# Catalyst 6500 Series with Cisco IOS Software Modularity

## Functionality and Feature Verification Tests

## Introduction

At the end of August 2005, Cisco Systems announced Cisco IOS Software Modularity for the Catalyst 6500 Series switches.

Continuous forwarding of mission-critical traffic is becoming increasingly important in triple-play (voice, video, and data) networks and for new interactive user services. In addition to high availability designs which include redundant systems, the highest level of software resiliency is required everywhere in the network. From the core and distribution to single points of failure such as the wiring closet, data center access and WAN edge of an enterprise network, software resilience allows strict end-to-end service level agreements to be delivered.

Cisco IOS Software Modularity for the Catalyst 6500 exactly addresses these high availability requirements. By enabling modular Cisco IOS subsystems to run in independent processes, unplanned downtime is minimized through self-healing processes, software changes are simplified through subsystem In-Service Software Upgrades (ISSU) and automated process-level policy control is enabled by integrating Embedded Event Manager (EEM).

Tested by EANTC 2005

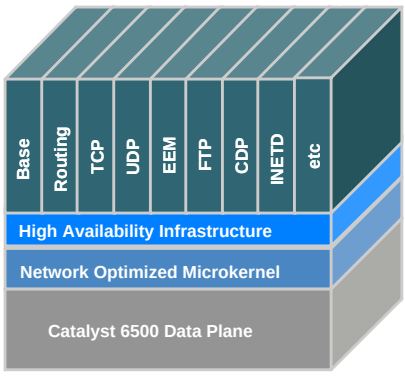
**EANTC extensively tested the first shipment of the Cisco IOS Software Modularity enhancements to the 12.2SX release for the Catalyst 6500 Series switches. Different maintenance actions from migrating to the new Cisco IOS 12.2SX release to real life operating tasks were analyzed for ease of use and double-checked against the high availability claims of Cisco. We found a system consistent in all areas tested with the previous 12.2SX IOS version and did not see, any packet loss during our tests thereby validating Cisco's claims for this new technology on the Catalyst 6500 Series.**

**Test Highlights**

- **Migration to a new 12.2SX release with Cisco IOS Software Modularity went smoothly and guaranteed operational consistency (CLI, configurations and »look & feel«) with the previous IOS 12.2SX release**
- **New »installer« feature to simplify image and patch management was easy to use**
- **Fault transparency with zero packet loss was observed due to the stateful process restart feature with memory protection and fault containment**
- **No service disruption with subsystem In-Service Software Upgrades (ISSU) and Non-Stop Forwarding (NSF)**
- **Zero packet loss while applying patches to address recreated vulnerabilities**
- **Intuitive »patch management« process that includes tagging, patch rollbacks and repackaging of patched images**
- **Easy to use multi-purpose Embedded Event Manger (EEM) features for diagnostics and automated policies based on TCL scripting**

## Venue & Test Equipment

Cisco commissioned the European Advanced Networking Test Center (EANTC) to verify the migration path and functionalities of the new IOS release offering Software Modularity. The tests were conducted at Cisco's test lab in San Jose, California in August 2005. EANTC test engineers performed all tests to verify the expected feature functionality and in parallel

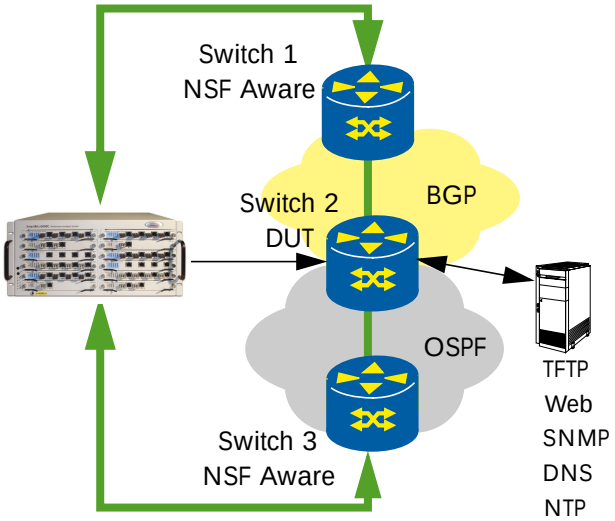



**Catalyst 6500 with Cisco IOS Sofware Modularity**

continuous control and data plane traffic forwarding. The testing environment included Spirent's SmartBits load generator in combination with TeraRouting Tester software.

## General Test Setup and Methodology

The test bed, seen in the following graph, consisted of three Cisco Catalyst 6504-E chassis. One was our DUT (device under test) and the other two 6500s were used to provide NSF (Non-Stop Forwarding) awareness in the test bed. At the start of the test all three devices ran Cisco IOS Version 12.2(18)SXE2 which is generally available on Cisco Connection Online (CCO). During the test, the DUT was upgraded to a pre-FCS (First Customer Shipment) release of 12.2(18)SXF1, the first release for the Catalyst 6500 series planned to offer Cisco IOS Software Modularity.




**Logical Test Setup**

Switch 1 represented the service provider edge, receiving 180,000 E-BGP routes from the SmartBits load generator. Switch 3 represented an enterprise Catalyst 6500 switch and was fed by the SmartBits with 10,000 OSPF routes. The DUT (Switch 2), configured between these two switches and set up for BGP and for OSPF, held a total of 190,000 routes. By setting up full-meshed traffic between the announced routes, the test bed forwarded a total of 3.6 million different flows. Two GigabitEthernet ports were used and a load of 95% was configured bi-directionally to allow for the exchange of control plane traffic, resulting in the DUT forwarding in total 2.828 million 64 Bytes frames per second in each direction. These reference streams showed no packet loss and 21.2 µs average end-to-end latency.

The detailed configuration of the test bed was:

Switch1:

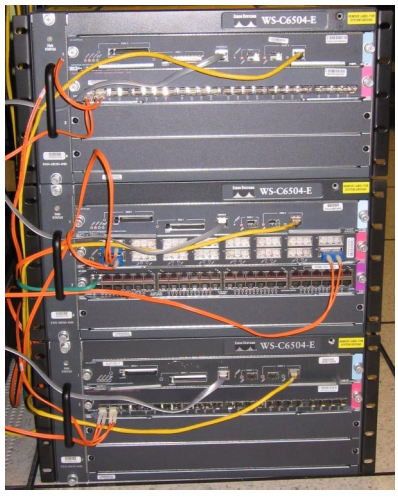
- Supervisor Engine 720 with Policy Feature Card (PFC) 3B
- 24 port 1GbE SFP, WS-X6724-SFP, with a Centralized Forwarding Card, WS-F6700-CFC

Switch 2:

- Supervisor Engine 720 with Policy Feature Card (PFC) 3BXL
- 16 port 1GbE GBIC, WS-X6516A-GBIC, and a 48 port 10/100 Mb RJ45, WS-X6348-RJ-45

Switch 3:

- Supervisor Engine 720 with Policy Feature Card (PFC) 3A
- 24 port 1GbE SFP, WS-X6724-SFP, with a Centralized Forwarding Card, WS-F6700-CFC



**Physical Test Setup in the Cisco Labs**

## 1) Migration to IOS with Software Modularity

**Test Highlights**

- Seamless migration to the new 12.2SX release with Cisco IOS Software Modularity
- Consistency with previous Cisco IOS versions (CLI, configurations and »look & feel«)
- Easy to use »installer« feature

### Test Objective

The aim of this test was

- to show the seamless migration path from a previous release to a new Cisco IOS release with Software Modularity,
- to verify CLI (command line interface) and configuration consistency of the new software release,
- to present the »installer« feature, which is used for new operations of Cisco IOS with Software Modularity and
- to show the new Cisco IOS enhancements in the form of modularized run-time processes.

### Test Methodology

We started this test by downloading the new software via FTP to the Supervisor Engine of the DUT. After reconfiguring the bootstring to point to the new image, the DUT was reloaded to run the new Cisco IOS in the well known 'binary' mode, which is a single image that is not expanded into a file system. At this time, a first verification of the DUT configuration and CLI consistency was done.

In order to be able to apply patches to software subsystems, we then changed from the »binary« to the »installed« mode configuration. In the »installed« mode, the system runs from a file system, either »boot-disk:« or »disk0/1:«. By using some »show« commands (e.g. »show install running« or »show install disk0:/sys«) we verified the new installation environment.

```
switch-2(DUT)#sh install running
Software running on card installed at location
s72033_rp - Slot 1 :

B/P C State     Filename
--- - --------  --------
 B    Active    disk0:/sys/s72033_rp/base/
DRACO2_MP
Software running on card installed at location
s72033 - Slot 1 :
B/P C State     Filename
--- - --------  --------
 B    Active    disk0:/sys/s72033/base/s72033-
adventerprisek9-vm(12.2(20050810:214544))


LEGEND:
-------:
B/P' - (B)ase image or (P)atch
'C' - (C)ommitted
Pruned - This file has been pruned from the system
Active - This file is active in the system
PendInst - This file is set to be made available to
run on the system after next activation.
PendRoll - This file is set to be rolled back after
next activation.
InstPRel - This file will run on the system after
next reload
RollPRel - This file will be removed from the
system after next reload
RPRPndIn - This file is both rolled back pending a
reload, and pending installation.  On reload, this
file will not run and will move to PendInst state.
If 'install activate' is done before reload,
pending removal and install cancel each other and
file simply remains active
IPRPndRo - This file is both installed pending a
reload, and pending rollback. If the card reloads,
it will be active on the system pending a roll-
back. If 'install activate' is done before a
reload, the pending install and removal with
cancel each other and the file will simply be
removed
```

To analyze the new process architecture, we issued commands which were enhanced compared to the previous Cisco IOS release, like »show process« and »show process cpu«, and we observed more than 20 modular processes.

During the different stages of these first tests, like downloading or installing the OS, we verified the packet forwarding performance to show zero packet loss and no increase in latency.

### EANTC Test Analysis

As expected, only a few small steps were necessary to migrate to the new 12.2SX release with Cisco IOS Software Modularity on the Catalyst 6500. The tested Cisco IOS software came up with the same look & feel (CLI) as the previous Cisco IOS version and the switch configuration remained exactly the same during this change (apart from the boot string). With the exception of the system restart to load the new image, no packet loss or increased latency could be observed.

## 2) Restarting Processes and Showing NSF Functionality

**Test Highlights**

- No out-of-service times during a routing process restart with memory protection, fault containment and transparency, and Non-Stop-Forwarding

### Test Objective

The aim of this test was to show that modular processes like the routing process, covering BGP and all IGPs such as OSPF, can be restarted without packet loss in the data plane.

### Test Methodology

As a precondition of this test, the BGP and OSPF neighbors of the DUT had to be aware of the NSF protocol, avoiding route flaps due to the failing adjacency.

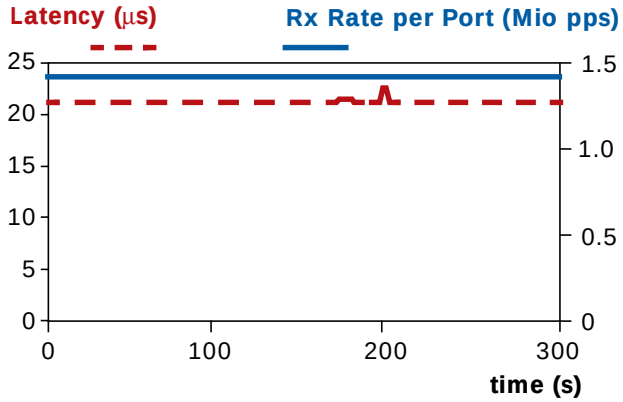
Before and after the manual restart of the routing process »iprouting.iosproc«, we observed the detailed process status with the »show process detail iprouting.iosproc«. In parallel to the process restart, we sent the reference test traffic to observe correct NSF functionality.

```
switch-2(DUT)#sh processes detailed
iprouting.iosproc
                   Job Id: 69
                      PID: 24619
         Executable name: iprouting.iosproc
         Executable Path: sbin/iprouting.iosproc
             Instance ID: 1
                 Respawn: ON
           Respawn count: 1
Respawn since last patch: 1
  Max. spawns per minute: 30
            Last started: Thu Aug 18 17:11:32 2005
           Process state: Run
            Feature name: iprouting
                    Core: SHAREDMEM MAINMEM
                Max. core: 0
                   Level: 100
               Mandatory: ON
    Last restart userid:
      Related Processes:

PID    TID  Stack pri state       Blked
HR:MM:SS:MSEC FLAGS    NAME
24619  1      32K  10 Receive     1
0:00:13:0248 00000000 iprouting.iospro
24619  2      32K  10 Receive     1
0:00:00:0000 00000000 iprouting.iospro
24619  3      32K  10 Receive     1
0:00:26:0344 00000000 iprouting.iospro
24619  4      32K  11 Nanosleep
0:00:00:0000 00000000 iprouting.iospro
24619  5      32K  10 Receive     1
0:00:35:0248 00000000 iprouting.iospro
24619  6      32K  10 Receive     1
0:00:00:0000 00000000 iprouting.iospro
```

### EANTC Test Analysis

As expected, we observed no packet loss during the process restart. As the following diagram shows, the average end-to-end latency was 21.2 µs with a variation of only 1.5 µs for a brief period of less than 5 seconds while the hardware tables were updated with newly calculated routes.




**Forwarding Performance Results during Restart of Routing Process**

## 3) Subsystem ISSU, Rollback and Repackaging

**Test Highlights**

- Zero packet loss while forcing a restart of the routing or TCP processes
- Zero packet loss while patching the known recreated vulnerabilities
- Intuitive »patch management« including tagging, rollbacks and repackaging

### Test Objective

The aim of this test was to show that two known vulnerabilities re-inserted into the Cisco IOS image with Software Modularity, which were exploited and lead to a restart of the affected processes, did not lead to any out-of-service times. Also a restart of processes after installing a patch to fix the vulnerability should not lead to any packet loss or increased latency.

Further, this test should show how process status information could be verified, how the installation and activation of patches works and how tag information for patches are configured. The last step of this test should verify the rollback to a specific patch tag and the repackaging feature including a re-installation of the repackage image afterwards.

### Test Methodology

To run these tests, we chose a total of three different patches, two of them fixed security vulnerabilities that had previously been corrected by Cisco but were re-created for test purposes (see www.cisco.com/go/psirt) and one changed the output of the »show cdp neighbor« command.

The first re-created vulnerability we tested is described as *Cisco IOS Malformed OSPF Packet Causes Reload* under the document-ID 61365. Here a malformed OSPF packet caused the routing process to crash.

The second re-created vulnerability we tested is described as *TCP Vulnerabilities in Multiple IOS-Based Cisco Products* under document-ID 50960. A TCP reset packet with matching source and destination addresses and ports and a sequence number within a certain range would bring down a TCP session.

The image we used for the test was modified by Cisco engineers to intentionally include the two known vulnerabilities. In a first test we verified the existence of the vulnerabilities and proved that they showed the expected bahavior. Then we applied the recommended patches and restarted the appropriate IOS processes to verify that the vulnerabilities were fixed. For the CDP patch we verified the different output of the command, which showed an additional line stating that the process was now patched.

After each single test for a patch, we specified a tag, which is a system snapshot in time, providing information about the base image and patch level at a certain time of the patching history.

```
switch-2(DUT)#sh install tag run
Tags defined over software running on location
s72033_rp - Slot 1 :
Tagname         # of Files          Date Committed
-------------   ----------      -----------------
OSPF-patched    2           14:16:56 PDT Aug 18 2005
TCP-patch       3           15:37:09 PDT Aug 18 2005
Tags defined over software running on location
s72033 - Slot 1 :
Tagname         # of Files          Date Committed
-------------   ----------      -----------------
OSPF-patched    2           14:16:56 PDT Aug 18 2005
TCP-patch       3           15:37:09 PDT Aug 18 2005
```

At the end of the three patch tests, we rolled-back one step in the patching history using the previously configured tag information.

For the resulting base and patch level, we applied the »repackage« feature which merges the base image with all active patches and tags currently on the system including the patch history into a single binary file to allow simplified image distribution.

Finally we re-installed the repackaged image to »disk0:/newsys«. After we activated the CDP patch again to this file system, we changed the boot variable to point to »disk0:/newsys« and reloaded the system. This way we verified a) that the repackaged image was consistent and b) that patching can also be done on non-active systems.

### EANTC Test Analysis

During the single test steps

- exploiting the re-created vulnerability,
- patching the image,
- proving that the re-created vulnerability was fixed by a patch,
- doing the rollback and
- repackaging the image,

we verified that we observe zero packet loss and no increase in latency for our reference test streams.

## 4) Embedded Event Manager (EEM) Functionality

**Test Highlights**

- Easy to use multi-purpose Embedded Event Manger (EEM) features for diagnostics and automated policies based on TCL scripting

### Test Objective

The aim of this test was to show the features of integrating predefined and custom TCL scripting into the Embedded Event Manager.

### Test Methodology

We chose two examples to show the integration of TCL scripts, a) a pre-defined script for the generation of event reports and b) a custom »pingcheck« script, which was designed to verify the reachability of a certain IP address and notify an administrator via e-mail of any reachability status changes.

We first configured necessary environment variables. Since the report generator would send all information to a general web server, we needed to configure the server's URL. For the »pingcheck« script, we had to configure the IP address to ping and the address of the e-mail server. To start the scripts, we registered the respective EEM policies by binding the TCL script to the EEM.

To verify the report generator functionality, we intentionally caused a crash of a process and expected the detailed event info to be listed on the report web server. To check the 'pingcheck' script functionality, we manually flapped the route to the targeted IP address.

### EANTC Test Analysis

The data the report collector sends to the central web server creates a reporting database with the following entry types (example):

| Parameter | Reported values (example) |
|---|---|
| Time Reported | Fri Aug 19 02:02:21 UTC 2005 |
| Process | iprouting.iosproc:1 |
| Hostname | 6500-2 |
| OS Version | 12.2(20050810:214544) |
| Crash Reason | SIGKILL, Kill (Signal from user) |
| Information | Process Info |
| | Crash Info |

The TCL scripting interaction worked reliably and quickly. Shortly after we removed the route to the monitored IP address, we received the notification e-mail. And right after we added the route again, another notification was sent to show that the address was reachable again.

## Conclusion

Catalyst 6500 with Cisco IOS Software Modularity fulfilled all of Cisco's functional claims for the areas analyzed. By offering stateful process restarts and subsystem In-Service Software Upgrades (ISSU), the system availability will be increased significantly. This will enable operators to offer highly available services in the data centers, wiring closets, core and distribution of enterprise networks.

These tests further re-affirmed that Cisco IOS Software Modularity on the Catalyst 6500 is easy to use because it is consistent with previous Cisco IOS versions. The new features necessary to manage the modularity enhancements are intuitive and allow effective operation of networks based on the Catalyst 6500 with Cisco IOS Software Modularity. These latest high availability features coupled with a well organized patch management process and integration with Embedded Event Manager can help significantly reduce operational costs and minimize both planned and unplanned downtime.

## About Cisco Systems

Cisco Systems, Inc. (Nasdaq:CSCO) is the worldwide leader in networking for the Internet. News and information are available at http://www.cisco.com.



## About Spirent Communications

Spirent Communications is a worldwide provider of integrated performance analysis and service assurance systems for next-generation network technologies. Spirent's solutions enable customers to develop and deploy network equipment and services more economically and efficiently by emulating real-world conditions in the lab and assuring end-to-end performance of large-scale networks.



http://www.spirentcom.com/

## About EANTC

The European Advanced Networking Test Center (EANTC) offers vendor neutral network test services for manufacturers, service providers and enterprise customers. Primary business areas include interoperability, conformance and performance testing for IP, MPLS, ATM, VoIP, Triple Play, and IP applications.

EANTC AG
Einsteinufer 17, 10587 Berlin, Germany
info@eantc.com, http://www.eantc.com/