be added.

Programmability in the Fault Management System

This feature provides a flexible infra and provides functionalities to tweak the behavior of the system to meet the requirements of every Cisco NX-OS process.

The behavior of the system can be programmed using a YAML file. A system default YAML file is present; this can be overwritten with a custom YAML file. When a custom YAML file is used, programming is performed incrementally over the system YAML file.

Note The custom YAML file name must be *fault-mgmt.yaml* in order to enable the file to overwrite the existing YAML file.

This example shows the contents of a YAML file:

```
applications:
 vlan:
         ts_name: vlan
         group_ts_name: "private-vlan,ethpm"
         max_msg_timeout: 30
 ethpm:
        ts_name: ethpm
        group_ts_name: "vlan,lim"
```

```
       max_msg_timeout: 30
       auto_trigger_disable_eve_seq_failure: 1
"private-vlan":
       ts_name: "private-vlan"
       group_ts_name: "ethpm,vlan,stp"
       max_msg_timeout: 30
"eltm detail":
       ts_name: "eltm detail"
       group_ts_name: "vlan,vni"
       max_msg_timeout: 30
"vpc":
      max_msg_timeout: 30
      auto_trigger_disable_eve_seq_failure: 1
```

The following table provides information about semantics used in the YAML file:

Table 1: YAML Semantics

Component	Description
ts_name	Specifies the technical support name for the given application.
group_ts_name	Specifies the names of the applications in the group of a given application.
auto_trigger_disable_mts_timeout	Disables message and transaction service leak detection.
max_msg_timeout	Specifies the message and transaction service leak detection time, in minutes.
auto_trigger_disable_eve_seq_failure	Disables auto trigger on event sequence failure.
auto_trigger_syslog_severity: *severity level*	Specifies syslog severity for the auto capture trigger. Severity level range is from 1 to 7. We do not recommend a severity level above 3.

Adding a Custom YAML File

Procedure

Step 1 Place the YAML file in the **bootflash:scripts/** directory.

Step 2 Use the **fault-management yaml reconfigure** command to overwrite the default YAML file.

Note The custom YAML file name must be *fault-mgmt.yaml* in order to enable the file to overwrite the existing YAML file.

Configuring the Auto Capture Feature

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **fault-management auto-capture**	**Note** The Auto Capture feature is enabled by default. If the Auto Capture feature is disabled, use this command to enable the feature. Use the following information to perform additional configurations in the Auto Capture feature: • Use the [**no**] **fault-management auto-capture** command to disable this feature. • Use the **dir bootflash:fault-management-logs/** command to list the auto captured files. • Use the **clear fault-management logs** [**active** | **standby** | **all**] command to clear the auto captured files.

Configuring the MTS Statistics Feature

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **system statistics mts sap** *sap-number* | **all** [**module** *module-number*]	Enables the Message and Transaction Service Statistics feature. **Note** The Message and Transaction Service Statistics feature is enabled by default. Use the following commands to perform additionally configurations in the Message and Transaction Service Statistics feature: • Use the [**no**] **system statistics mts sap** *sap-number* | **all** [**module** *module-number*] command to disable this feature.

	Command or Action	Purpose
		• Use the **show system statistics mts sap** {*sap-number* | **all**} {**brief** | **module** | **receive** | **transmit**} {**us** | **ms** | **detail**} [**sort** {**ascending** | **descending**} **by** {**last-time** | **max-time** | **avg-time** | **count**} command to display Message and Transaction Service Statistics. **Caution** We recommended that you do not use the **all** keyword for service access points (SAPs) because it retrieves data from all the components, which may, in turn results in a long output. Instead, use the *sap-num* argument to retrieve data from a specific component. • Use the **clear statistics mts sap** {**all** | *sap-number*} [**module** *module-number*] command to reset the Message and Transaction Service Statistics.

Configuration Examples for Fault Management System

Example: Enabling the Auto Capture Feature

This example shows how to enable the Auto Capture feature:

```
switch# configure terminal
switch(config)# fault-management auto-capture
```

Example: Enabling the Message and Transaction Service Statistics Feature

This example shows how to enable the Message and Transaction Service Statistics feature:

```
switch# configure terminal
switch(config)# system statistics mts sap all
```

Example: Clearing the Fault-Management Logs

This example shows how to clear the fault-management logs:

```
switch# configure terminal
switch(config)# clear fault-management logs all
```

Example: Programming the System YAML File

This example shows how to program the system YAML file incrementally:

```
switch# configure terminal
switch(config)# fault-management yaml reconfigure
```

Manageability

This section describes the manageability features in the Cisco NX-OS software.

Simple Network Management Protocol

The Cisco NX-OS software is compliant with Simple Network Management Protocol (SNMP) version 1, version 2, and version 3. A large number of MIBs is supported. For more information about SNMP, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 6.x*.

Configuration Verification and Rollback

The Cisco NX-OS software allows you to verify the consistency of a configuration and the availability of necessary hardware resources prior to committing the configuration. You can preconfigure a device and apply the verified configuration at a later time. Configurations also include checkpoints that allow you to roll back to a known good configuration as needed. For more information about rollbacks, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 6.x*.

Role-Based Access Control

With role-based access control (RBAC), you can limit access to device operations by assigning roles to users. You can customize access and restrict it to the users who require it. For more information about RBAC, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 6.x*.

Connectivity Management Processor

The Cisco NX-OS software supports the use of a Connectivity Management Processor (CMP) for remote platform management. The CMP provides an out-of-band access channel to the Cisco NX-OS console. For more information about CMP, see the *Cisco Nexus 7000 Series Connectivity Management Processor Configuration Guide*.

Cisco NX-OS Device Configuration Methods

You can configure devices using the CLI from a Secure Shell (SSH) session or a Telnet session. SSH provides a secure connection to the device. The CLI configuration guides and command references are organized by feature. For more information, see the Cisco NX-OS configuration guides and the Cisco NX-OS command references. For more information on SSH and Telnet, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 6.x*.

You can also configure devices using the XML management interface, which is a programmatic method based on the NETCONF protocol that complements the CLI. For more information, see the *Cisco NX-OS XML Interface User Guide* .

Traffic Routing, Forwarding, and Management

This section describes the traffic routing, forwarding, and management features supported by the Cisco NX-OS software.

Ethernet Switching

The Cisco NX-OS software supports high-density, high-performance Ethernet systems and provides the following Ethernet switching features:

- IEEE 802.1D-2004 Rapid and Multiple Spanning Tree Protocols (802.1w and 802.1s)
- IEEE 802.1Q VLANs and trunks
- 16,000-subscriber VLANs
- IEEE 802.3ad link aggregation
- Private VLANs
- Cross-chassis private VLANs
- Unidirectional Link Detection (UDLD) in aggressive and standard modes

For more information, see the *Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 6.x* and the *Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide*.

IP Routing

The Cisco NX-OS software supports IP version 4 (IPv4) and IP version 6 (IPv6) and the following routing protocols:

- Open Shortest Path First (OSPF) Protocol Versions 2 (IPv4) and 3 (IPv6)
- Intermediate System-to-Intermediate System (IS-IS) Protocol
- Border Gateway Protocol (BGP)
- Enhanced Interior Gateway Routing Protocol (EIGRP)
- Routing Information Protocol Version 2 (RIPv2)

The Cisco NX-OS software implementations of these protocols are fully compliant with the latest standards and include 4-byte autonomous system numbers (ASNs) and incremental shortest path first (SPF). All unicast protocols support Non-Stop Forwarding Graceful Restart (NSF-GR). All protocols support all interface types, including Ethernet interfaces, VLAN interfaces, subinterfaces, port channels, tunnel interfaces, and loopback interfaces.

For more information, see the *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 6.x*.

IP Services

The following IP services are available in the Cisco NX-OS software:

- Virtual Routing and Forwarding (VRF)
- Dynamic Host Configuration Protocol (DHCP) Helper
- Hot-Standby Routing Protocol (HSRP)
- Gateway Load Balancing Protocol (GLBP)
- Enhanced Object Tracking
- Policy-Based Routing (PBR)
- Unicast Graceful Restart for all protocols in IPv4 Unicast Graceful Restart for OPSFv3 in IPv6

For more information, see the *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 6.x*.

IP Multicast

The Cisco NX-OS software includes the following multicast protocols and functions:

- Protocol Independent Multicast (PIM) Version 2 (PIMv2)
- Source Specific Multicast (SSM)
- PIM sparse mode (Any-Source Multicast [ASM] for IPv4 and IPv6)

Note The Cisco NX-OS software does not support PIM dense mode.

- Bidirectional Protocol Independent Multicast (Bidir PIM)
- Anycast rendezvous point (Anycast-RP)
- Multicast NSF for IPv4 and IPv6
- RP-Discovery using bootstrap router (BSR) (Auto-RP and static)
- Internet Group Management Protocol (IGMP) Versions 1, 2, and 3 router role
- IGMPv2 host mode
- IGMP snooping
- Multicast Listener Discovery (MLD) Protocol Version 2 (for IPv6)
- Multicast Source Discovery Protocol (MSDP) (for IPv4 only)

For more information, see the *Cisco Nexus 7000 Series NX-OS Multicast Routing Command Reference*.

Quality of Service

The Cisco NX-OS software supports quality of service (QoS) functions for classification, marking, queuing, policing, and scheduling. Modular QoS CLI (MQC) supports all QoS features. You can use MQC to provide uniform configurations across various Cisco platforms. For more information, see the *Cisco Nexus 7000 Series NX-OS Quality of Service Configuration Guide, Release 6.x*.

Network Security

This section describes the network security features support by the Cisco NX-OS software.

Cisco TrustSec

Cisco TrustSec security provides data confidentiality and integrity and supports standard IEEE 802.1AE link-layer cryptography with 128-bit Advanced Encryption Standard (AES) cryptography. Link-layer cryptography guarantees end-to-end data privacy while allowing the insertion of security service devices along the encrypted path. Cisco TrustSec uses security group access control lists (SGACLs), which are based on security group tags instead of IP addresses. SGACLs enable policies that are more concise and easier to manage due to their topology independence. For more information, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 6.x*.

Additional Network Security Features

In addition to Cisco TrustSec, the Cisco NX-OS software includes the following security features:

- Data path intrusion detection system (IDS) for protocol conformance checks
- Control Plane Policing (CoPP)
- Message-digest algorithm 5 (MD5) routing protocol authentication
- Cisco-integrated security features, including Dynamic Address Resolution Protocol (ARP) inspection (DAI), DHCP snooping, and IP Source Guard
- Authentication, authorization, and accounting (AAA)
- RADIUS and TACACS+
- SSH Protocol Version 2
- SNMPv3
- Port security
- IEEE 802.1X authentication
- Layer 2 Cisco Network Admission Control (NAC) LAN port IP
- Policies based on MAC and IPv4 addresses supported by named ACLs (port-based ACLs [PACLs], VLAN-based ACLs [VACLs], and router-based ACLs [RACLs])
- Traffic storm control (unicast, multicast, and broadcast)

- Unicast Reverse Path Forwarding (Unicast RPF)

For more information, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 6.x*.

Licensing

The Cisco NX-OS software licensing feature allows you to access premium features on the device after you install the appropriate license for that feature. Any feature not included in a license package is bundled with the Cisco NX-OS software and is provided to you at no extra charge.

You must purchase and install a license for each device.

Note With the exception of the Cisco TrustSec feature, you can enable a feature without installing its license. The Cisco NX-OS software gives you a grace period that allows you to try a feature before purchasing its license. You must install the Advanced Services license package to enable the Cisco TrustSec feature.

For detailed information about Cisco NX-OS software licensing, see the *Cisco NX-OS Licensing Guide*.

For information about troubleshooting licensing issues, see the Cisco Nexus 7000 Series NX-OS Troubleshooting Guide.

Supported Standards

This table lists the IEEE compliance standards.

Table 2: IEEE Compliance Standards

Standard	Description
802.1D	MAC Bridges
802.1s	Multiple Spanning Tree Protocol
802.1w	Rapid Spanning Tree Protocol
802.1AE	MAC Security (link layer cryptography)
802.3ad	Link aggregation with LACP
802.3ab	1000BASE-T (10/100/1000 Ethernet over copper)
802.3ae	10-Gigabit Ethernet
802.1Q	VLAN Tagging
802.1p	Class of Service Tagging for Ethernet frames
802.1X	Port-based network access control

This table lists the RFC compliance standards.

Table 3: RFC Compliance Standards

Standard	Description
BGP	
RFC 1997	BGP Communities Attribute
RFC 2385	Protection of BGP Sessions via the TCP MD5 Signature Option
RFC 2439	BGP Route flap damping
RFC 2519	A Framework for Inter-Domain Route Aggregation
RFC 2858	Multiprotocol Extensions for BGP-4
RFC 3065	Autonomous System Confederations for BGP
RFC 3392	Capabilities Advertisement with BGP-4
RFC 4271	BGP version 4
RFC 4273	BGP4 MIB - Definitions of Managed Objects for BGP-4
RFC 4456	BGP Route reflection
RFC 4486	Subcodes for BGP cease notification message
RFC 4724	Graceful Restart Mechanism for BGP
RFC 4893	BGP Support for Four-octet AS Number Space
ietf-draft	Bestpath transition avoidance (draft-ietf-idr-avoid-transition-05.txt)
ietf-draft	Peer table objects (draft-ietf-idr-bgp4-mib-15.txt)
ietf-draft	Dynamic Capability (draft-ietf-idr-dynamic-cap-03.txt)
OSPF	
RFC 2370	OSPF Opaque LSA Option

Standard	Description
RFC 2328	OSPF Version 2
RFC 2740	OSPF for IPv6 (OSPF version 3)
RFC 3101	OSPF Not-So-Stubby-Area (NSSA) Option
RFC 3137	OSPF Stub Router Advertisement
RFC 3509	Alternative Implementations of OSPF Area Border Routers
RFC 3623	Graceful OSPF Restart
RFC 4750	OSPF Version 2 MIB
RIP	
RFC 1724	RIPv2 MIB extension
RFC 2082	RIPv2 MD5 Authentication
RFC 2453	RIP Version 2
IS-IS	
RFC 1142 (OSI 10589)	OSI 10589 Intermediate system to intermediate system intra-domain routing exchange protocol
RFC 1195	Use of OSI IS-IS for routing in TCP/IP and dual environment
RFC 2763	Dynamic Hostname Exchange Mechanism for IS-IS
RFC 2966	Domain-wide Prefix Distribution with Two-Level IS-IS
RFC 2973	IS-IS Mesh Groups
RFC 3277	IS-IS Transient Blackhole Avoidance
RFC 3373	Three-Way Handshake for IS-IS Point-to-Point Adjacencies
RFC 3567	IS-IS Cryptographic Authentication
RFC 3847	Restart Signaling for IS-IS

Standard	Description
ietf-draft	Internet Draft Point-to-point operation over LAN in link-state routing protocols (draft-ietf-isis-igp-p2p-over-lan-06.txt)
IP Services	
RFC 768	UDP
RFC 783	TFTP
RFC 791	IP
RFC 792	ICMP
RFC 793	TCP
RFC 826	ARP
RFC 854	Telnet
RFC 959	FTP
RFC 1027	Proxy ARP
RFC 1305	NTP v3
RFC 1519	CIDR
RFC 1542	BootP relay
RFC 1591	DNS client
RFC 1812	IPv4 routers
RFC 2131	DHCP Helper
RFC 2338	VRRP
RFC 2784	Generic Routing Encapsulation (GRE)
IP-Multicast	
RFC 2236	Internet Group Management Protocol, Version 2
RFC 2710	Multicast Listener Discovery (MLD) for IPv6
RFC 3376	Internet Group Management Protocol, Version 3

Standard	Description
RFC 3446	Anycast Rendezvous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)
RFC 3569	An Overview of Source-Specific Multicast (SSM)
RFC 3618	Multicast Source Discovery Protocol (MSDP)
RFC 3810	Multicast Listener Discovery Version 2 (MLDv2) for IPv6
RFC 4601	ASM - Sparse Mode (PIM-SM): Protocol Specification (Revised)
RFC 4607	Source-Specific Multicast for IP
RFC 4610	Anycast-RP Using Protocol Independent Multicast (PIM)
ietf-draft	Mtrace server functionality, to process mtrace-requests, draft-ietf-idmr-traceroute-ipm-07.txt
ietf-draft	Bi-directional Protocol Independent Multicast (BIDIR-PIM), draft-ietf-pim-bidir-09.txt