Cisco 5921 Embedded Services Router

Example Installation on “Reference System”

Revision 5: March 2015

Cisco 5921 Platforms

1. **Linux on Windows Machine (bare metal) (slide #3)**

- Processor: Intel x86 architecture, such as Intel Atom, Core i3/i5/i7
- RAM: 512 MB minimum
- Hard Drive: 300 MB minimum
- OS: glibc-compiled Linux operating system (kernel 2.6.32 or higher)
- License: Cisco Software License (*.lic) or Smart License (command Line)
- Software: Cisco IOS 15.2(4)GC or 15.5(1)T or later

2. **Linux on Virtual Machine (ESXi, KVM) (slide#4)**

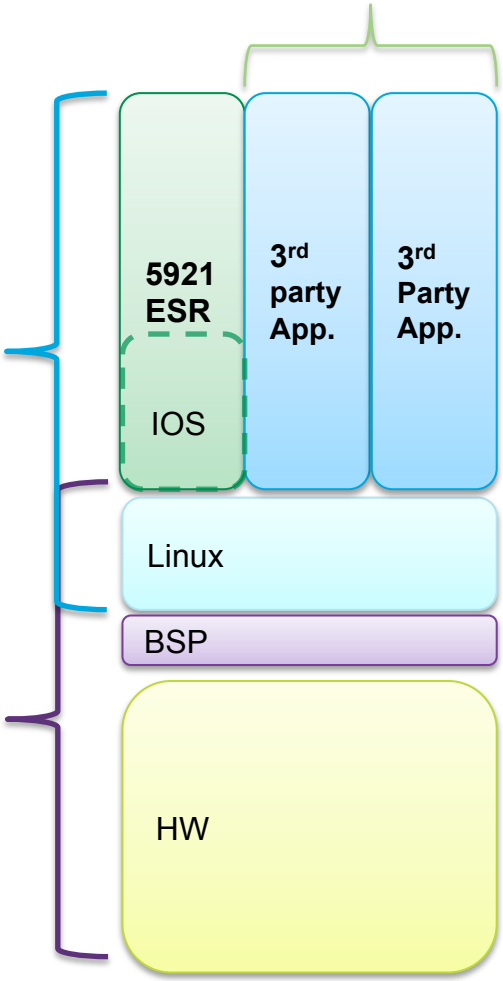
- Platform: Intel x86 platforms supported by ESXi, KVM
- ESXi: Vmware vSphere 5 ESXi Installation Guide
- KVM: http://www.linux-kvm.org/page/Main_Page
- Software: Cisco IOS 15.5(1)T or newer required
- License: Smart License (command line)

IOS Revision and Licenses

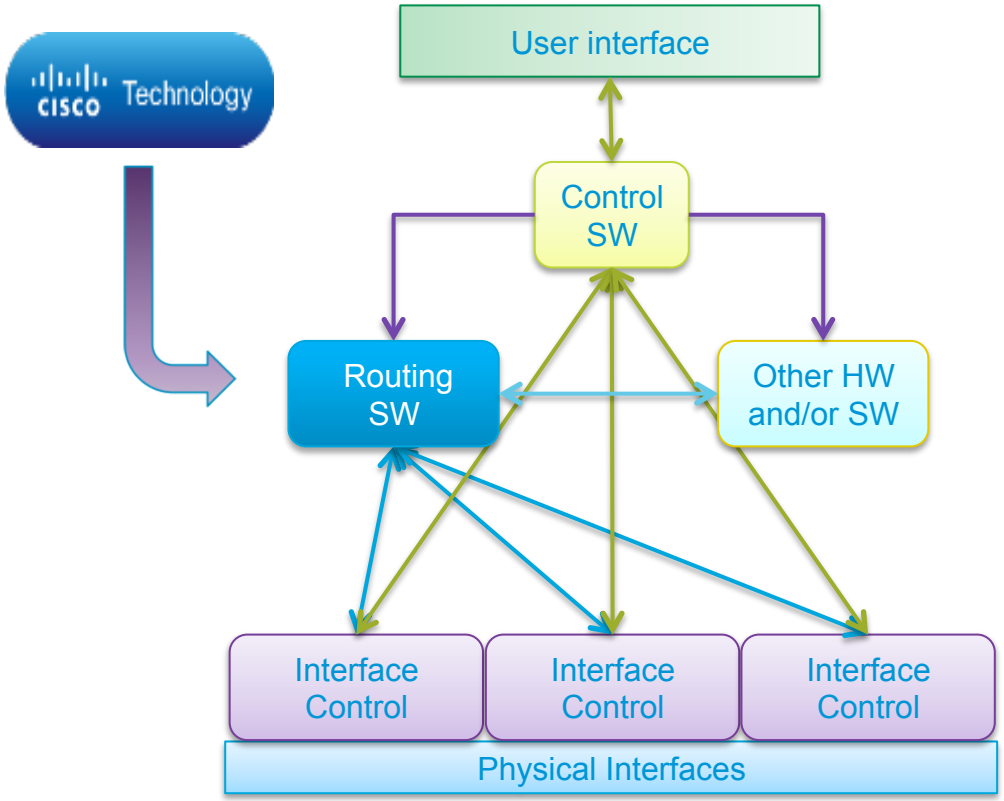
Operational System	Linux	Hypervisor VMware
Processor	Intel x86 (bare metal)	Intel x86, ESXi or KVM
15.2(4)GC	Yes	No
15.5(1)T	Yes	Yes
CLS license *.lic	Yes, if is 15.2(4)GC	No
SL: Smart License (command Line)	Yes, if is 15.5(1)T	Yes, if is 15.5(1)T

License	**Supported Release**	**Supported PI**
CSL *.lic	15.4(2)T	PI24
Smart License	15.5(1)T	PI26

5921 IOS Linux System Architecture

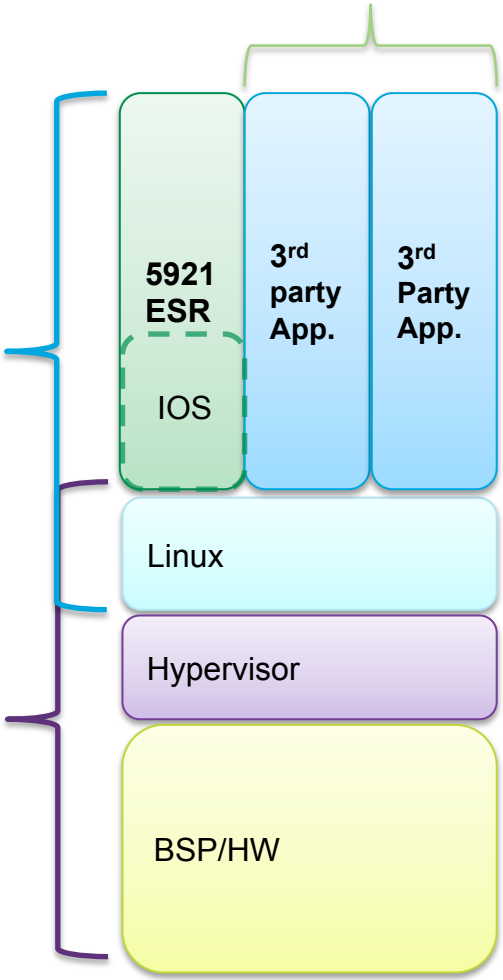


Typical Use Case

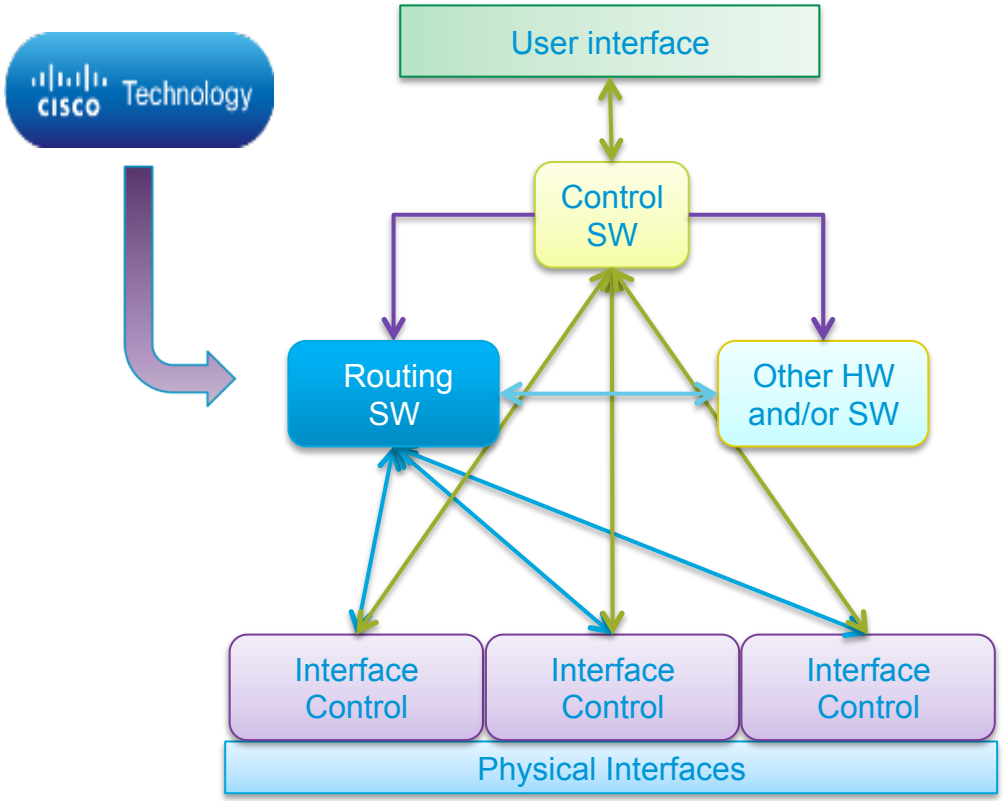


IOS as a Linux Application - Running on 3rd Party HW

5921 IOS Linux/VM Architecture



Typical Use Case



IOS as a Linux Application - Running on VM/3rd Party HW

Example integration on "Reference System"

If use linux on VM as platform please refer to slide#6
If install on bare metal using Linux system, please refer to slide#7.

5921 with Linux on VMware

- Install CentOS on VMware
- How to VMWare with CentOS 6.x
- IOS release: minimum Cisco5921ESR1551T [15.5(1)T]
- Licensing: Smart Licensing (via Command Line)

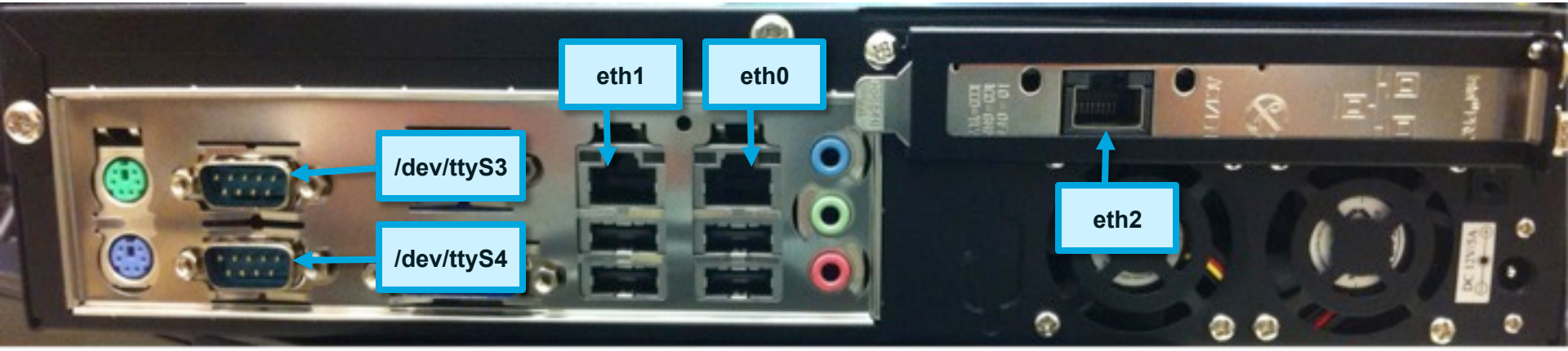
(Reference system: CentOS)

Reference System Platform
Linux System

- All Integration issues must be reproduced on the reference platform.
- Reference OS
  - CentOS Linux 6.6 32 bit
  - Motherboard: Intel D2500CC mini-itx (form factor) http://www.intel.com/content/www/us/en/motherboards/desktop-motherboards/desktop-board-d2500cc.html
- Memory
  - 4 gigabytes, DDR3 (2x2gig), 1333mhz, so-dimm
  - Should be 2x2gig modules to ensure dual-channel memory operation
- Extra NIC
  - Any Intel based PCI NIC should be sufficient, Cisco used Intel PRO/1000 GT during development
- Additional parts, such as enclosure, hard drive, optical drive, are up to the discretion of the system integrator
- IOS Software: 15.5(x)T, with Smart Licensing

Reference Platform – Rear Panel

- Reference hardware when running CentOS 6.5 will (typically) number connectors as follows:



- During internal test, the NICs were configured as follows
  - eth0 and eth1 are used by the Cisco 5921
  - eth2 is used for Linux management, it uses DHCP by default
- Customer may or may not use the interfaces in this manner

Assumptions

- Cisco 5921 .tar file has been copied to USB flash drive
- Ethernet ports eth0 and eth1 on reference system will be used by IOS
- IOS to be installed in the default location, **/opt/cisco/c5921**
- Raw sockets to be used for communication to outside
- tap0 used for Linux <-> IOS communication

Installing CentOS 6.6 32 bit

- Download CentOS 6.6 32 bit from the "minimal" .iso image

  http://mirror.cs.vt.edu/pub/CentOS/6.5/isos/i386/CentOS-6.5-i386-minimal.iso

- Burn ISO to a blank CD
- Boot reference hardware from newly created CD
- Select “Install or upgrade an existing system”
- Skip the media check
- When installer loads (takes a few minutes), click "Next"
- Select "English" for the language
- Select "U.S. English" for the keyboard
- Select "Basic Storage Devices"
- If prompted, select "Yes, discard any data"
- Enter a hostname for the system, "centos" was used during test
- Select the correct timezone
- Enter the root password, "cisco123” is what we typically use in lab
  - Select "Use anyway" warning about bad password
- Select "Use All Space" for installation type
- Select "Write changes to disk" to save configuration to disk
- Wait for install to complete, click "Reboot" when finished

Prepare CentOS 6.6 32 bit

- Once install is complete, login via root account
- Configure eth2 for DHCP

  Edit eth2 network configuration file:

```
vi /etc/sysconfig/network-scripts/ifcfg-eth2
```

  Populate file as follows:

```
DEVICE="eth2"
BOOTPROTO="dhcp"
NM_CONTROLLED="no"
ONBOOT="yes"
TYPE="Ethernet"
UUID="<some value>"
```

  Insert Internet-connected cable into eth2 and restart networking to obtain an IP address:

```
service network restart
```

Prepare continued…

- Download latest CentOS updates

```
yum update -y
```

- When updates are complete, reboot
- Install additional dependencies (needed for TAP interfaces and viewing man pages)

```
yum install -y tunctl man
```

- Reboot when compete

Prepare continued…

- Configure eth0 and eth1 interfaces

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

- Populate file as follows:

```
DEVICE=eth0
BOOTPROTO=none
NM_CONTROLLED=no
ONBOOT=yes
TYPE=Ethernet
UUID=<some value>
HWADDR=<some value>
```

- Be sure to repeat process for eth1
- When finished, restart networking to bring the interfaces online

```
service network restart
```

Prepare continued…

- The steps on this page are optional. Only perform these steps if you desire an internal connection between Linux and IOS.
- Configure tap0 for Linux <-> IOS communication

```
vi /etc/sysconfig/network-scripts/ifcfg-tap0
```

- Populate file as follows:

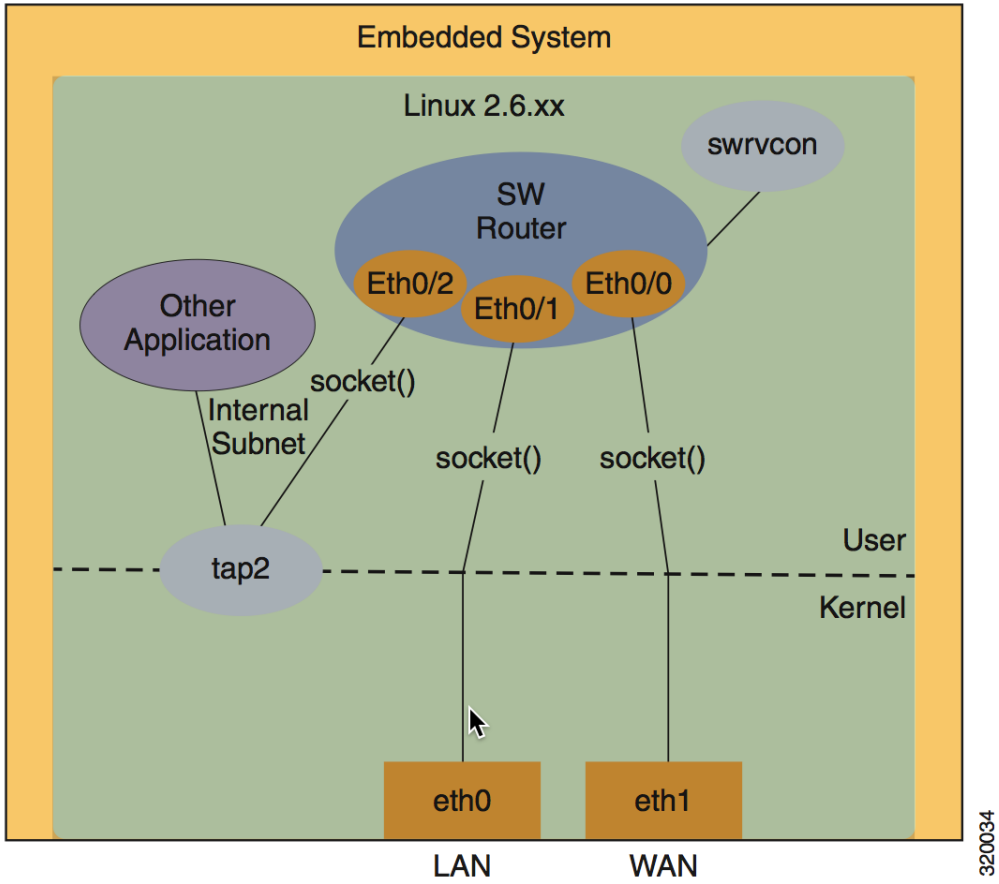
```
DEVICE=tap0
BOOTPROTO=static
IPADDR=10.1.1.2
NETMASK=255.255.255.252
ONBOOT=yes
TYPE=Tap
```

- When finished, restart networking to bring the interfaces online

```
service network restart
```

5921 TAP Interfaces

Figure 1-1 Interactions between Cisco 5921 ESR Components



Prepare continued…

- Disable unneeded services

```
chkconfig iptables off
chkconfig ip6tables off
```

- This will disable the CentOS firewall; do not do this in production
- Enable sshd for management via eth2 IP address

```
chkconfig sshd on
```

- Reboot

```
reboot
```

Install the Cisco 5921 Software

- Copy Cisco 5921 .tar file to a USB flash drive
- Create a directory to mount the USB flash drive to:

```
mkdir –p /media/usbdrive
```

- Insert drive into system and mount it to the above directory:

```
mount /dev/sdb1 /media/usbdrive
```

- Note, the "/dev/sdb1" portion may be different. Use "dmesg | grep sd[a-z]" to determine the correct name
- Create a directory to hold the Cisco 5921 installation:

```
mkdir –p /opt/cisco/c5921
```

Installing continued…

- Extract the contents of the .tar file to the installation directory:

```
tar –xf /media/usbdrive/<name_of_tar_file.tar> –C /opt/cisco/c5921
```

- Unmount the usb flash drive so it can be safely removed:

```
umount /dev/sdb1
```

- Change to the installation directory

```
cd /opt/cisco/c5921/<name of tar file>
```

Installing continued…

- Copy necessary files to installation root directory

```
cp c5921i86-universalk9-ms.S* ../
cp libdyncs.so ../
Cp rvcon ../
cp swr_reload ../
cp c5921-swr-init.sh ../
cp SWROPTIONS.example.txt ../SWROPTIONS
```

- This should copy the critical items to the main "/opt/cisco/c5921" directory.
- Copying the files is recommended so backup copies are available in the "/opt/cisco/c5921/<version_name>" directory

Installing continued…

- Change directory to the 5921 home directory

```
cd ..
```

- Set correct permissions and open SWROPTIONS

```
chmod 744 SWROPTIONS
vi SWROPTIONS
```

- SWROPTIONS file has configurations for eth0, eth1, and eth2 as given in the below example.

```
[interface]
linux=eth1
ios=e0/1
type=raw
promiscuous=true
monitor-state=true
push-mon-int=true
```

- In an earlier step, if you opted to not include the internal TAP connection between Linux and IOS, delete the TAP section of the SWROPTIONS file.

- The SWROPTIONS file is now set-up to map:

  Linux eth0 <-> IOS ethernet0/0
  Linux eth1 <-> IOS ethernet0/1
  Linux tap0 <-> IOS ethernet1/0 (Optional)

Installing continued…

- Link the start up script to "/etc/init.d" and set correct permissions:

```
ln -s /opt/cisco/c5921/c5921-swr-init.sh /etc/init.d/
chmod 744 /etc/init.d/c5921-swr-init.sh
```

- Launch IOS for the first time:

```
/etc/init.d/c5921-swr-init.sh start
```

- IOS should start up. When launched via the init script, IOS will be started in the background, no output will be seen from IOS on the console:

```
[root@centos c5921]# /etc/init.d/c5921-swr-init.sh start
starting ./swr_reload...\n

Loading Image:./c5921i86-universalk9-ms.SPA
Child process will exec swr image now....
./c5921i86-universalk9-ms.SPA running SWR the background, pid=1283, SWR=1284
```

Installing continued…

- When launched via the init script, IOS will not show any output
- Use "swrvcon" to connect to IOS's internal console

```
./swrvcon 100
```

- Press Cntl-C to exit the IOS console and return to the Linux console. To start IOS automatically on bootup, execute this command (optional):

```
chkconfig c5921-swr-init.sh on
```

Installing continued…

- In earlier steps, if you opted to use a TAP interface between Linux and IOS, add the following IOS configuration to complete the “IOS side” of the TAP interface:

```
Router>enable
Router#conf t
Router#interface eth1/0
Router(config-if)#ip address 10.1.1.1 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#end
Router#copy running-config startup-config
```

- Installation is complete
- Next required step is to install appropriate licensing.

Licensing

Cisco 5921 Licensing

- Cisco 5921 is a licensed product
- There are two types of licensing available on the Cisco 5921 router. Classical licensing and Smart Licensing. Default evaluation state bandwidth is 8 kbps.
- Support for the two types of licensing is based on the following parameters:

1) Classical Licensing (refer page 22)

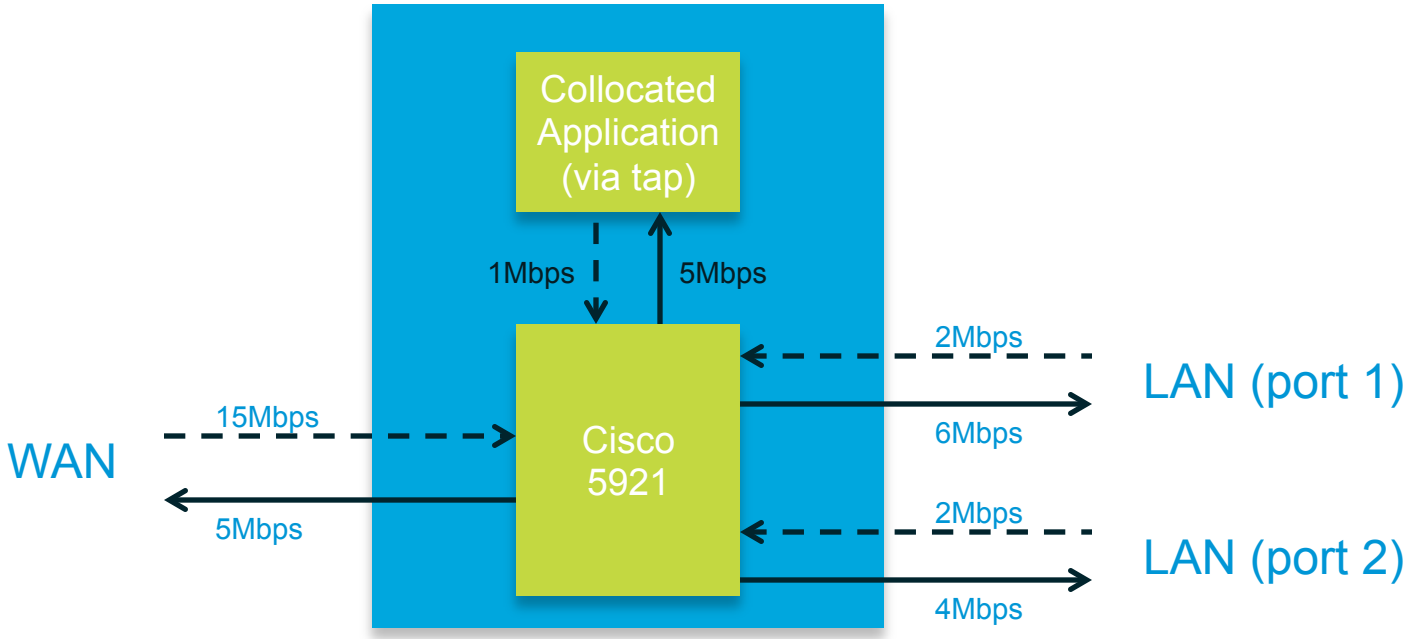
Supported only on C5921 router installed directly on reference boxes.

2) Smart Licensing (refer page 27)

This is supported on C5921 installed either on reference boxes or on virtual machines.

License Bandwidth Limiting

- Depending upon the installed license, the aggregate outbound bandwidth will be capped at 10Mbps, 25Mbps, or 50Mbps (referred to as level 1, level 2, and level 3, respectively).
- For outbound bandwidth calculations, only the layer 2 payload and header are used. Most traffic generators are going to include several other fields, including the CRC, Inter-frame gap, and preamble.
- The following example shows how the bandwidth is calculated.
  - Only the solid lines “count”
  - In this case, bandwidth is calculated at 20Mbps



Classical Licensing
15.2(4)GC

Classical Licensing – Overview

- "show platform software license" will show which, if any, license is installed.

```
show platform software license
```

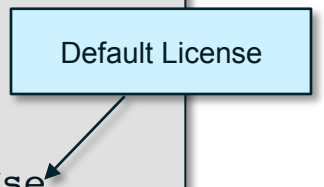
```
Router#show platform software license

Packet forwarding: Enabled

Current enforcement forwarding rate: 8 Kbps

Unique Device Identifier: CISCO5921-K9:9UN4106F2NB

License features supported:
       Feature                 Rate      Status
  ---------------------    ---------   ------
  c5921-x86-default           8 Kbps     In Use
  c5921-x86-evaluation       50 Mbps     -
  c5921-x86-level1           10 Mbps     -
  c5921-x86-level2           25 Mbps     -
  c5921-x86-level3           50 Mbps     -
```



Classical Licensing – obtain license file

- Licenses are installed in the same manner as regular IOS CSL-based licenses
  - Obtain Product Authorization Key (PAK) by purchasing a license
    - Top level product ID is "CISCO5921-K9". Individual license levels are configurable options under the top level PID.
  - Obtain Unique Device Identifier (UDI) from Cisco 5921 installation

```
show license udi
```

UDI is a combination of the product ID and the serial number

```
Router#show license udi
Device#   PID                    SN              UDI
------------------------------------------------------------------------
*0        CISCO5921-K9           9UN4106F2NB     CISCO5921-K9:9UN4106F2NB
```

- PAK and UDI are entered in the Cisco SWIFT system
  - https://tools.cisco.com/SWIFT/LicensingUI/pakReg.Home
  - Once validated, Cisco will email license .lic file

Classical Licensing – installing license

- For enabling other licenses:
  - License files must be installed by copying .lic file to the directory where the "unix:" file system is mapped
    - By default, this is the directory from which IOS is started
    - System integrator is responsible for transferring license to Linux system
- License is installed with standard "license install" command

```
Router#license install unix:9UN4106F2NB_20130809100908761.lic
Installing licenses from "unix:9UN4106F2NB_20130809100908761.lic"

Installing...Feature:c5921-x86-evaluation...Successful:Supported
1/1 licenses were successfully installed
0/1 licenses were existing licenses
0/1 licenses were failed to install

Router#
*Apr  8 07:19:05.873: %LICENSE_C5920-6-LICENSE_ACTIVATED: Installed license for feature
c5921-x86-evaluation now in use. Forwarding bandwidth limited to 50 Mbps
*Apr  8 07:19:06.119: %LICENSE-6-EULA_ACCEPTED: EULA for feature c5921-x86-evaluation 1.0
has been accepted. UDI=CISCO5921-K9:9UN4106F2NB; StoreIndex=0:Primary License Storage
*Apr  8 07:19:06.270: %LICENSE-6-INSTALL: Feature c5921-x86-evaluation 1.0 was installed
in this device. UDI=CISCO5921-K9:9UN4106F2NB; StoreIndex=0:Primary License Storage
```

Classical Licensing – verify license is in use

- After license is installed, the below command shows the status of the license in use.

```
Router#show platform software license

Packet forwarding: Enabled

Current enforcement forwarding rate: 50 Mbps

Unique Device Identifier: CISCO5921-K9:9UN4106F2NB

License features supported:
       Feature                Rate      Status
  ---------------------    ---------   ------
  c5921-x86-default           8 Kbps     -
  c5921-x86-evaluation       50 Mbps   In Use
  c5921-x86-level1           10 Mbps     -
  c5921-x86-level2           25 Mbps     -
  c5921-x86-level3           50 Mbps     -
```

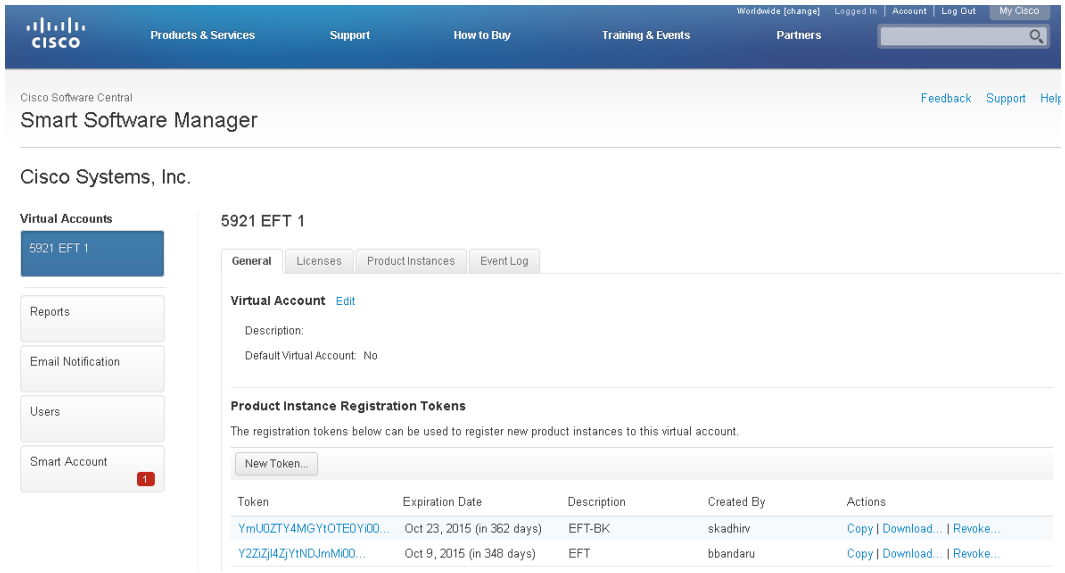
Packet forwarding is enabled @ 50 Mbps.

License is installed and is "In Use"

Smart Licensing
15.5(1)T

Smart Licensing – Overview

- Smart Licensing is software based licensing end-to-end Cisco platforms that consists of tools and processes to authorize the usage and reporting of the Cisco products.
- Two components of Smart Licensing:
  - IOS configuration for installation
  - Smart Software Manager



`Smart Software Manager`

`URL:`http://tools.cisco.com/rhodui/index#/home

Smart Software Manager is a Cisco software Central tool for smart licensing, where the licenses are managed for an account.

Smart Licensing – How to install

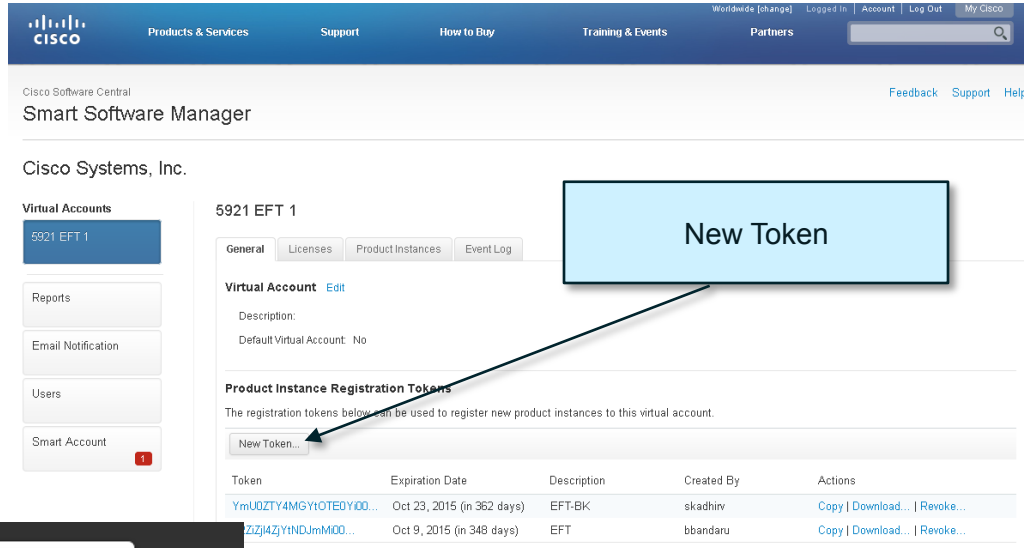
- Smart Licensing can be enabled in configuration mode with the following command.

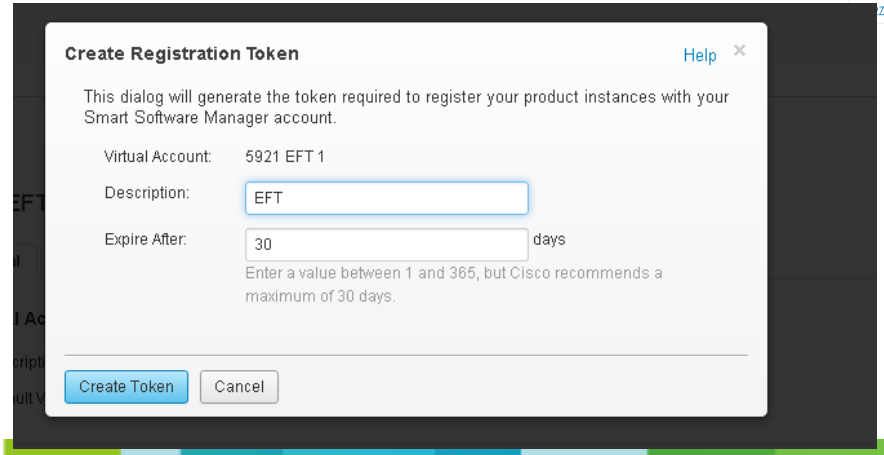
```
Router(config)#license smart enable
Certificate
  Status: Available
  Certificate Serial Number (hex): 01
  Certificate Usage: Signature
  Issuer:
    cn=Cisco Licensing Root CA
    o=Cisco
  Subject:
    Name: Cisco Licensing Root CA
    cn=Cisco Licensing Root CA
    o=Cisco
  …….
```

- Next step is to get the token from Smart Software Manager.

Smart Licensing – Creating Token

- Token can be created by clicking on “New Token” button in Smart Software Manager





Smart Licensing – Product Authorization

- Token can be copied or downloaded to use it for registration from smart software manager.
- Use the token to register the router instances with Smart Software manager.

```
Router#license smart register idtoken
YmU0ZTY4MGYtOTE0Yi00ODRhLTg3Y2YtZmI2MDBjOGEyMWJkLTE0NDU2MDI3%0AMzE3MzZ8T01Ne
DRra1dCWldJVjhLaUt6U2xDOGFvNGNST0N6dFVKdEMVRO%0AZnFFND0%3D%0A
Router#% Generating 2048 bit RSA keys, keys will be exportable...
[OK] (elapsed time was 1 seconds)

*Oct 26 21:59:26.330: %SSH-5-ENABLED: SSH 1.99 has been enabled
Router#
```

Note: Ensure the C5921 is reachable to pubic/Cisco network, so that the token will be registered successfully.

- Next step to verify the product has been successfully authorized by smart software manager.

Smart Licensing – Verify Product Authorization

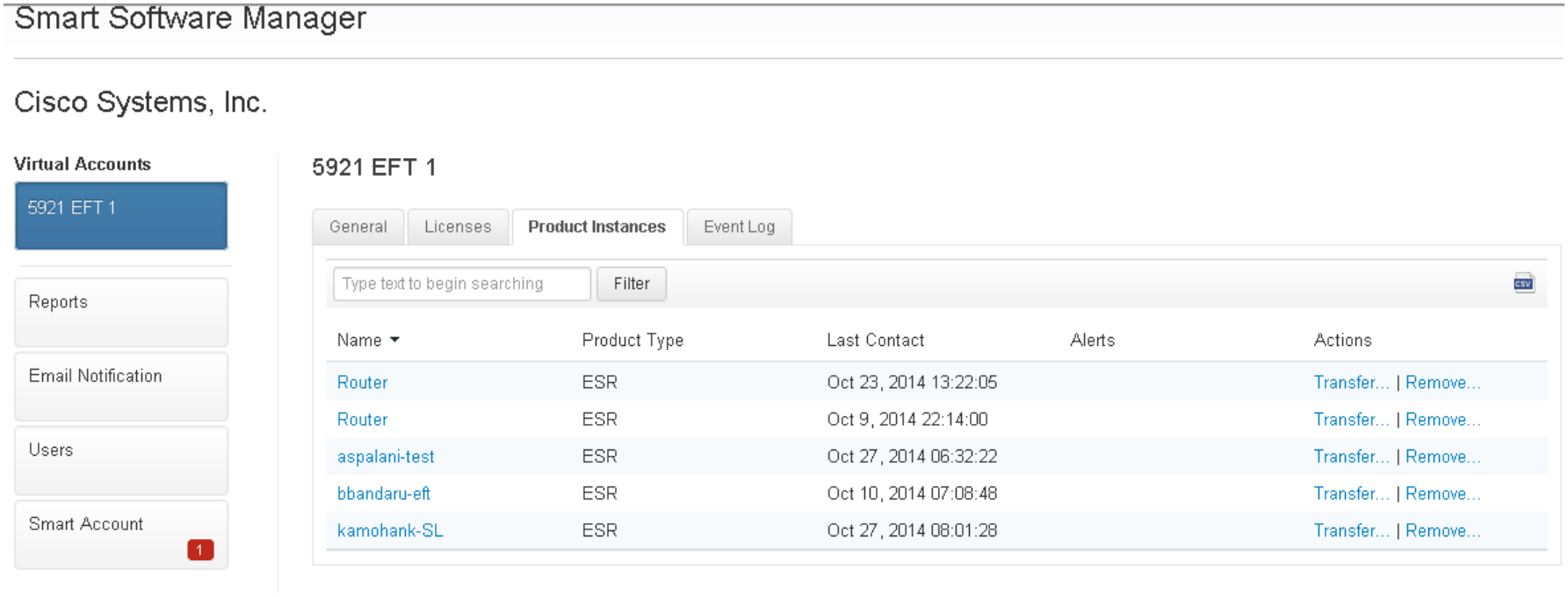
- "Show License tech support" will show product authorization state

```
Router#show license tech support
Cisco Smart Licensing Agent, Version 1.0.0_development
Smart Licensing Enabled: Yes
UDI:
PID:CISCO5921-K9,SN:9D21806DBOR
Compliance Status: In Compliance
Assigned License Pool: Default Virtual Account
Grace period: Not in use
No Entitlements in use
Smart Licensing State: authorized (4)
Licensing Certificates:
ID Cert Info:
Start Date: Oct 6 20:44:04 2014 UTC. Expiry Date: Oct 6 20:44:04 2015 UTC
Serial Number: 70368
Version: 3
Subject/SN: 69f60f2b-28e1-4ffe-96ac-50c658d3b3ac
Common Name: 416DA794E535CBDD70B3E07C66FF0A1E88F465AF::1,2
Signing Cert Info:
Start Date: Jun 14 20:18:52 2013 UTC. Expiry Date: Apr 24 21:55:42 2033 UTC
Serial Number: 3
Version: 3
Upcoming Scheduled Jobs:
Certificate Renewal: Apr 4 20:46:33 2015 UTC (179 days, 23 hours, 59 minutes, 32
seconds remaining)
Certificate Expiration: Oct 6 20:44:10 2015 UTC (364 days, 23 hours, 57 minutes, 9
seconds remaining)
Authorization Renewal: Nov 5 20:46:42 2014 UTC (29 days, 23 hours, 59 minutes, 41
seconds remaining)
Authorization Expiration: Jan 4 20:44:23 2015 UTC (89 days, 23 hours, 57 minutes, 22
seconds remaining)
Daily Job: Oct 7 20:41:07 2014 UTC (23 hours, 54 minutes, 6 seconds remaining)
Component Versions: SA:(1_0_4_throttle)1.0.0, SI:(rel19)1.0.0, CH:(rel4)1.0.7,
PK:(rel15_twig)1.0.0
Other Tech Support Dump:
Grace period time remaining: Not in use
Eval period time remaining: 89 days, 23 hours, 56 minutes, 3 seconds (Not in use)
Stored State Machine State: 4
Transport Mode: Callhome
Router#
```

Authorized

Smart Licensing – Verify Product instances

- Authorized product after token registration can be verified by looking in to the product instances tab in Smart Software Manager.

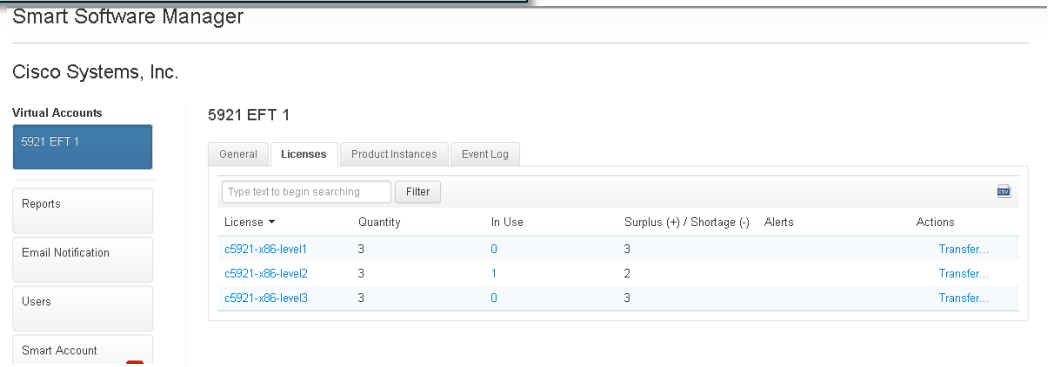


Smart Licensing – Enabling Licenses

- “license platform throughput level” will enable the license in C5921.

```
c5921(config)#license smart enable ← before enable different level licenses
Router(config)#license platform throughput level ?
  c5921-x86-level1  10 Mbps throughput rate
  c5921-x86-level2  25 Mbps throughput rate
  c5921-x86-level3  50 Mbps throughput rate
Router(config)#license platform throughput level c5921-x86-level2
Router(config)#license platform throughput level c5921-x86-level2
Router(config)#
*Oct 26 22:00:09.432: %LICENSE_C5920-6-LICENSE_ACTIVATED: Installed license for feature c5921-x86-level2 now in use. Forwarding bandwidth limited to 25 Mbps
Router#
```

```
Router#show platform software license
Packet forwarding: Enabled
Current enforcement forwarding rate: 25 Mbps
Unique Device Identifier: CISCO5921-K9:9D21806DBOR
License features supported:
Feature Rate Status
---------------------- --------- ------
c5921-x86-default 8 Kbps -
c5921-x86-evaluation 50 Mbps -
c5921-x86-level1 10 Mbps -
c5921-x86-level2 25 Mbps In Use
c5921-x86-level3 50 Mbps -
Router#
```



- Licenses tab in Smart Software manager will show the availed and surplus/shortage licenses.

How to Upgrade IOS Software with License already Installed

License remains valid after the upgrade

Procedures:

1. Stop the 5921 process: ./c5921 –swr-init.sh stop

2. After IOS tar file is untared copy over the following files to replace the older version of those files including:

c5921i86-universalk9-ms.SPA, SWROPTIONS, c5921-swr-init.sh, swr_reload, libdyncs.so

3. Restart 5921 process: ./c5921 –swr-init.sh start

Note: if you a non-CCO image, replace the IOS with SPA for SSA in order to boot it correctly

Other Installation Options…

Enabling serial console - step 1

- Allows access to CentOS console (not IOS) via serial ports
- Create new file for each serial port, where "ttySx" is the serial port being used. For the “reference system,” the proper ports are ttyS3 or ttyS4.

```
vi /etc/init/serial-ttySx.conf
```

- Populate the file with the following:
  - Be sure to change "ttySx" to the actual serial port being used

```
# This service maintains a getty on /dev/ttySx.

start on stopped rc RUNLEVEL=[2345]
stop on starting runlevel [016]

respawn
exec /sbin/agetty /dev/ttySx 9600 vt100-nav
```

Enabling serial console - step 2

- Edit /etc/securetty to allow access via the serial ports:

```
vi /etc/securetty
```

- Add the desired serial port to the end of the file:

```
console
<snip>
tty10
tty11
ttySx
```

Add this line. Be sure to use the real port number in place of "ttySx"

- Reboot to enable the serial console
- Connect with standard console application with 9600/8/N/1
- Please note: You will need a null-modem adapter or cable to use the “reference system’s” console port

References

- ESXi installation: http://virtualization.softwareinsider.com/l/4/VMware-vSphere-ESXi