



June 2-6, 2008
University of Zurich, Switzerland



University of Zurich

Managing Information from your Network

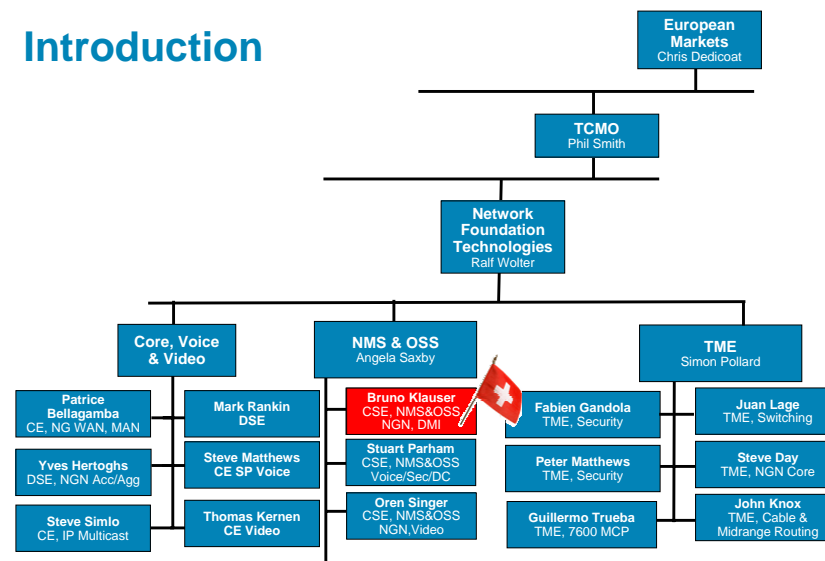
[20080605 – EMANICS – University of Zurich]



Bruno Klauser
Consulting Engineer NMS/OSS
European Markets

bklauser@cisco.com
www.win-people.cisco.com/bklauser

Introduction



seo: www.win-people.cisco.com/bklauser/aboutme.html

seo: <http://zed.cisco.com/confluence/display/EUTMO/Network+Management+Team>

Abstract

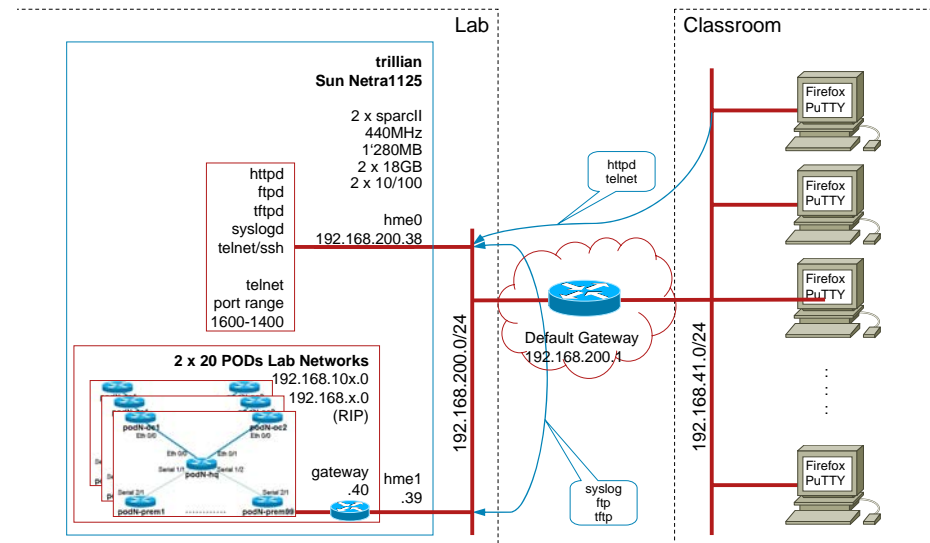
On what network and traffic data does your planning and engineering rely?
Can you validate your design assumptions?
Does your network meet the expectations and requirements implied by business critical services? If so: can you prove it?

Today's network elements provide a plethora of Device Manageability Instrumentation capabilities suitable to answer the need for service relevant information all along a service life cycle:

- service planning
- deployment and activation
- testing and verification
- ongoing service assurance
- troubleshooting and optimization

This session discusses technology fundamentals as well as the choice, design and use of appropriate practices through a combination of presentation and hands-on exercises.

Lab Infrastructure



Agenda

Theoretical Part

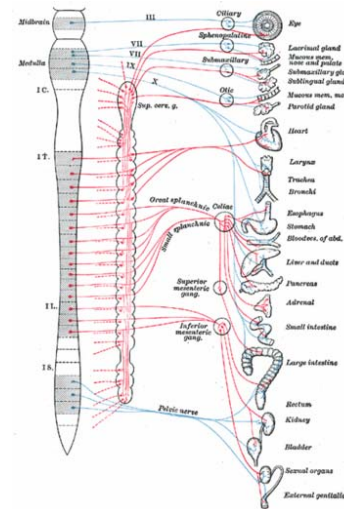
➔ Introduction & Overview

- Service Planning
- Service Deployment & Activation
- Service Testing, Verification & Assurance
- Troubleshooting & Optimization
- Summary

Hands-On Lab Part

- Task 1: Monitoring of Device Resources
- Task 2: Visibility into Traffic Flows
- Task 3: Embedded Event Manager
- Task 4: Collecting Baseline Information

An Analogy ...



The autonomic nervous system (ANS) (or visceral nervous system) is the part of the peripheral nervous system that controls homeostasis, that is the constancy of the content of tissues in gasses, ions and nutrients. It does so mostly by controlling cardiovascular, digestive and respiratory functions, but also salivation, perspiration, diameter of the pupils, micturition - (the discharge of urine), and erection.

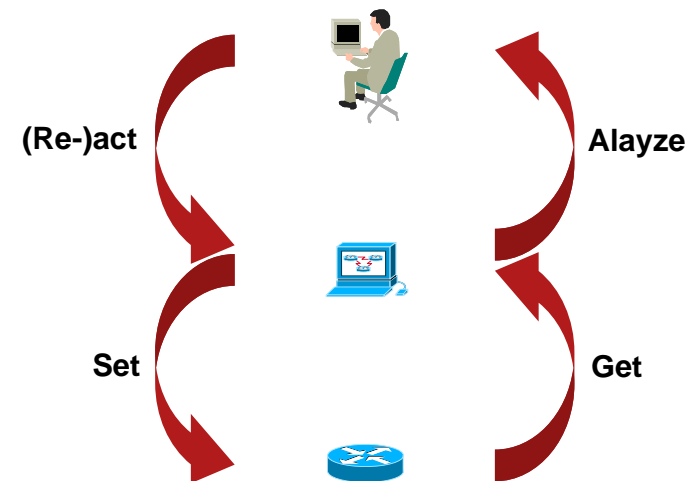
Source: Wikipedia

... But We're Engineers

“Civilization advances by extending the number of important operations which we can perform without thinking about them.”

Alfred North Whitehead,
English Mathematician & Philosopher
(1861–1947)

Network Management in the Past



Introduction & Overview Manageability and Self-* Networks



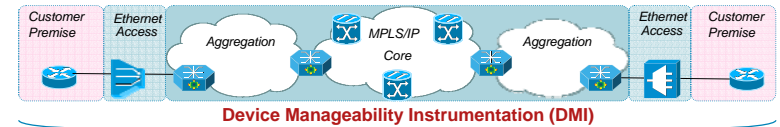
Airliner	Router	Network
8'000 'instruments'	MIB OIDs	Routers
21'000 sensors		Links

- With increasing scale and complexity, things become hard to control entirely from the outside
(hard = inaccurate, time- or resource-consuming, otherwise expensive)

From: Full control by a single central authority

To: Operating a system of self-managing components

Introduction & Overview What is Manageability – 2/2



Fault	Configuration	Performance	Accounting
<ul style="list-style-type: none"> 802.3ah—Link monitoring and remote fault indication 802.1 ag—Continuity check, L2 ping, trace, AIS MPLS OAM—LSP ping, LSP trace, VCCV IP OAM—Ping, Trace, BFD, ISG per session EEM—Embedded Event Manager EVENT-MIB—OID-based triggers, events, or SNMP Set, IETF DISMON EXPRESSION-MIB—OID expression-based triggers, IETF DISMON ... 	<ul style="list-style-type: none"> E-LMI—(service parameter and status signaling) E-DI—(Enhanced Device Interface, CLI, Perl, IETF Netconf) XML PI—(IETF Netconf) TR-069 KRON—command scheduler Config change—logging and notifications Config replace and rollback Diff—context diff utility MIB persistence ... 	<ul style="list-style-type: none"> IP SLA—delay, jitter, packet loss, MPLS health monitoring, advanced object tracking CBQoS MIB—(class-based QoS) NBAR RMON ERM—Embedded Resource Manager GOLD—Generic Online Diagnosis ... 	<ul style="list-style-type: none"> Flexible NetFlow—IETF IPFIX BGP policy accounting—includes AS information Periodic MIB bulk data collection and transfer ...
			Security
			<ul style="list-style-type: none"> Auto Secure—one-touch device hardening LDP Auth—message authentication Routing Auth—MD5 authentication, BGP, OSPF ...

See also: www.cisco.com/go/instrumentation

Introduction & Overview Questions during a Service Life Cycle

Is there room for yet another service?

- How do we perform today ?
- Are there existing issues ?
- Will we meet specs ?
- Resource consumption ?
- ...

How to configure?

- 1 or many nodes ?
- CLI, scripts, automation ?
- Can we afford downtime ?
- Quality & Security ?
- ...

Is it working as specified?

- Configuration ?
- Control Plane ?
- Data Plane ?
- Were my design assumptions right ?
- ...

Service Planning

Deployment & Activation

Testing & Verification

Troubleshooting & Optimization

Service Assurance

- How to Diagnose ?
- Could we offer even tighter SLA ?
- Automate Remedy & Mitigation ?
- ...

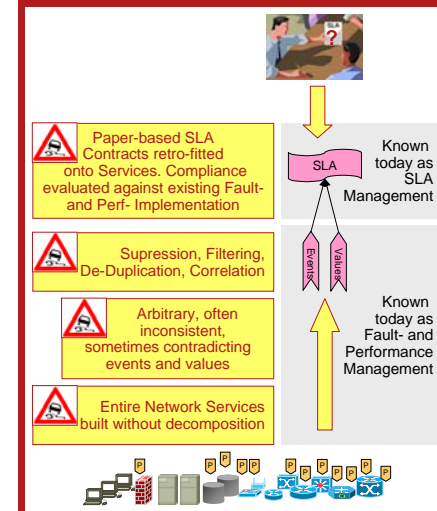
What if something goes wrong?

- Will we breach any SLA ?
- What is our performance ?
- ...

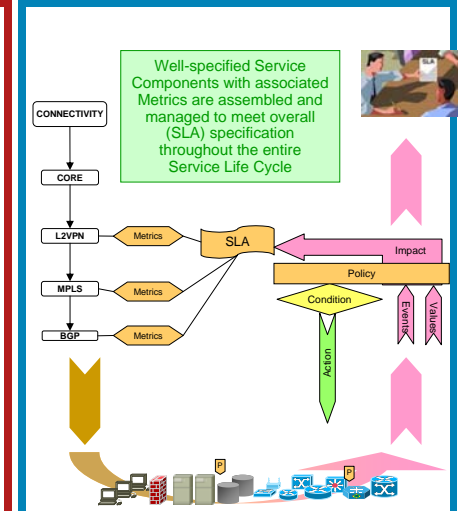
Are we meeting SLA?

Introduction & Overview Evolving the Operations Paradigm

From: Open Loop, Retro-Fit



To: Closed Loop, Deterministic Design



Agenda

Theoretical Part

Introduction & Overview

➔ Service Planning

- Service Deployment & Activation
- Service Testing, Verification & Assurance
- Troubleshooting & Optimization
- Summary

Hands-On Lab Part

- Task 1: Monitoring of Device Resources
- Task 2: Visibility into Traffic Flows
- Task 3: Embedded Event Manager
- Task 4: Collecting Baseline Information

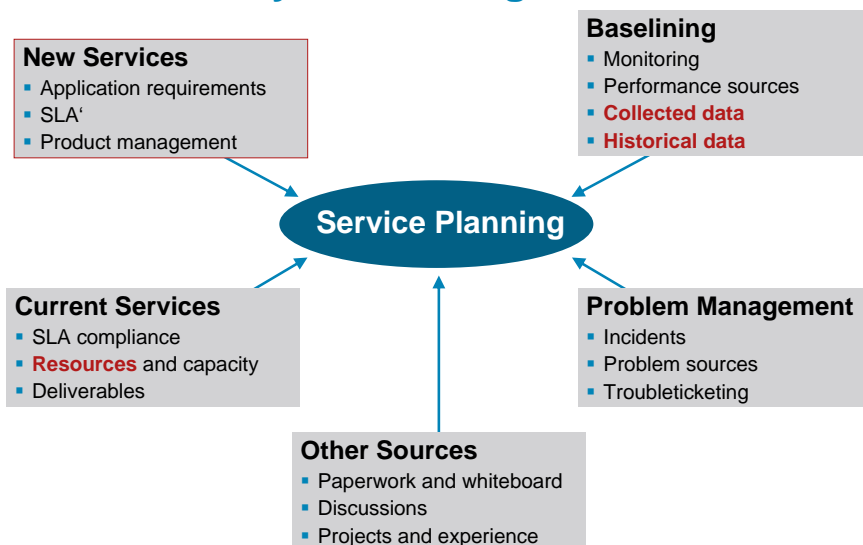
“Plan [noun]

A set of decisions about how to do something in the future.”

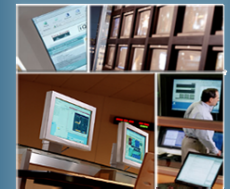


Cambridge Dictionary
<http://dictionary.cambridge.org>

Service Planning Learn from your existing Services ...



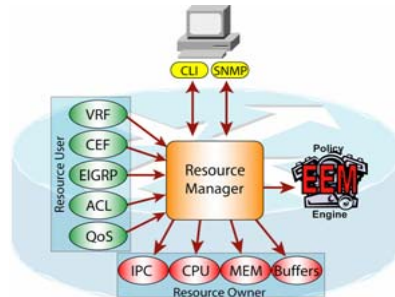
How Is My Current Use of Resources?



Service Planning Embedded Resource Manager (ERM)

Monitor system resource usage to better understand scalability needs

- **Resource:** CPU, Buffer, Memory for System or Line Card
- **Resource User (RU):** Entity or application that consumes one or more resources, e.g. a process
- **Resource Owner (RO):** Entity that allocates its resources to a RU, e.g. CPU, memory, buffer
- **Threshold Notifications:**
 - **System Global** upon entire resource reaching specified value. Notification sent to all RUs.
 - **User Local** upon a specified RU's utilization reaching specified value. Notification sent to specified RU only.
 - **Per User Global** upon entire resource reaching specified value. Notification sent to specified RU only
- **Interface into EEM**



Available since 12.3(14)T (1800, 2800, 3800, 7200)

© 2008 Cisco Systems, Inc. All rights reserved. Cisco Public

17

Service Planning Example: Monitoring Resources

- **Problem:** During the planning cycle, we would like to understand if total CPU usage reaches critical levels
- **Solution:** Define an ERM policy to notify upon resource depletion

```
resource policy
policy my-erm-policy-1 type iosprocess
system
cpu total
critical rising 90 interval 15 falling 20 interval 10 global
major rising 70 interval 15 falling 15 interval 10 global
minor rising 60 interval 15 falling 10 interval 10 global
!
```

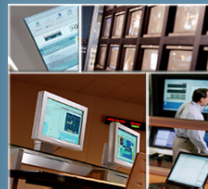
- ➔ If **Total** CPU Usage Count Rises Above 90% at an Interval of 15s, a Critical Up Notification Is Sent to the iosprocess RU

```
Feb 17 13:32:18.283: %SYS-4-CPURESRISE: System is seeing global
cpu util 62% at total level more than the configured minor limit 60%
```

© 2008 Cisco Systems, Inc. All rights reserved. Cisco Public

18

What Traffic Flows Through My Network?



© 2008 Cisco Systems, Inc. All rights reserved. Cisco Public

19

Service Planning What is NetFlow ?

- Developed and patented at Cisco® Systems in 1996
- NetFlow is the defacto standard for acquiring IP operational data
- Provides network and security monitoring, network planning, traffic analysis, and IP accounting
- NetFlow v9 serves as the basis for IETF IPFIX Standard (RFC3954)

Network World article – NetFlow Adoption on the Rise

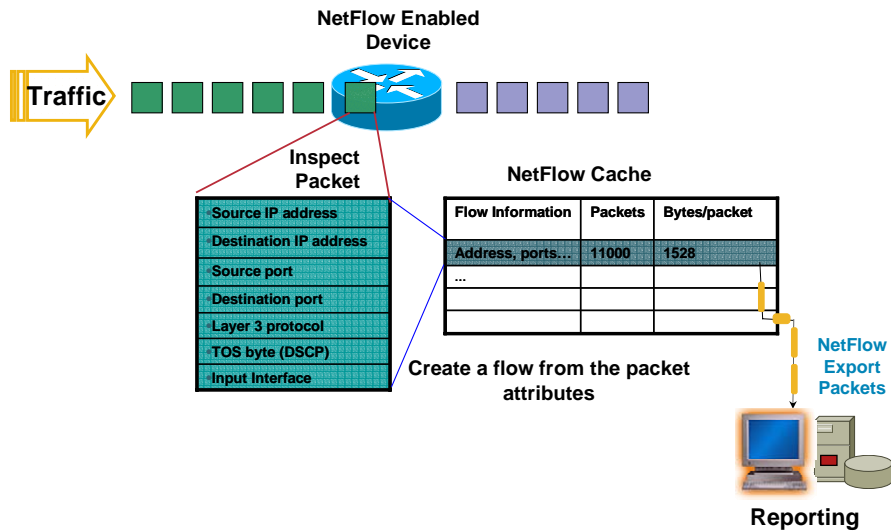
<http://www.networkworld.com/newsletters/nsm/2005/0314nsm1.html>



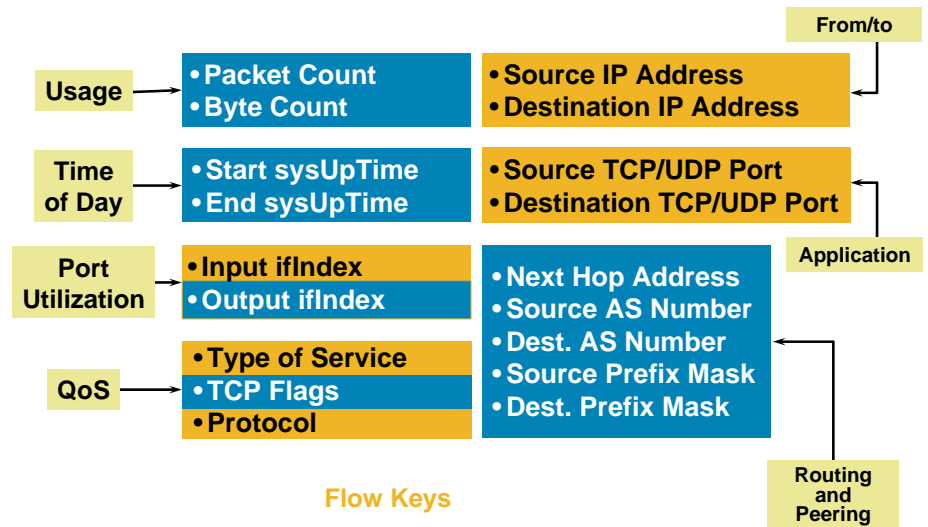
© 2008 Cisco Systems, Inc. All rights reserved. Cisco Public

20

Flow Is Defined By Seven Unique Keys



Version 5 Flow Format



NetFlow Cache Example

1. Create and update flows in NetFlow cache

SrcIf	SrcIPadd	DstIf	DstIPadd	Protocol	TOS	Flgs	Pkts	Src Port	Src Msk	Src AS	Dst Port	Dst Msk	Dst AS	NextHop	Bytes/Pkt	Active	Idle
Fa1/0	173.100.21.2	Fa0/0	10.0.227.12	11	80	10	11000	00A2	/24	5	00A2	/24	15	10.0.23.2	1528	1745	4
Fa1/0	173.100.3.2	Fa0/0	10.0.227.12	6	40	0	2491	15	/26	196	15	/24	15	10.0.23.2	740	41.5	1
Fa1/0	173.100.20.2	Fa0/0	10.0.227.12	11	80	10	10000	00A1	/24	180	00A1	/24	15	10.0.23.2	1428	1145.5	3
Fa1/0	173.100.6.2	Fa0/0	10.0.227.12	6	40	0	2210	19	/30	180	19	/24	15	10.0.23.2	1040	24.5	14

2. Expiration

- Inactive Timer Expired (15 Sec Is Default)
- Active Timer Expired (30 Min Is Default)
- NetFlow Cache Is Full (Oldest Flows Are Expired)
- RST or FIN TCP Flag

SrcIf	SrcIPadd	DstIf	DstIPadd	Protocol	TOS	Flgs	Pkts	Src Port	Src Msk	Src AS	Dst Port	Dst Msk	Dst AS	NextHop	Bytes/Pkt	Active	Idle
Fa1/0	173.100.21.2	Fa0/0	10.0.227.12	11	80	10	11000	00A2	/24	5	00A2	/24	15	10.0.23.2	1528	1800	4

3. Aggregation

4. Export version

Non-aggregated flows—export version 5 or 9

5. Transport protocol



E.g. Protocol-Port Aggregation Scheme Becomes

Protocol	Pkts	SrcPort	DstPort	Bytes/Pkt
11	11000	00A2	00A2	1528

Aggregated Flows—Export Version 8 or 9

NetFlow Export Version 5 and Main Cache Configuration Example

```

Router(config)# interface <slot/port/subinterface>
Router(config-if)# ip flow ingress
Router(config-if)# ip flow egress

Router(config)# ip flow-cache entries <number>
Router(config)# ip flow-cache timeout active <minutes>
Router(config)# ip flow-cache timeout inactive <seconds>

Router(config)# ip flow-export version 5 peer-as
Router(config)# ip flow-export destination 10.10.10.10 1234
Router(config)# ip flow-export source loopback 0
    
```

Show NetFlow Information 'show ip cache flow'

```
router_A#sh ip cache flow
IP packet size distribution (85435 total packets):
1-32 64 96 128 160 192 224 256 288 320 352 384 416 448 480
.000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

512 544 576 1024 1536 2048 2560 3072 3584 4096 4608
.000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes
2728 active, 368 inactive, 85310 added
463824 aged polls, 0 flow alloc failures
Active flows timeout in 30 minutes
Inactive flows timeout in 15 seconds
last clearing of statistics never

Protocol      Total    Flows    Packets Bytes    Packets Active(Sec) Idle(Sec)
-----
Flows        /Sec    /Flow  /Pkt    /Sec    /Flow    /Flow
TCP-X         2         0.0      1 1440      0.0      0.0      9.5
TCP-other    82580     11.2      1 1440     11.2      0.0     12.0
Total:       82582     11.2      1 1440     0.0      0.0     12.0

SrcIf      SrcIPaddress  DstIf      DstIPaddress  Pr SrcP DstP  Pkts
-----
Et0/0      132.122.25.60 Se0/0      192.168.1.1   06 9AEE 0007 1
Et0/0      139.57.220.28 Se0/0      192.168.1.1   06 708D 0007 1
Et0/0      165.172.153.65 Se0/0      192.168.1.1   06 CB46 0007 1
```

Packet sizes

of active flows

Rates and duration

Flow details cache

'show ip cache verbose flow'

```
router_A#sh ip cache verbose flow
IP packet size distribution (23597 total packets):
1-32 64 96 128 160 192 224 256 288 320 352 384 416 448 480
.000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

512 544 576 1024 1536 2048 2560 3072 3584 4096 4608
.000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes
1323 active, 2773 inactive, 23533 added
151644 aged polls, 0 flow alloc failures
Active flows timeout in 30 minutes
Inactive flows timeout in 15 seconds
last clearing of statistics never

Protocol      Total    Flows    Packets Bytes    Packets Active(Sec) Idle(Sec)
-----
Flows        /Sec    /Flow  /Pkt    /Sec    /Flow    /Flow
TCP-X         2         0.0      1 1440      0.0      0.0     9.5
TCP-other    82580     11.2      1 1440     11.2      0.0     12.0
Total:       82582     11.2      1 1440     0.0      0.0     12.0

SrcIf      SrcIPaddress  DstIf      DstIPaddress  Pr SrcP DstP  Pkts
-----
Port Msk AS  Port Msk AS  NextHop
Et0/0      216.120.112.114 Se0/0      192.168.1.1   06 00 10 1
5FA7 /0 0    0007 /0 0    0.0.0.0      1440 0.0
Et0/0      175.182.253.65 Se0/0      192.168.1.1   06 00 10 1
```

Flow rate and duration

Destination information

ToS byte and TCP flags

Source mask and ISP AS

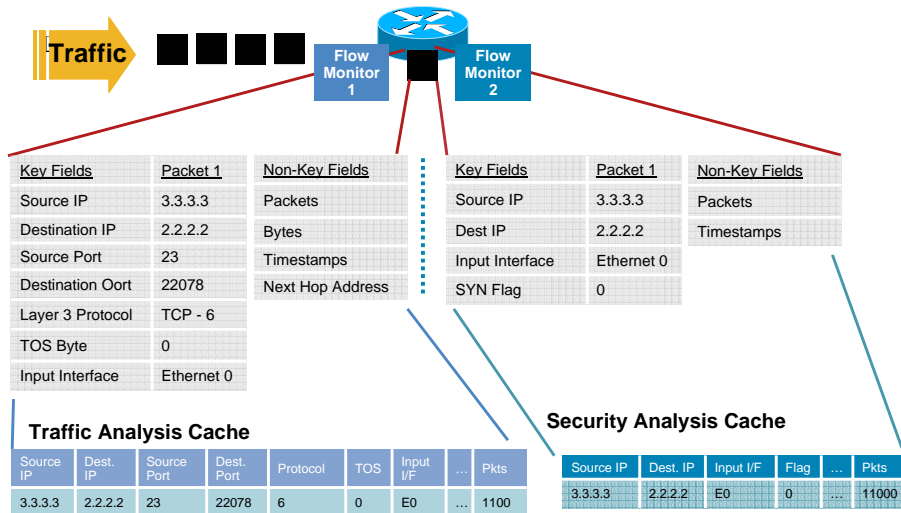
Extensibility and Flexibility Requirements Phases Approach

- Traditional NetFlow with the v5, v7, or v8 NetFlow export
 - New requirements: build something flexible and extensible
 - Phase 1: **NetFlow version 9**
 - Advantages: **extensibility**
 - Integrate new technologies/data types quicker (MPLS, IPv6, BGP next hop, etc.)
 - Integrate new aggregations quicker
 - Note: for now, the template definitions are fixed
 - Phase 2: **Flexible NetFlow**
 - Advantages: cache and export content **flexibility**
 - User selection of flow keys
 - User definition of the records
- } **Exporting Process**
- } **Metering Process**

Flexible NetFlow High Level Concepts and Advantages

- Flexible NetFlow feature allows user configurable NetFlow record formats, selecting from a collection of fields:
 - Key
 - Non-key
 - Counter
 - Timestamp
- Advantages:
 - Tailor a cache for specific applications, not covered by existing 21 NetFlow features
 - Better scalability since flow record customization for particular application reduces number of flows to monitor
 - Different NetFlow configuration:
 - Per subinterface
 - Per direction (ingress/egress)
 - Per sampler
 - Etc.

Flexible NetFlow Multiple Monitors with Unique Key Fields



Flexible Flow Record—Key Fields

IPv4		Routing	Transport	
IP (Source or Destination)	Payload Size	src or dest AS	Destination Port	TCP Flag: ACK
Prefix (Source or Destination)	Packet Section (Header)	Peer AS	Source Port	TCP Flag: CWR
Mask (Source or Destination)	Packet Section (Payload)	Traffic Index	ICMP Code	TCP Flag: ECE
Minimum-Mask (Source or Destination)	TTL	Forwarding Status	ICMP Type	TCP Flag: FIN
Protocol	Options bitmap	Is-Multicast	IGMP Type	TCP Flag: PSH
Fragmentation Flags	Version	IGP Next Hop	TCP ACK Number	TCP Flag: RST
Fragmentation Offset	Precedence	BGP Next Hop	TCP Header Length	TCP Flag: SYN
ID	DSCP	Flow	TCP Sequence Number	TCP Flag: URG
Header Length	TOS	Sampler ID	TCP Window-Size	UDP Message Length
Total Length		Direction	TCP Source Port	UDP Source Port
		Interface	TCP Destination Port	UDP Destination Port
		Input	TCP Urgent Pointer	
		Output		

Flexible Flow Record—Non-Key Fields

Counters	Timestamp	IPv4
Bytes	sysUpTime First Packet	Total Length Minimum
Bytes Long	sysUpTime First Packet	Total Length Maximum
Bytes Square Sum		TTL Minimum
Bytes Square Sum Long		TTL Maximum
Packets		
Packets Long		

- Plus any of the potential “key” field: will be the value from the first packet in the flow

Agenda

Theoretical Part

Introduction & Overview

Service Planning

➔ Service Deployment & Activation

Service Testing, Verification & Assurance

Troubleshooting & Optimization

Summary

Hands-On Lab Part






Task 1: Monitoring of Device Resources

Task 2: Visibility into Traffic Flows

Task 3: Embedded Event Manager

Task 4: Collecting Baseline Information

Deployment & Activation Definition of Activities

	Deployment	Move physical network equipment into its operating location
	Commissioning	Make new network equipment ready for use and reachable by operations, NMS
	Configuration	Configure a network element depending on its role and function in the network
	Provisioning	Configure portions of a network for the purpose of a specific user and/or service
	Activation	Enable users to start using a service

Focus

On the CLI of a Single Router...



Deployment & Activation IOS Configuration Features

- **Contextual configuration diff utility** (from 12.3(4)T, 12.2(25)S)
 - Easily show differences between running and startup configuration
 - Compare any two configuration files
- **Config change logging and notification** (from 12.3(4)T, 12.2(25)S)
 - Tracks config commands entered per user, per session
 - Notification sent indicating config change has taken place—changes can be retrieved via SNMP
- **Configuration replace and rollback** (from 12.3(7)T, 12.2(25)S)
 - Replace running config with any saved configuration (only the diffs are applied) to return to previous state
 - Automatically save configs locally or off box
- **Configuration locking** (from 12.3(14)T, 12.2(25)S)
 - Ensures exclusive configuration change access

Deployment & Activation Example: Using Config Rollback

- **Problem:** critical config change to a remote router may result in loss of connectivity, requiring a reload
- **Solution:** replace the running configuration with the latest good archive after two minutes—unless the change being made is confirmed

```
Router#show archive
There are currently 4 archive configurations saved.
The next archive file will be named disk0:/config-archive-4
Archive # Name
0
1 disk0:/config-archive-1
2 disk0:/config-archive-2
3 disk0:/config-archive-3 <- Most Recent

Router#config replace disk0:/config-archive-3 time 120
:
... your Config Change work here ...
:
Router# no config replace disk0:/config-archive-3
```

Deployment & Activation Tool Command Language (TCL)

- Language resources found at: <http://www.tcl.tk/>
- TCL 7.x has been in Cisco IOS since 1994
- TCL 8.3.4 first released in Cisco IOS in 12.3(2)T and merged into 12.2(25)S
- Use 12.3(14)T or later for best results
- Signed TCL Scripts introduced in 12.4(15)T



```
Router#tclsh slot0:myscript.tcl
Router#tclsh
Router(tcl)#source tftp://10.1.1.1/myscript.tcl
```

- Use low-memory to prevent malloc failures
- TCL process runs at medium priority, so be careful with loops

```
Router(config)#scripting tcl low-memory <water_mark>
```

Deployment & Activation Tool Command Language (TCL)

- <http://www.cisco.com/go/ciscobeyond>
- <http://www.cisco.com/go/eem>
- <http://www.cisco.com/go/ioscommercial>

Example: A VPN failure is defined as failure to reach a set of remote peer's L3 tunnel interface(s) that are configured using GRE + IPSEC over DMVPN

- "Guide To Writing EEM Policies" documentation

```
Router#tclsh
Router(tcl)#puts "Hello EMANICS"
Hello Networkers
Router(tcl)#ios_config "interface fa0/0"
"description EMANICS Uplink"
Router(tcl)#exit
Router#
```

TCL Cisco IOS
Extended Commands
TCL Built In Command
Cisco IOS Command

What if CLI Doesn't Scale?



Deployment & Activation Zero-Touch Deployment Methods

Method	Cisco IOS Deployment Agents	External Mediation Server	Notes
DOCSIS	DOCSIS	Cisco Broadband Access Center (BAC)	For Cable Modem Access Only Widely Standardized
TR-069	TR-069	Cisco Broadband Access Center (BAC)	For DSL Access Standard Is Work in Progress with Currently Loose Definition, Check Interop Test from Plugfest
EEM	Embedded Event Manager	FTP, TFTP, SCP,...	Flexibility for Scenarios Not Covered by Any Other Method Sometimes Used in Concert with Other Methods
Kron	Kron and TCL	FTP, TFTP, SCP,...	When EEM Is Not Available
DHCP	DHCP	Cisco Network Registrar, TFTP	Agnostic of Access Technology Partially Standardized, Multiple Options Used
CNS	CNS Config Agent CNS Image Agent CNS Inventory Agent CNS Event Agent	Cisco Configuration Engine	Most Secure and Robust Agnostic of Access Technology Agnostic of IP Addressing

Zero-Touch Deployment = Embedded Agents + External Mediation

Deployment & Activation Example: Zero-Touch Deployment – 1/2

▪ **Problem:** A large number of Teleworker Routers have to be deployed. Access Technology and Service Provider vary; IP Addressing is not known in advance

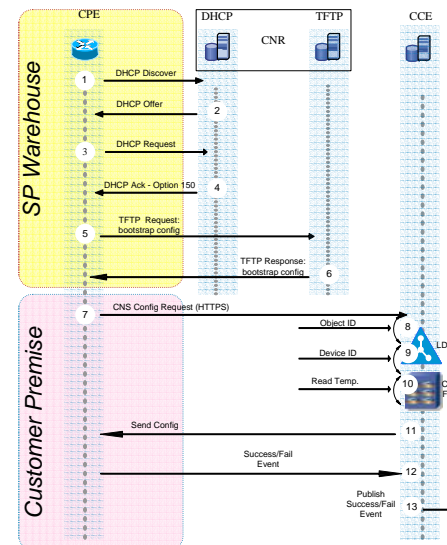
▪ **Solution:** Pre-Configure Routers with a **generic bootstrap config**. This config ensures initial IP connectivity, identifies the device and communicates back to Configuration Engine for appropriate config

```
Router # cns id hardware-serial
Router # cns config initial MyConfigEngine 80 event no-persist
Router # cns id hardware-serial event
Router # cns event MyConfigEngine 11011
```

Note: Many other options for ID exist and are often used instead of hardware-serial:

```
AMBO730F2X(config)#cns id ?
Async Async interface
SVT Bridge-Group Virtual Interface
CTunnel CTunnel interface
Dialer Dialer interface
Ethernet IEEE 802.3
FastEthernet FastEthernet IEEE 802.3
Group-Async Async Group interface
Loopback Loopback interface
MFR Multilink Frame Relay bundle interface
Multilink Multilink-group interface
Tunnel Tunnel interface
Vif PGM Multicast Host interface
Virtual-PPP Virtual PPP interface
Virtual-Template Virtual Template interface
Virtual-TokenRing Virtual TokenRing
hardware-serial Use hardware serial number as unique ID
hostname Use hostname as unique ID
string Use an arbitrary string as the unique ID
```

Deployment & Activation Example: Zero-Touch Deployment – 2/2



1. CPE sends DHCP Discover
2. DHCP Server replies with Offer
3. CPE sends DHCP Request
4. DHCP Server replies with option 150
5. CPE requests bootstrap-config file via TFTP
6. TFTP server sends CPE bootstrap-config file

=> CPE is shipped to Customer Site
=> Customer Order linked to CPE ID

7. CPE sends HTTP request to CNS-CE
8. CNS-CE verifies object ID
9. CNS-CE verifies Device ID
10. CNS-CE reads template from File System
11. CNS-CE sends Config (Config = template + parameters from LDAP)
12. Successful event
13. Publish success event

Deployment & Activation XML PI is ... – 1/3

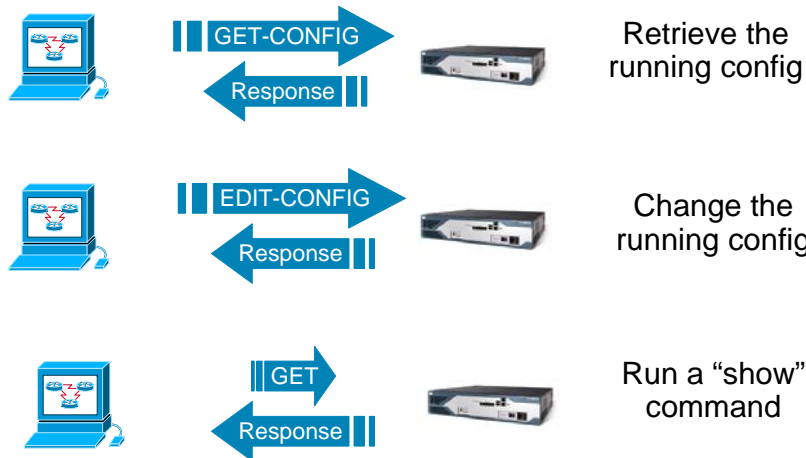
An **XML Interface** to a **Cisco IOS Network Element**, for customers and partners needing to remotely adapt and control the behavior of Cisco devices.

XML-PI provides **unambiguous** and **robust information access** without the complexity and expense of screen-scraping technologies or external mediation software.

Deployment & Activation XML PI is ... – 2/3

- **XML-PI** runs on top of **NETCONF** and **SSH V2** to send and receive CLI commands through a reliable stack without screen scraping or expect scripts
- XML-PI and NETCONF is currently being implemented on many major Cisco platforms
- Devices can have their running configuration changed
- Applications can retrieve the current running configuration
- NETCONF uses XML-based data encoding for the configuration data and protocol messages
- NETCONF runs over SSH and BEEP

Deployment & Activation XML PI is ... – 3/3



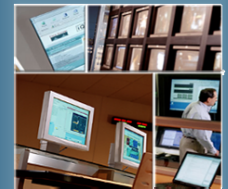
Deployment & Activation Example: Edit the running config



Deployment & Activation XML PI – Why do we care ?

- **IETF** standard-based configuration management
- Provides **reliable and secure transport** of configurations over encrypted TCP connections
- Improves the **speed of configuration** changes since it is not limited to console speeds
- **Eliminates** scripting and “**screen scraping**” via telnet
- Allows **concurrent configuration changes**
- Leverages the vast number of **XML** tools available
- Foundation for future **XML** configuration capabilities

Multiple Devices and Scripting

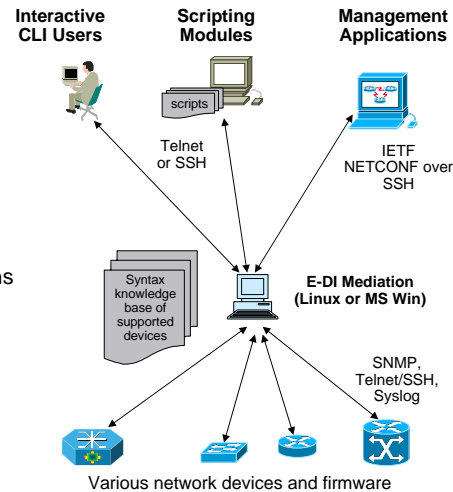


Deployment & Activation What is Enhanced Device Interface (E-DI) ?

- E-DI is:
 - An extension to the network device's interface
 - Complementary to EMS/NMS
- E-DI provides three interfaces
 - Enhanced Command Line Interface (CLI)** to human users
 - Perl Scripting Interface** and platform for scripting applications
 - XML programmatic interface** to management applications

IETF NETCONF draft 5 compliant

Codeployment with Cisco IOS XML PI



Deployment & Activation New in Enhanced Device Interface 2.2

New Feature	Description
Linux / Windows	Support for server and client apps on Linux and Windows
IDU	Incremental Device Support
Operational Data Model	XML interface for the show commands from NEs
Macro CLI Commands	<ul style="list-style-type: none"> - Define consistent Macros for a set of commands across various OS versions - CLI and GUI Interface for Macro CLI configuration - Provision the Network using Macro Grouping capability
Command Modeler and Analyzer	<ul style="list-style-type: none"> - IDE over the EDI Device CLI KB. - Analyze Commands across Device/OS. - Model Based Configs can be created using this.

Free of Charge Download from:

<https://upload.cisco.com/cgi-bin/swc/fileexg/main.cgi?CONTYPES=ccu-forum>

(easier to remember url: <http://tinyurl.com/2jrttr>)

Agenda

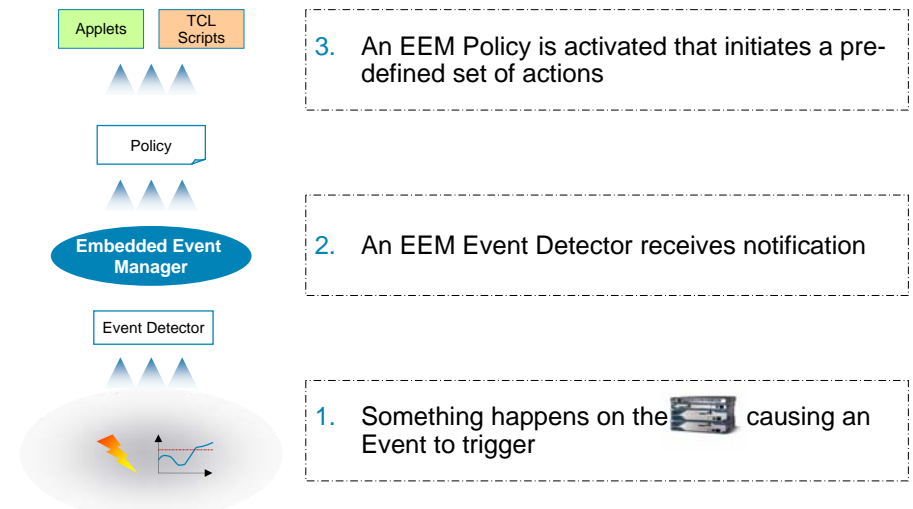
Theoretical Part

- Introduction & Overview
- Service Planning
- Service Deployment & Activation
- ➔ **Service Testing, Verification & Assurance**
- Troubleshooting & Optimization
- Summary

Hands-On Lab Part

- Task 1: Monitoring of Device Resources
- Task 2: Visibility into Traffic Flows
- Task 3: Embedded Event Manager
- Task 4: Collecting Baseline Information

Service Testing, Verification and Assurance Embedded Event Manager (EEM)



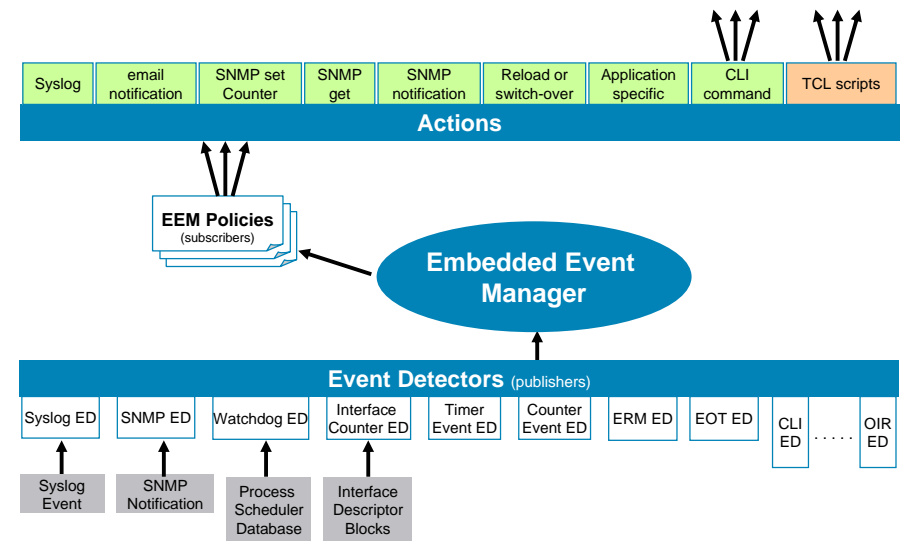
Service Testing, Verification and Assurance What Is Embedded Event Manager (EEM) ?

- Embedded monitoring of different components of the system via a set of software agents (event detectors)
- Event detectors (ED) notify EEM when an event of interest occurs; based on this, a policy will trigger an action to be taken
- Advantages: Local programmable actions, triggered by specific events – growing set of detectors and actions:

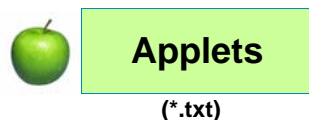
- Version 1.0 introduced in 12.0(26)S, 12.3(4)T
- Version 2.0 introduced in 12.2(25)S
- Version 2.1 introduced in 12.3(14)T
- Version 2.2 introduced in 12.4(2)T
- Version 2.3 introduced in 12.4(11)T
- Upcoming Version 2.4 in 12.4(20)T
- Upcoming Version 3.0 in 12.5(pi1)T
- stay tuned ...

Adds multi-event correlation

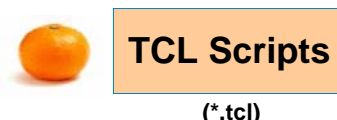
Service Testing, Verification and Assurance EEM Architecture



EEM Policies can be either Applets or TCL Scripts

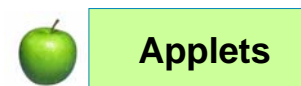


- Applets are created using a set of CLI commands
- The applet becomes part of the Cisco IOS configuration file and is persistent across system reboots
- Use a single “event” statement following by a number of “action” statements



- TCL scripts cannot be built from the switch CLI
- This form of script offers a more flexible and powerful option for network administrators to apply actions on a given event occurrence
- Like the applet, a registered TCL script is persistent across system reboots

EEM Applets



Environment Variable(s)
(Optional)

1. Configure any required environment variables

Applet Name

2. Register applet

Event Statement

3. Define event used to trigger applet

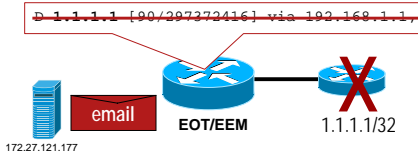
Action Statement

4. Specify actions to be taken

Service Testing, Verification and Assurance Example: Layer 3 Path Failure Detection

▪ **Problem:** A Notification is required upon failure of a specific route

▪ **Solution:** Track the Route using Enhanced Object Tracking (EOT) and Embedded Event Manager (EEM)



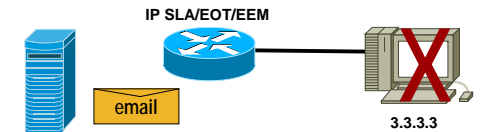
```
track 400 ip route 1.1.1.1/32 reachability
  delay down 10 up 10
!
event manager environment my_server 172.27.121.177
event manager environment my_from router-abc@customer.com
event manager environment my_to attach@cisco.com
event manager environment my_route 1.1.1.1/32
!
event manager applet email_track_iproute
event track 400 state down
action 1.0 syslog msg "Prefix to [$my_route] has been withdrawn!"
action 1.1 mail server "$my_server" to "$_email_to" from "$my_from"
  subject "EEM: Prefix to Remote Site [$my_route] is DOWN" body ""
action 1.2 syslog msg "EEM: Path Failure alert email sent!"
```

Service Testing, Verification and Assurance Example: Track Server Reachability

IP SLA
ip sla 10
icmp-echo 3.3.3.3
timeout 500
frequency 3
ip sla schedule 10 life forever start-time now

Embedded Object Tracking (EOT)
track 10 rtr 10 reachability
delay down 10 up 20

Environment Variables
(\$_* variables to be defined)



EEM Applet
event manager applet email_server_unreachable
event track 10 state down
action 1.0 syslog msg "Ping has failed, server unreachable!"
action 1.1 cli command "enable"
action 1.2 cli command "del /force flash:server_unreachable"
action 1.3 cli command "show clock | append server_unreachable"
action 1.4 cli command "show ip route | append server_unreachable"
action 1.5 cli command "more flash:server_unreachable"
action 1.6 mail server "\$_email_server" to "\$_email_to" from "\$_email_from" subject "Server Unreachable: ICMP-Echos Failed" body "\$_cli_result"
action 1.7 syslog msg "Server unreachable alert has been sent to email server!"

EEM TCL Scripts



TCL Scripts

Event Register Keyword

Environment Variables (Optional)

Namespace Import

Body of Code

EEM TCL Script Example

```
:::cisco::eem::event_register_syslog occurs 1 pattern. *%SYS-4-FREEMEM.* queue_priority low nice 1 maxrun 90
#####
#
# Revision # 1.7
# Last Updated : September 23, 2007
# Author/Contributor : David Lin, dalin@cisco.com
# Description This TCL script utilizes the Memory Threshold feature introduced in Cisco IOS 12.2(18)S, 12.0
# This feature allows one to mitigate low-memory conditions on a router.
# When free processor or I/O memory has fallen below a configured threshold,
# an email will be sent and include output from the syslog, "show version", "show memory summary"
# and "sh processes memory sorted holding"
#
# Requirements : -Email related environment variables-
# event manager environment _email_server <your-mailserver-ipaddress or dns-name>
# event manager environment _email_from <your-email-from-address>
# event manager environment _email_to <your-email-to-address>
#
# Example: event manager environment _email_server 10.10.10.10
# event manager environment _email_from router-123@cisco.com
# event manager environment _email_to noc@cisco.com
```

Event register

EEM runtime

Default = 20 seconds
Increase this value if you see a "Process Forced Exit" message from the router.

EEM TCL Script Example

Other types of event registers you may encounter...

None: Triggered manually via "event manager run" command.

```
::cisco::eem::event_register_none queue_priority low nice 1 maxrun 60
```

Watchdog Timer: Triggered by time (in sec) specified by value/environment variable after the keyword "time"

```
::cisco::eem::event_register_timer watchdog name foobar time $time_period queue_priority low nice 1
```

(The above example requires the global command "event manager environment time_period <sec>")

Syslog: Triggered by pattern match of syslog msg

```
::cisco::eem::event_register_syslog occurs 1 pattern .%SYS-5-CONFIG-I.* queue_priority low nice 1 maxrun 90
```

Object Tracking: Triggered by state of Enhanced Object Tracking (EOT) reaching "DOWN" state.

```
::cisco::eem::event_register_track 1 state up queue_priority low nice 1
```

EEM TCL Script Example

Other types of event registers you may encounter (cont.)

Cron Job

```
::cisco::eem::event_register_timer cron name business_hours cron_entry "0 9-17 * * 1-5" queue_priority low nice 1
```

The above cron job will trigger every hour between 9am-5pm, Mon-Fri

The cron_entry "0 9-17 * * 1-5" will do this:

The 0 means the first minute of the hour.

The 9-17 means hours 9am to 5pm

The next * means every day of the month.

The next * means every month.

The final 1-5 means Monday through Friday.

EEM TCL Script Example

```
...
# Namespace imports
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*

#--- Check required environment variable(s) has been defined
if {[info exists _email_server]} {
    set result "EEM Policy Error: variable _email_server has not been set."
    error $result $errorInfo
}
if {[info exists _email_to]} {
    set result "EEM Policy Error: variable _email_to has not been set."
    error $result $errorInfo
}
if {[info exists _email_from]} {
    set result "EEM Policy Error: variable _email_from has not been set."
    error $result $errorInfo
}
}
```

Import EEM Library Files

Environment
Variable Check

EEM TCL Script Example

```
#----- hostname -----
set routename [info hostname]
#
#----- " cli open" -----
#
if [catch {cli open} result] {
    error $result $errorInfo
} else {
    array set cli $result
}
```

copy hostname to variable
'routename'

Open CLI command

EEM TCL Script Example

```
#----- "show commands" -----
if [catch {cli_exec $cli(fd) "enable"} result] {
    error $result $errorInfo
}

if [catch {cli_exec $cli(fd) "show version"} result] {
    error $result $errorInfo
}
set show_version $result

if [catch {cli_exec $cli(fd) "show memory summary | include ^
                                Head|^
                                I/O|^Processor"} result] {
    error $result $errorInfo
}
set output_1 $result

if [catch {cli_exec $cli(fd) "show processes memory sorted holding"} result] {
    error $result $errorInfo
}
set output_2 $result

#----- end of show commands -----
```

Enter CLI enable mode

Capture show commands

EEM TCL Script Example

```
#----- send mail -----
action_syslog msg "Creating mail header..."
set body [format "Mailservername: %s" "$_email_server"]
set body [format "%s\nFrom: %s" "$body" "$_email_from"]
set body [format "%s\nTo: %s" "$body" "$_email_to"]
set _email_cc ""
set body [format "%s\nCc: %s" "$body" ""]
set body [format "%s\nSubject: %s\n" "$body" "Router is running low on memory! (hostname:$routername)"]
set body [format "%s\n%" "$body" "Report Summary:"]
set body [format "%s\n%" "$body" "- Show Version"]
set body [format "%s\n%" "$body" "- Syslog Message"]
set body [format "%s\n%" "$body" "- Show Memory Summary"]
set body [format "%s\n%" "$body" "- Show Processes Memory Sorted Holding"]
set body [format "%s\n%" "$body" "----- Show Version -----"]
set body [format "%s\n%" "$body" "$show_version"]
set body [format "%s\n%" "$body" "----- Syslog Message -----"]
set body [format "%s\n%" "$body" "$syslog_msg"]
set body [format "%s\n%" "$body" "----- Show Memory Summary -----"]
set body [format "%s\n%" "$body" "$output_1"]
set body [format "%s\n%" "$body" "----- Show Processes Memory Sorted Holding -----"]
set body [format "%s\n%" "$body" "$output_2"]
if [catch {smtp_send_email $body} result] {
    action_syslog msg "smtp_send_email: $result"
}
}
```

Compose email message with
show output

EEM TCL Script Example

```
#
#----- cli close -----
#
cli_close $cli(fd) $cli(tty_id)

# eeeeeeeeeeeeeeeeeeee End of composite_device_health_memory_threshold.tcl eeeeeeeeeeeeeeeeeeee
```

Service Testing, Verification and Assurance EEM Event Detectors currently available

Cisco IOS CLI

Triggers policies based on commands entered via the CLI.

Cisco IOS Counter

Policies can be triggered based on a change of the designated Cisco IOS counter.

Cisco IOS Redundancy Facility

Provides for detection of hardware and software failures related to the Stateful Switchover service. This ED will trigger policies based on the RF state change. It is also used to initiate switchovers as a result of a policy action.

Cisco IOS Timer Services

Policies can be scheduled to occur at the designated time or interval.

Cisco IOS Watchdog / System Monitor

Triggers policies based on certain conditions relative to a certain Cisco IOS process or subsystem's activity.

EEM Application Specific

Application specific events can be detected or set by a Cisco IOS subsystem or a policy script. This provides the ability for one policy to trigger another policy.

XML RPC (SOAP over SSHv2) (new in EEM 2.4)

Triggers upon receipt of an incoming XML message

Interface Counter

Policies can be triggered based on the specific interface counter; includes thresholds.

Online Insertion and Removal

Triggers policies based on hardware installation and removal activity.

Object Tracking

Triggers policies based on routing protocol events.

SNMP

Triggers policies based on the associated SNMP MIB variable; includes MIB variable threshold setting.

SNMP Proxy (new in EEM 2.4)

Triggers upon receipt of an incoming trap or inform

Syslog

Triggers policies based on the regular expression match of a local Syslog message.

Resource Thresholding (ERM)

Triggers policies based on certain internal resource usage and conditions; interface to Embedded Resource Manager.

Generic Online Diagnostics (GOLD)

Triggers policies based on diagnostic results

"None" ED

Triggers policies by command

EEM Feature/Product Support Matrix

CISCO IOS EMBEDDED EVENT MANAGER										
EEM VERSION - PRODUCT MATRIX										
2/22/08 4:59 PM										
Legend: █ Shipping █ In EFT █ IC █ Planning █ NA										
CISCO ACCESS ROUTERS										
EEM Version	Cisco 800 Series	Cisco 1800 Series	Cisco 2800 Series	Cisco 3800 Series	Cisco 1700 Series	Cisco 2600 Series	Cisco 3500XM Series	Cisco 2691 Series	Cisco 3600 Series	Cisco 3700 Series
1.0										
2.0		12.3(11)T	12.3(11)T	12.3(11)T	12.3(4)T	12.3(4)T	12.3(4)T		12.3(4)T	
2.1		12.3(14)T1	12.3(14)T1	12.3(14)T1	12.3(14)T1	12.3(14)T1	12.3(14)T1	12.3(14)T1	12.3(14)T1	12.3(14)T1
2.1.5		12.4(2)T	12.4(2)T	12.4(2)T	12.4(2)T	12.4(2)T	12.4(2)T	12.4(2)T	12.4(2)T	12.4(2)T
2.2		12.4(11)T	12.4(11)T	12.4(11)T	12.4(11)T	12.4(11)T	12.4(11)T	12.4(11)T	12.4(11)T	12.4(11)T
2.3		12.4(20)T	12.4(20)T	12.4(20)T	12.4(20)T	12.4(20)T	12.4(20)T	12.4(20)T	12.4(20)T	12.4(20)T
2.4		12.5(pl1)T	12.5(pl1)T	12.5(pl1)T	12.5(pl1)T	12.5(pl1)T	12.5(pl1)T	12.5(pl1)T	12.5(pl1)T	12.5(pl1)T
3.0		12.5(pl1)T	12.5(pl1)T	12.5(pl1)T	Planning	Planning	Planning	Planning	Planning	Planning
CISCO 5000 SERIES & UP										
EEM Version	Cisco 7200 Series	Cisco 7301	Cisco 7304	Cisco 7600 Series	Cisco 10000	Cisco 12000 Series	Cisco XR 12000	Cisco CRS-1	Cisco 7500 Series	Cisco 5000 Series
1.0						12.0(20)S			12.0(20)S	
2.0			12.2(27)S9C				See IOS-XR Fault Mgr	See IOS-XR Fault Mgr		12.4R
2.1	12.3(14)T1	12.3(14)T1	12.3(28)S8	12.3(18)SXP5	12.3(28)S8		See IOS-XR Fault Mgr	See IOS-XR Fault Mgr		
2.1.5							See IOS-XR Fault Mgr	See IOS-XR Fault Mgr		
2.2	12.4(2)T	12.4(2)T1					See IOS-XR Fault Mgr	See IOS-XR Fault Mgr		
2.3	12.4(11)T	12.3(23)S8	12.3(33)S8	12.3(33)S8B	12.3(33)S8		See IOS-XR Fault Mgr	See IOS-XR Fault Mgr		12.4(11)T
2.4	12.4(20)T	12.3(28)S8	12.3(28)S8	12.3(28)S8	12.3(28)S8		See IOS-XR Fault Mgr	See IOS-XR Fault Mgr		Planning
3.0	12.5(pl1)T	Planning	Planning	Planning	Planning		See IOS-XR Fault Mgr	See IOS-XR Fault Mgr		Planning
CISCO CATALYST SWITCHES										
EEM Version	Cisco 3750 Switches	Cisco 4500 Switches	Cisco 6500 Switches							
1.0										
2.0										
2.1			106 w/o Modularity 12.3(18)SXP4 w/ Modularity 12.3(18)SXP4							
2.1.5			12.3(18)SXP4							
2.2										
2.3			12.3(33)SXP4							
2.4	12.3(40)S8	Planning	12.3(33)SXP4							
3.0	Planning	Planning	Halfdone							

Includes Futures, Subject to Change; No Commitment Implied

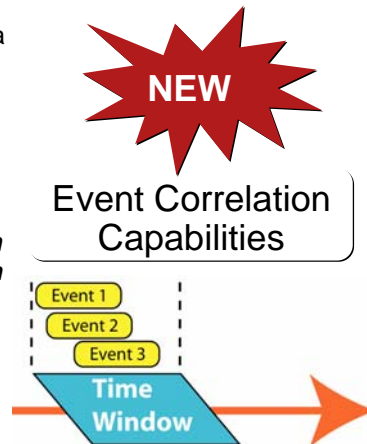
Embedded Event Manager Event Detectors

EEM Event Detector - Release Matrix							
Updated: 02/22/2008							
Event Detector Name	12.0(26) 12.3(4)T	12.3(25)S	12.3(14)T	12.4(2)T	12.3(18)SXP4 (Mod IOS) 12.2(18)SXP5 IOS	12.4(20)T	Description
Application		YES	YES	YES	YES	YES	Custom application events, action script interaction
CLI			YES	YES	YES	YES	Exec command match and run
Counter		YES	YES	YES	YES	YES	Custom counter events
GOLD					YES		Generic Online Diagnostics event detection
Interface		YES	YES	YES	YES	YES	Interface counters and events
Memory Thresholding (Deprecated)							Detect memory resource related events
None (by run command)			YES	YES	YES	YES	
Object Tracking				YES			Integration with Enhanced Object Tracking
OR			YES	YES	YES	YES	Card Online Insertion & Removal detection
Resource Thresholding				YES	YES	YES	Integration with Embedded Resource Manager, supercedes Memory Thresholding ED
RF				YES	YES	YES	IOS Infrastructure Redundancy Facility events
SNMP		YES	YES	YES	YES	YES	Detect MIB Var match and thresholds
SNMP Proxy							Allows device to raise an event on RECEIPT of a trap or inform and execute a policy
Syslog	YES	YES	YES	YES	YES	YES	Reg exp pattern match on emitted syslog messages
Timer		YES	YES	YES	YES	YES	Custom timed events
IOS Watchdog Monitor		YES	YES	YES	YES	YES	IOS scheduler, watchdog events
WDSysMon*					YES		IOS Modularity: System monitor event
XML-RPC (SOAP over SSHv2)						YES	Send a message to invoke a policy from outside the box

1.0 2.0 2.1 2.2 2.1+ 2.4

Service Testing, Verification and Assurance EEM 2.4: Multiple Event Correlation – 1/2

- Previous to EEM v2.4, there was a one-to-one correspondence between a single event and the triggered policy
- In other words, a policy could only be triggered by a single event and any event correlation had to be coded by the user
- Multiple Event Support ushers in an event correlation specification such that multiple events may be considered together to trigger a policy**
- For example:
If (Event 1 OR Event 2) AND Event 3, then
Trigger Policy A



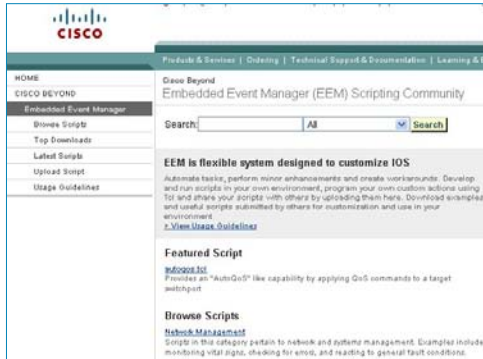
Service Testing, Verification and Assurance EEM 2.4: Multiple Event Correlation – 2/2

- This example results in a Syslog message being issued if either Ethernet1/0 OR Ethernet1/1 changes state up or down
- The **correlate** statement within the trigger block indicates the logic of the relationship between events
- Optional **occurs** clauses define the number of times a specific event must be raised before being used in the correlation or for the number of times the total correlation set must be true before invoking the action (two levels)

```

event manager applet example
event tag e1 syslog pattern ".*UPDOWN.*Ethernet1/0.*"
event tag e2 syslog pattern ".*UPDOWN.*Ethernet1/1.*"
trigger occurs 1
    correlate event e1 or event e2
    attribute e1 occurs 1
    attribute e2 occurs 1
action 1.0 syslog msg "Critical interface status change"
set 2.0 _exit_status 0
    
```

Embedded Event Manager – engage now!



- Device Manageability Instrumentation (DMI): www.cisco.com/go/instrumentation
- Embedded Event Manager: www.cisco.com/go/eem
- EEM Scripting Community: www.cisco.com/go/ciscobeyond (internally: <http://www-win-swpkq.cisco.com/fm/central/index.html>)

Agenda

Theoretical Part

- Introduction & Overview
- Service Planning
- Service Deployment & Activation
- Service Testing, Verification & Assurance
- ➔ Troubleshooting & Optimization
- Summary

Hands-On Lab Part

- Task 1: Monitoring of Device Resources
- Task 2: Visibility into Traffic Flows
- Task 3: Embedded Event Manager
- Task 4: Collecting Baseline Information

```
*** STOP: 0x0000007B (0xF201B84C, 0xC0000034, 0x00000000, 0x00000000)
INACCESSIBLE_BOOT_DEVICE

If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check for viruses on your computer. Remove any newly installed
hard drives or hard drive controllers. Check your hard drive
to make sure it is properly configured and terminated.
Run CHKDSK /F to check for hard drive corruption, and then
restart your computer.

Refer to your Getting Started manual for more information on
troubleshooting Stop errors.
```

POST (Power-On Self-Test) is a great thing

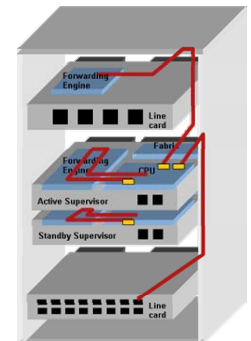
But some errors you prefer to know while the system is still running ...

Manageability Generic OnLine Diagnostics (GOLD)

CLI and scheduling for Functional Runtime Diagnostics

- Bootup Diagnostics (upon bootup and OIR)
- Periodic Health Monitoring (during operation)
- OnDemand (from CLI)
- Scheduled Testing (from CLI)
- Test Types include:
 - Packet switching tests
 - Are supervisor control plane & forwarding plane functioning properly?
 - Is the standby supervisor ready to take over?
 - Are linecards forwarding packets properly?
 - Are all ports working?
 - Is the backplane connection working?
 - Memory Tests
 - Error Correlation Tests
- Complementary to POST

Good Practice: schedule all non-disruptive tests periodically



Manageability

Example: The effect of wear and tear – 1/2

Problem: Repeated insertion and removal of Modules can lead to wear and tear damage on connectors. This in turn can cause failures ... how do you find out during operation, without power-cycling the box ?

Solution: Use GOLD to verify functionality of a mis-behaving module

1) Let's see which GOLD tests are available and scheduled for our Module:

```
Router# show diagnostic content module 3
Module 3:

Diagnostics test suite attributes:
M/C/* - Minimal level test / Complete level test / Not applicable
B/* - Bypass bootup test / Not applicable
P/* - Per port test / Not applicable
D/N/* - Disruptive test / Non-disruptive test/ Not applicable
S/* - Only applicable to standby unit / Not applicable
X/* - Not a health monitoring test / Not applicable
F/* - Fixed monitoring interval test / Not applicable
E/* - Always enabled monitoring test / Not applicable
A/I - Monitoring is active / Monitoring is inactive

Testing Interval
ID  Test Name                Attributes      (day hh:mm:ss.ms)
====
 1) TestScratchRegister -----> *B*N****A    000 00:00:30.00
 2) TestSPRPInbandPing -----> *B*N****A    000 00:00:15.00
 3) TestGBICIntegrity -----> *BPD****I    not configured
:
:
18) TestL3VlanMet -----> M**N****I    not configured
:
:
```

Manageability

Example: The effect of wear and tear – 2/2

2) Now let's run TestL3VlanMet on-demand for Module 3:

```
Router# diagnostic start module 3 test 18
:
00:09:59: %DIAG-SP-3-MINOR: Module 3: Online Diagnostics detected a
Minor Error. Please use 'show diagnostic result <target>' to see
test results.
```

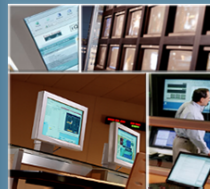
3) Then check the test results:

```
Router# show diagnostic result module 3
Module 3: CEF720 48 port 1000mb SFP SerialNo : xxxxxxxx

Overall Diagnostic Result for Module 3 : MINOR ERROR
Diagnostic level at card bootup: minimal

Test results: (. = Pass, F = Fail, U = Untested)
1) TestTransceiverIntegrity:
Port 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
-----
      U U U U U U U U U U U U U U U U U U U U U U U U U
Port 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
-----
      U U U U U U U U U U U U U U U U U U U U U U U U U
:
:
18) TestL3VlanMet -----> F
```

Reliable Delivery and Filtering of Syslog



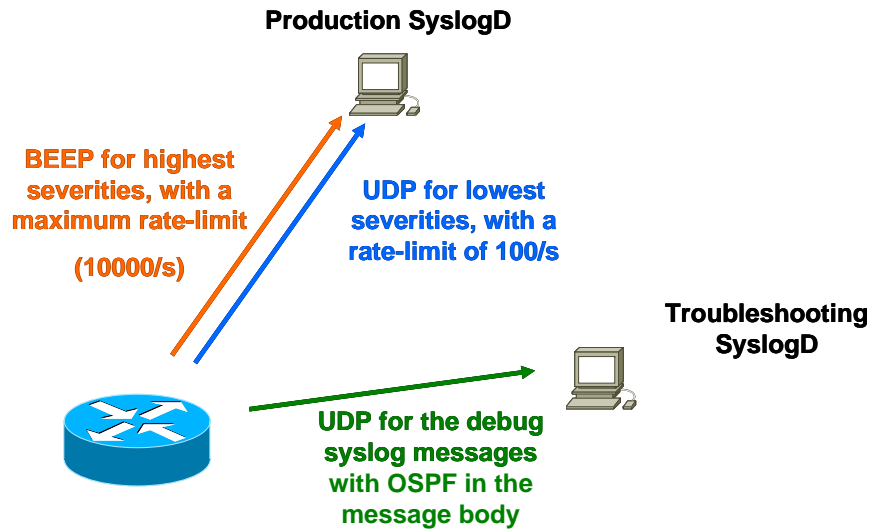
Troubleshooting & Optimization

Reliable Delivery and Filtering of Syslog

- Provides for **reliable** and **secure** delivery for syslog messages using Blocks Extensible Exchange Protocol (BEEP)
 - RFC 3195, "Reliable Delivery for syslog"
- Provides a **filtering** mechanism per syslog session, called a message discriminator
- Provides a **rate-limiter** per syslog session
- Integrated in 12.4(11)T, even if the BEEP framework was supported for quite some time, 12.4(2)T
- Which syslog servers support BEEP?

<http://www.syslog.cc/ietf/rfc3195.html>

Troubleshooting & Optimization Example: Filtering of Syslog – 1/2

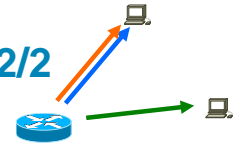


© 2008 Cisco Systems, Inc. All rights reserved.

Cisco Public

81

Troubleshooting & Optimization Example: Filtering of Syslog – 2/2



```
Router(config)# logging discriminator filter1
severity includes 0,1,2,3 rate-limit 10000
Router(config)# logging discriminator filter2
severity includes 4,5,6,7 rate-limit 100
Router(config)# logging discriminator filter3 msg-
body includes debug includes facility OSPF

Router(config)# logging trap debugging

Router(config)# logging host <production> transport
beep discriminator filter1
Router(config)# logging host <production> transport
udp port 1471 discriminator filter2
Router(config)# logging host <troubleshooting>
discriminator filter3
```

© 2008 Cisco Systems, Inc. All rights reserved.

Cisco Public

82

Agenda

Theoretical Part

- Introduction & Overview
- Service Planning
- Service Deployment & Activation
- Service Testing, Verification & Assurance
- Troubleshooting & Optimization

Summary

Hands-On Lab Part

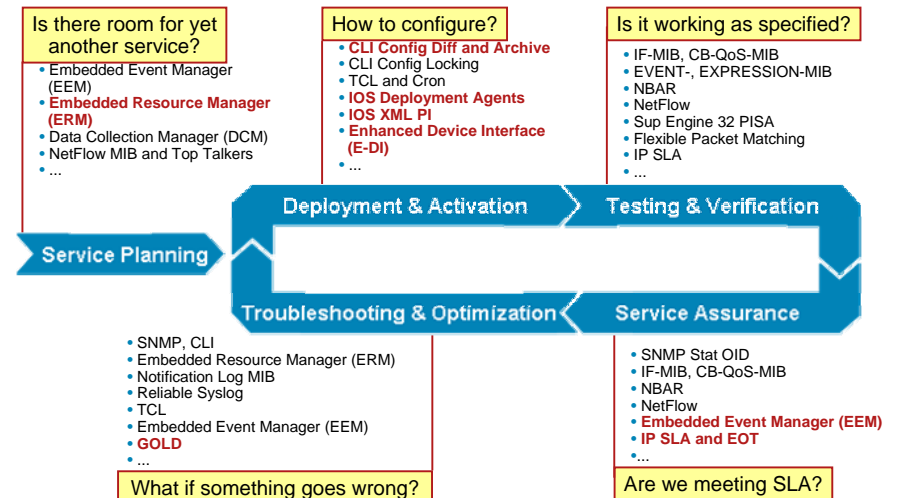
- Task 1: Monitoring of Device Resources
- Task 2: Visibility into Traffic Flows
- Task 3: Embedded Event Manager
- Task 4: Collecting Baseline Information

© 2008 Cisco Systems, Inc. All rights reserved.

Cisco Public

83

Wrap-Up & Close Questions during a Service Life Cycle



© 2008 Cisco Systems, Inc. All rights reserved.

Cisco Public

84

References

Device Manageability Instrumentation (DMI) www.cisco.com/go/instrumentation

- NetFlow: www.cisco.com/go/netflow
- IPSLA (aka SAA, aka RTR): www.cisco.com/go/ipsla
- Enhanced Device Interface (E-DI): www.cisco.com/en/US/products/ps6456/
- Cisco Beyond – EEM Community: www.cisco.com/go/ciscobeyond
- Feature Navigator: www.cisco.com/go/fn
- MIB Locator: www.cisco.com/go/mibs

Cisco Research Collaboration

- Cisco Research: www.cisco.com/go/research
- Conferences and Workshops (CASEMANS, MANWEEK, NMOS/IM, IARIA, SASO, ...)

Monthly Newsletter

- Cisco Network Management Newsletter (email subscription possible):
http://www.cisco.com/external/networkmanagement/cnm-newsletter/April_08.html



Agenda

Theoretical Part

- Introduction & Overview
- Service Planning
- Service Deployment & Activation
- Service Testing, Verification & Assurance
- Troubleshooting & Optimization
- Summary

Hands-On Lab Part

- Task 1: Monitoring of Device Resources
- Task 2: Visibility into Traffic Flows
- Task 3: Embedded Event Manager
- Task 4: Collecting Baseline Information

Lab Starting Instructions

- Point your Browser to the LAB portal
<http://192.168.200.38/emanics/>
- Choose your POD (My POD Number: __ __)
- ensure your browser uses a decent protocol handler for telnet://
(such as <http://www.putty.org/>)

```
[HKEY_CLASSES_ROOT\telnetshell\open\command] @="\"C:\Program Files\PuTTY\putty.exe\" %1
```

- You will configure this entire lab on the **Headquarter Router**
- You can reach your syslog and ftp server from the POD web page
- As an example the EEM „IP_INPUT“ applet and ERM resource policy „IP_INPUT“ have been preconfigured.

Agenda

Theoretical Part

- Introduction & Overview
- Service Planning
- Service Deployment & Activation
- Service Testing, Verification & Assurance
- Troubleshooting & Optimization
- Summary

Hands-On Lab Part

- ➔ **Task 1: Monitoring of Device Resources**
- Task 2: Visibility into Traffic Flows
- Task 3: Embedded Event Manager
- Task 4: Collecting Baseline Information

Task 1 – Proactive Monitoring of System Resources

▪ Background

It is important to reflect any critical status of the network device resources. Proactively getting informed, if any of the pre-defined alarm levels (thresholds) gets violated. It is key for a stable network. Also this capability can be used as an input for service planning tasks.

Consider the following, that unstable current services can give indications on how a network can impact new or future services. This means, if I understand my network behaviour, in terms of what were the triggers of a critical situation, it can be used to avoid this situations for other services as well. This helps to ensure current and future service quality. All of that can be done with Embedded Resource Manager

Let's take an example. A high cpu utilization load can indicate, serious network problems. This could be because a routing process is consuming too much system resources, because of too many routes needed to be processed. Also the type of the packets transferred by the device can bring down the network resources, like many small packets of 64 Bytes. In both examples, the service planning process need to take this into account. Whether it is to ensure and improve the quality of current services, as well as to avoid such a bad behaviour with new services. This can be done by choosing more bandwidth, more scalable network devices or just tune some QoS parameters.

Task 1 – Proactive Monitoring of System Resources

▪ Embedded Resource Manager

The Embedded Resource Manager is a tracking mechanism inside the network resources. It proactively monitors thresholds for system resources, like CPU, buffer, and memory.

ERM provides a mechanism to send notifications whenever the specified threshold values are violated by any resource user (Helps in reducing the CPU, buffer, and memory utilization issues).

Three different alarm levels are available :
Minor, Major, and Critical alarms

http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124cg/hnm_c/ch05/h_rmimg.htm#wp1027177

▪ Task – Configuration of Embedded Resource Manager

In this lab task, you have to configure inside the embedded resource manager a monitor for the system resource cpu. You have to define a critical alarm level with a rising threshold of 30% and a falling threshold of 15% for the total cpu. If this thresholds were reached, the system sends out a syslog message. Later in the next Task, you will use the resource manager notification within the embedded event manager as the input to execute additional commands.

The configuration steps are:

1. create the system global policy
2. define the critical cpu threshold
3. apply the system global policy
4. testing of the defined cpu threshold

Task 1 – Proactive Monitoring of System Resources

Step 1 : create the system global policy

Perform this task to create a system global policy, which monitors the total cpu utilization

1. Change to the Resource Policy configuration modus
Router# configure terminal
Router(config)# resource policy
2. Configure a global Resource Policy
Router(config-erm)# policy CPU_MONITOR global
3. Configure System Level Resource Owners. Here it is the whole system.
Router(config-erm-policy)# system
4. Configure what resource will be monitored. Here it is the total cpu utilization
Router(config-policy-node)# cpu total

Task 1 – Proactive Monitoring of System Resources

Step 2 – Define the critical cpu threshold

Three alarm levels are available. Here we are going to use only the critical alarm level (threshold).

1. Configure the critical level threshold of percentage cpu utilization. Rising threshold for a 5 second interval is 30% and falling threshold for a 5 second interval is 15%

```
Router(config-owner-cpu)# critical rising 5 interval 5 falling 1 interval 5
```

2. Leave the resource policy configuration modus

```
Router(config-owner-cpu)# exit
Router(config-policy-node)# exit
Router(config-erm-policy)# exit
```

Step 3 – Apply the system global policy

1. Configure resource user and apply the defined global policy

```
Router(config-erm)# user global CPU_MONITOR
```

2. Leave the resource manager configuration modus

```
Router(config-erm)# exit
```

3. Leave the global configuration modus

```
Router(config)# exit
```

© 2008 Cisco Systems, Inc. All rights reserved. Cisco Public

93

Task 1 – Proactive Monitoring of System Resources

Step 4 – Testing of the defined cpu threshold

1. Start an extended ping to the predefined loopback interface 0. Use a repeat count of 1000 and a datagram size of 10000

```
Router# ping
Protocol [ip]: <enter>
Target IP address: 100.101.0.1
Repeat count [5]: 2000
Datagram size [100]: 18000
Timeout in seconds [2]: <enter>
Extended commands [n]: <enter>
Sweep range of sizes [n]: <enter>
```

2. You should see two syslog messages, indicating the raising and the falling thresholds

```
Jan 21 10:32:43.284: %SYS-4-CPURESISING: System is seeing global cpu util 45% at total level more than the configured critical limit 30%
```

```
Jan 21 10:33:03.286: %SYS-6-CPURESFALLING: System is no longer seeing global high cpu at total level for the configured major limit 15%, current value 8%
```

© 2008 Cisco Systems, Inc. All rights reserved. Cisco Public

94

Task 1 – Proactive Monitoring of System Resources

Useful show commands

1. Use this command to view a brief CPU report details for event tracing for a networking device:

```
show monitor event-trace cpu-report handle 1
```

2. Use this command to view an extended CPU load report:

```
show processes cpu extended
```

3. Use this command to display the resource details:

```
show resource all
```

4. Use this command to display the relationship details between different resource owners:

```
show resource user all brief
```

© 2008 Cisco Systems, Inc. All rights reserved. Cisco Public

95

Task 1 – Proactive Monitoring of System Resources

Useful debug commands

```
Router#debug resource policy notification
```

When a threshold is violated:

```
*Mar  3 09:50:44.081: Owner: 'memory' initiated a notification:
*Mar  3 09:50:44.081: %SYS-4-RESMEMEXCEED: Resource user usrr1 has exceeded the Major memory threshold
Pool: Processor Used: 42932864 Threshold :42932860
*Mar  3 09:50:46.081: Notification from Owner: 'memory' is dispatched for User: 'usrr1' (ID: 0x10000B9)
*Mar  3 09:50:46.081: %SYS-4-RESMEMEXCEED: Resource user usrr1 has exceeded the Major memory threshold
Pool: Processor Used: 42932864 Threshold :42932860
```

© 2008 Cisco Systems, Inc. All rights reserved. Cisco Public

96

Agenda

Theoretical Part

- Introduction & Overview
- Service Planning
- Service Deployment & Activation
- Service Testing, Verification & Assurance
- Troubleshooting & Optimization
- Summary

Hands-On Lab Part

- Task 1: Monitoring of Device Resources
- ➔ Task 2: Visibility into Traffic Flows
- Task 3: Embedded Event Manager
- Task 4: Collecting Baseline Information

Task 2 – Getting visibility on the network

▪ Background

Visibility in the network is a key function of understanding the behavior of applications and services. It can be used to proactively find out where the network usage evolves, in terms of where are potential bottlenecks and which users and services are using the network in which way.

Visibility must be generated on demand, as well as a function for baselining and trending on network usage.

Netflow is the tool for getting visibility on the network and taking the collected data as the input for service planning.

Traffic analysis—Consulting the data retrieved from the NetFlow MIB and Top Talkers feature can assist you in general traffic study and planning for your network.

Task 2 – Getting visibility on the network

▪ NetFlow

Netflow is a technology that provides highly granular per-flow statistics on traffic in a Cisco router. It has been used for many applications, including TE, usage-based billing, DoS monitoring.

▪ Netflow MIB

Netflow MIB is an alternate method of handling netflow data, without sending all the flows via the network. The collected netflow data can be accessed internally on the router for further investigation. Netflow MIB can be used for configuration Netflow parameters as well as for monitoring the collected data. With the netflow MIB you can collect all netflow information in a special cache on the router and giving access to it, without exporting all the flows over the network

▪ Netflow Top-Talkers

The NetFlow Top Talkers feature uses NetFlow functionality to obtain information regarding heaviest traffic patterns and most-used applications in the network.

Task 2 – Getting visibility on the network

▪ Task – Configuration of Netflow

In this lab task, you have to enable netflow on Ethernet0/1 of podN-hq and configure the netflow top-talkers feature.

You will play around with netflow show commands to understand the traffic usage on the HQ router.

Later in the next Task, you will use the netflow MIB by retrieving the „number of packets on device“ MIB object within the embedded event manager as an execute command. The output will send via syslog to the central syslog server.

The configuration steps are:

1. Enable netflow
2. Enable netflow top-talkers
3. Generating traffic
4. Display netflow statistics

Task 2 – Getting visibility on the network

Step 1 : Enable netflow

Perform this task to enable netflow on the Interface Ethernet 0/1 and increase the timeout value for inactive flows

1. Change to the interface configuration modus
Router# configure terminal
Router(config)# interface ethernet0/3
2. Configure ingress and egress netflow
Router(config-if)# ip flow ingress
Router(config-if)# ip flow egress
3. Make the same netflow configurations for interface ethernet 0/0, like you did in the two netflow configuration steps before
4. Leave the interface configuration modus
Router(config-if)# exit
5. Configures the number of seconds that an inactive entry will stay in the main cache before it times out. The range is from 10 to 600 seconds. Here we want you to have each entry staying at maximum
Router(config)# ip flow-cache timeout inactive 600

Task 2 – Getting visibility on the network

Step 1 : Enable netflow top-talkers

Perform this task to enable netflow top-talkers

- Change to the netflow top-talkers configuration modus
Router(config)# ip flow-top-talkers
- Specifies the 10 as the number of top talkers that will be retrieved by a NetFlow top talkers query.
Router(config-flow-top-talkers)# top 10
- Specifies the sort criterion for the top talkers.
Router(config-flow-top-talkers)# sort-by packets
- Leave the configuration modus
Router(config-flow-top-talkers)# exit
Router(config)# exit

Task 2 – Getting visibility on the network

Step 3 – Generating traffic

Telnet to OC1 router (IP is 100.101.1.2) and ping the netflow enabled interface on the HQ router (IP is 100.101.1.1)

Exit back to the HQ router

Step 4 – Display netflow statistics

1. Verify that the NetFlow MIB and Top Talkers feature is operational.
Router# show ip flow top-talkers
2. Show the netflow statistics on the router
Router# show ip cache flow

Useful debug commands

1. Debug flow cache allocation events
Router# debug ip flow cache
2. Debug flow top talkers
Router# debug ip flow top

Agenda

Theoretical Part

- Introduction & Overview
- Service Planning
- Service Deployment & Activation
- Service Testing, Verification & Assurance
- Troubleshooting & Optimization
- Summary

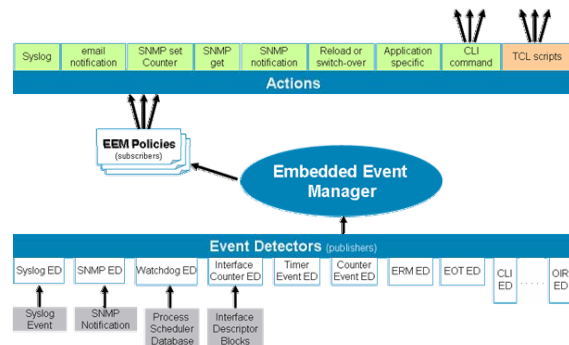
Hands-On Lab Part

- Task 1: Monitoring of Device Resources
- Task 2: Visibility into Traffic Flows
- ➔ Task 3: Embedded Event Manager
- Task 4: Collecting Baseline Information

Task 3 – Embedded Event Manager

Embedded Event Manager (EEM) is a distributed, scalable, and customized approach to event detection and recovery offered directly in a Cisco IOS device. EEM offers the ability to monitor events and take informational or corrective action when the monitored events occur or when a threshold is reached. An EEM policy is an entity that defines an event and the actions to be taken when that event occurs.

http://www.cisco.com/en/US/products/ps6350/products_configuration_guide_chapter09186a008045578a.html



© 2008 Cisco Systems, Inc. All rights reserved.

Cisco Public

105

Task 3 – Built-in event manager

Task – Configuration of Embedded Event Manager

In this lab, you will learn how to configure an applet triggered by a syslog message. The „HIGH CPU“ syslog message which will be used as the event trigger in EEM, is sent from Embedded Resource Manager (see Task „Proactive Monitoring of system resources“). Based on this syslog message, you will take several actions which are related to your service planning. EEM will query for several resource data in your device like „number of packets“, „top talkers“ and send these data as SNMP trap („number of packets“) and syslog message („top talkers“) to the network admin.

The configuration steps are:

- 1) Configure EEM applet „HIGH CPU“
- 2) Configure EEM event criteria
- 3) Take EEM action „SNMP query“: Netflow-MIB
- 4) Take EEM action „CLI command“: Show command
- 5) Take EEM action „syslog“ and to network admin
- 6) Take EEM action „SNMP trap“ and to network admin
- 7) Test EEM actions

© 2008 Cisco Systems, Inc. All rights reserved.

Cisco Public

106

Task 3 – Built-in event manager

Step 1 - Configure EEM applet „HIGH CPU“

Perform this task to register an applet with Embedded Event Manager and to define the EEM applet using event applet and action applet commands. Registers the applet with the Embedded Event Manager (EEM) and enters applet configuration mode.

```
Router# configuration terminal
Router(config)# event manager applet CPU_MONITOR
```

Step 2 - Configure EEM event criteria

Specifies the event criteria that cause the EEM applet to run. We are referring here to ERM policy „CPU_MONITOR“

```
Router(config-applet)# event resource policy "CPU_MONITOR"
```

© 2008 Cisco Systems, Inc. All rights reserved.

Cisco Public

107

Task 3 – Built-in event manager

Step 3 - Take EEM action „SNMP query“: Netflow-MIB

Specifies the action to be taken when an EEM applet is triggered. In this example we will query the netflow MIB for number of packets on the device.

```
Router(config-applet)# action 1.0 info type snmp oid
"1.3.6.1.4.1.9.9.387.1.5.3.1.3" get-type next
```

Step 4 - Take EEM action „CLI command“: Show command

Specifies the action to be taken when an EEM applet is triggered. In this example we run the CLI command: „show ip flow top-talkers“.

```
Router(config-applet)# action 2.0 cli command "show ip flow top-talkers"
```

© 2008 Cisco Systems, Inc. All rights reserved.

Cisco Public

108

Task 3 – Built-in event manager

Step 5 - Take EEM action „syslog“ and send to network admin

Specifies the action to be taken when an EEM applet is triggered. In this example we create a syslog message and use "show ip flow top-talkers" as input.

```
Router(config-applet)# action 3.0 syslog msg "NETFLOW TOP-TALKERS"
= $_cli_result"
```

Step 6 - Take EEM action „SNMP trap“ and to network admin

Specifies the action to be taken when an EEM applet is triggered. In this example, we create a SNMP trap and use netflow MIB for number of packets on the device as input.

Instead of using a SNMP Trap, you can use a syslog message

```
Router(config-applet)# action 4.0 snmp-trap strdata "ERM : Netflow
packets: $_info_snmp_value <--> CPU rising / falling Threshold
$_resource_configured_threshold value = $_resource_current_value"
```

Task 3 – Built-in event manager

Step 7 – Testing of the defined cpu threshold

1. Start an extended ping to the predefined loopback interface 0. Use a repeat count of 1000 and a datagram size of 10000

```
Router# ping
Protocol [ip]: <enter>
Target IP address: 100.101.0.1
Repeat count [5]: 2000
Datagram size [100]: 18000
Timeout in seconds [2]: <enter>
Extended commands [n]: <enter>
Sweep range of sizes [n]: <enter>
```

Task 3 – Built-in event manager

Step 8 - Test EEM actions

- Check your syslog server if the syslog message created by EEM is received.
- On the syslog server you will see the netflow-top-talkers output syslog message.
- A SNMP-trap is sent, but you won't see any in this lab. There is no SNMP-server available. Instead you can use a syslog message with the same information as in the SNMP Trap.

Task 3 – Built-in event manager

Step 8 – Useful commands

1. Displays the EEM policies that are currently registered.
show event manager policy registered
2. Use this command to display detailed information about each EEM event
show event manager history events detailed

Agenda

Theoretical Part

- Introduction & Overview
- Service Planning
- Service Deployment & Activation
- Service Testing, Verification & Assurance
- Troubleshooting & Optimization
- Summary

Hands-On Lab Part

- Task 1: Monitoring of Device Resources
- Task 2: Visibility into Traffic Flows
- Task 3: Embedded Event Manager
- ➔ Task 4: Collecting Baseline Information

Task 4 – Collecting baseline information

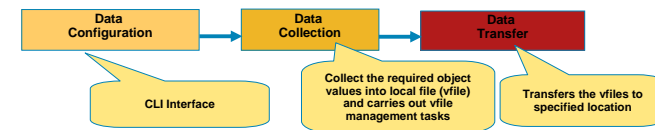
Background

Collecting network data has traditionally been performed by external applications. The data is polled with SNMP through the whole network which causes additional load on the network. Instead of polling data for all the instances and sending each packet through the network, it would be more efficient when network admin can define his collection once on the devices, the data is collected locally and transferred. The following example shows the efficiency:

Average Response time for getting 5 MIB objects for all predefined interfaces:

SNMP polling model: 58 Seconds under no load condition, with effective SNMP ENGINE time of 15 seconds.

Data Collection Manager: Effective response time is 18 seconds approximately; 7.5 seconds for polling the data and 11.5 seconds for transfer the collected data.



Task 4 – Collecting baseline information

Data Collection Manager

This feature provides the ability to periodically transfer selected MIB data from Cisco IOS-based devices to specified Network Management Stations (NMS). Using the CLI, data from multiple MIBs can be grouped into lists, and a polling interval (frequency of data collection) can be configured. All the MIB objects in a list are periodically polled using this specified interval. The collected data from the lists can then be transferred to a specified NMS at a user-specified transfer interval (frequency of data transfer) using TFTP, RCP, or FTP.

To configure the Periodic MIB Data Collection and Transfer Mechanism, you must understand the following concepts:

- SNMP Objects and Instances
- Bulk Statistics Object Lists
- Bulk Statistics Schemas
- Bulk Statistics Transfer Options

http://www.cisco.com/en/US/products/sw/iosswrel/ps1829/products_feature_guide09186a008014c77d.html

Task 4 – Collecting baseline information

Task – Data Collection Manager

In this lab, you will learn how to configure Data Collection Manager and how you can collect bulk data to baseline your service and send this data through the network to your network admin.

To plan for current and new services the network admin needs to get real time and baselining data from the network. With the collected data the network admin can see a trend in his network which supports to plan his new service and monitor if the trend impacts his current service.

Data Collection Manager will provide service planning data for baselining and send the collected data to an external server.

In this lab you will configure data collection for IfInErrors, IfInOctets, IfOutErrors, IfOutOctets and cnfPSPackets (NetflowMIB – Number of packets on device) and define for each MIB (IF-MIB and Netflow-MIB) and object list in order to get these data and send via FTP to an external server.

The configuration steps are:

1. Configuring a Bulk Statistics Object List
2. Configuring a Bulk Statistics Schema
3. Configuring a Bulk Statistics Transfer Options
4. Monitoring and Troubleshooting
5. Check the file on FTP-server

Task 4 – Collecting baseline information

Step 1 - Configuring a Bulk Statistics Object List

The first step in configuring the Periodic MIB Data Collection and Transfer Mechanism is to configure one or more object lists.

All the objects in a bulk statistics object list have to be indexed by the same MIB index. However, the objects in the object list do not need to belong to the same MIB or MIB table. When specifying an object name instead of an OID (using the add command), only object names from the Interfaces MIB (IF-MIB.my), Cisco Committed Access Rate MIB (CISCO-CAR-MIB.my) and the MPLS Traffic Engineering MIB (MPLS-TE-MIB.my) may be used.

In this lab you will use router podN-hq.

```
Router# configuration terminal
Router(config)# snmp mib bulkstat object-list IF-MIB-List
Router(config-bulk-objects)# add ifInErrors
Router(config-bulk-objects)# add ifOutErrors
Router(config-bulk-objects)# add ifInOctet
Router(config-bulk-objects)# add ifOutOctet
Router(config-bulk-objects)# exit
Router(config)# snmp mib bulkstat object-list NETFLOW-MIB-List
Router(config-bulk-objects)# add 1.3.6.1.4.1.9.9.387.1.5.3.1.3
Router(config-bulk-objects)# exit
```

cnfPSPackets

Task 4 – Collecting baseline information

Step 2 - Configuring a Bulk Statistics Schema

The next step in configuring Periodic MIB Data Collection and Transfer is to configure one or more schemas.

```
Router(config)# snmp mib bulkstat schema IF-MIB-Schema
Router(config-bulk-sc)# object-list IF-MIB-List
Router(config-bulk-sc)# instance exact interface ethernet0/3
Router(config-bulk-sc)# poll-interval 1
Router(config-bulk-sc)# exit
Router(config)# snmp mib bulkstat schema NETFLOW-MIB-Schema
Router(config-bulk-sc)# object-list NETFLOW-MIB-List
Router(config-bulk-sc)# instance exact interface ethernet0/3
Router(config-bulk-sc)# poll-interval 1
Router(config-bulk-sc)# exit
```

Task 4 – Collecting baseline information

Step 3 - Configuring a Bulk Statistics Transfer Options

The final step in configuring the Periodic MIB Data Collection and Transfer Mechanism is to configure the transfer options. The collected MIB data are kept in a local file-like entity called a VFile (virtual file, referred to as a bulk statistics file in this document). This file can be transferred to a remote network management station (NMS) at intervals you specify.

```
Router(config)# snmp mib bulkstat transfer 1
Router(config-bulk-tr)# buffersize 3500
Router(config-bulk-tr)# schema IF-MIB-Schema
Router(config-bulk-tr)# schema NETFLOW-MIB-Schema
Router(config-bulk-tr)# transfer-interval 2
Router(config-bulk-tr)# url primary x
Router(config-bulk-tr)# retry 3
Router(config-bulk-tr)# retain 20
Router(config-bulk-tr)# enable
Router(config-bulk-tr)# exit
Router(config)# exit
```

format: ftp://user:password@host/dir/file
ie.:
ftp://pod21:cisco@10.31.220.10/task4.txt

Task 4 – Collecting baseline information

Step 5 - Monitoring and Troubleshooting

The show command for this feature lists all bulk statistics virtual files (VFiles) on the system that have finished collecting data. (Data files that are not complete are not displayed.)

The output lists all of the completed local bulk statistics files, the remaining time left before the bulk statistics file is deleted (remaining retention period), and the state of the bulk statistics file.

Router# show snmp mib bulkstat transfer

Enables standard debugging output for the Bulk Statistics feature. Debugging output includes messages about the creation, transfer, and deletion of bulk statistics files.

Router# debug snmp bulkstat

Task 4 – Collecting baseline information

Step 6 - Check the file on FTP-server

The file should look like this:

```
Schema-def IF-MIB-Schema "%u, %s, %s, %u, %u, %u, %u"
    epochtime ifDescr instanceOID ifInErrors ifInOctets ifOutErrors ifOutOctets
Schema-def NETFLOW-MIB-Schema "%u, %s, %s, %llu"
    epochtime ifDescr instanceOID 1.3.6.1.4.1.9.9.387.1.5.3.1.3
Schema-def GLOBAL "%s, %s, %u, %u, %u, %u, %u"
    hostname date timeofday sysuptime cpu5min cpu1min cpu5sec
IF-MIB-Schema: 1198751401, FastEthernet0/1, .2, 0, 361296, 0, 486673
NETFLOW-MIB-Schema: 1198751401, FastEthernet0/1, .2, 0
Global: DMI-c1841, 20071227, 103100, 10069, 0%, 1%, 0%
```



Lab Sample Solution – 1/4

```
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname HeadQuater
!
boot-start-marker
boot-end-marker
!
enable password cisco
!
no aaa new-model
!
resource policy
policy CPU_MONITOR global
system
cpu total
critical rising 5 interval 5 falling 1 interval 5
!
!
policy IP_INPUT_MONITOR type iosprocess
system
cpu process
critical rising 5 interval 5 falling 1 interval 5
!
!
user global CPU_MONITOR
!
user "IP Input" iosprocess IP_INPUT_MONITOR
!
!
clock timezone CET 0
no ip domain lookup
!
```

```
!
multilink bundle-name authenticated
no mpls ip
!
!
archive
log config
hidekeys
!!
interface Loopback0
ip address 100.101.0.1 255.255.255.0
!
interface Loopback1
ip address 192.168.1.1 255.255.255.0
!
interface Ethernet0/0
description link Headquarter <--> OC-1
ip address 100.101.1.1 255.255.255.0
ip flow ingress
ip flow egress
!
interface Ethernet0/1
description link Headquarter <--> OC-2
ip address 100.101.4.1 255.255.255.0
!
interface Ethernet0/2
no ip address
shutdown
!
interface Ethernet0/3
description HD<-->Gateway
ip address 192.168.101.1 255.255.255.0
ip flow ingress
ip flow egress
```

Lab Sample Solution – 2/4

```
!
interface Ethernet1/0
no ip address
shutdown
!
interface Ethernet1/1
no ip address
shutdown
!
interface Ethernet1/2
no ip address
shutdown
!
interface Ethernet1/3
no ip address
shutdown
!
interface Serial2/0
description link Headquarter <--> Prem01
bandwidth 2000
ip address 100.101.2.1 255.255.255.0
serial restart-delay 0
!
interface Serial2/1
description link Headquarter <--> Prem99
bandwidth 2000
ip address 100.101.99.1 255.255.255.0
serial restart-delay 0
!
interface Serial2/2
no ip address
serial restart-delay 0
!
interface Serial2/3
no ip address
shutdown
serial restart-delay 0
!
```

```
interface Serial3/0
no ip address
shutdown
serial restart-delay 0
!
interface Serial3/1
no ip address
shutdown
serial restart-delay 0
!
interface Serial3/2
no ip address
shutdown
serial restart-delay 0
!
interface Serial3/3
no ip address
shutdown
serial restart-delay 0
!
router ospf 1
log-adjacency-changes
network 100.101.1.0 0.0.0.255 area 0
network 100.101.2.0 0.0.0.255 area 0
network 100.101.4.0 0.0.0.255 area 0
network 100.101.99.0 0.0.0.255 area 0
network 101.101.1.0 0.0.0.255 area 0
network 101.101.2.0 0.0.0.255 area 0
network 101.101.3.0 0.0.0.255 area 0
network 101.101.4.0 0.0.0.255 area 0
network 101.101.99.0 0.0.0.255 area 0
!
router rip
version 2
network 10.0.0.0
network 192.168.1.0
network 192.168.101.0
no auto-summary
```

Lab Sample Solution – 3/4

```
ip flow-top-talkers
top 10
sort-by packets
!
no ip http server
no ip http secure-server
!
logging trap notifications
logging source-interface Loopback1
logging 10.31.220.10
snmp-server community premspublic RO
snmp-server community premsprivate RW
snmp-server host 10.1.1.10 public
!
snmp mib bulkstat object-list IF-MIB-List
add ifInErrors
add ifInOctets
add ifOutErrors
add ifOutOctets
snmp mib bulkstat object-list NETFLOW-MIB-List
add 1.3.6.1.4.1.9.9.387.1.5.3.1.3
snmp mib bulkstat schema IF-MIB-Schema
object-list IF-MIB-List
poll-interval 1
instance exact interface Ethernet0/3
```

```
snmp mib bulkstat schema NETFLOW-MIB-Schema
object-list NETFLOW-MIB-List
poll-interval 1
instance exact interface Ethernet0/3
snmp mib bulkstat transfer 1
schema IF-MIB-Schema
schema NETFLOW-MIB-Schema
format schemaASCII
transfer-interval 2
retain 30
buffer-size 3500
enable
snmp mib bulkstat transfer ftp://pod11:cisco@10.31.220.10/task4.txt
format schemaASCII
!
control-plane
!
!
line con 0
logging synchronous
line aux 0
line vty 0 4
password cisco
login
!
```

Lab Sample Solution – 4/4

```
event manager applet CPU_MONITOR
event resource policy "CPU_MONITOR"
action 1.0 info type snmp oid "1.3.6.1.4.1.9.9.387.1.5.3.1.3" get-type next
action 2.0 cli command "show ip flow top-talkers"
action 3.0 syslog msg "NETFLOW TOP TALKERS = $_cli_result"
action 4.0 snmp-trap strdata "ERM: netflow packets: $_info_snmp_value <-> cpu rising / falling threshold
$_resource_configured_threshold value= $_resource_current_value"
event manager applet IP_INPUT_MONITOR
event resource policy "IP_INPUT_MONITOR"
action 1.0 syslog priority critical msg "ERM : rising / falling CPU PROC : IP Input TH= $_resource_configured_threshold"
action 2.0 info type snmp oid "1.3.6.1.4.1.9.9.387.1.5.3.1.3" get-type next
action 3.0 cli command "show ip flow top-talkers"
action 4.0 syslog msg "NETFLOW TOP TALKERS = $_cli_result"
action 5.0 snmp-trap strdata "ERM : Netflow packets: $_info_snmp_value <-> CPU PROC : IP Input rising / falling !
Threshold $_resource_configured_threshold with value $_resource_current_value"
!
end
```



June 2-6, 2008
University of Zurich, Switzerland



University of Zurich

