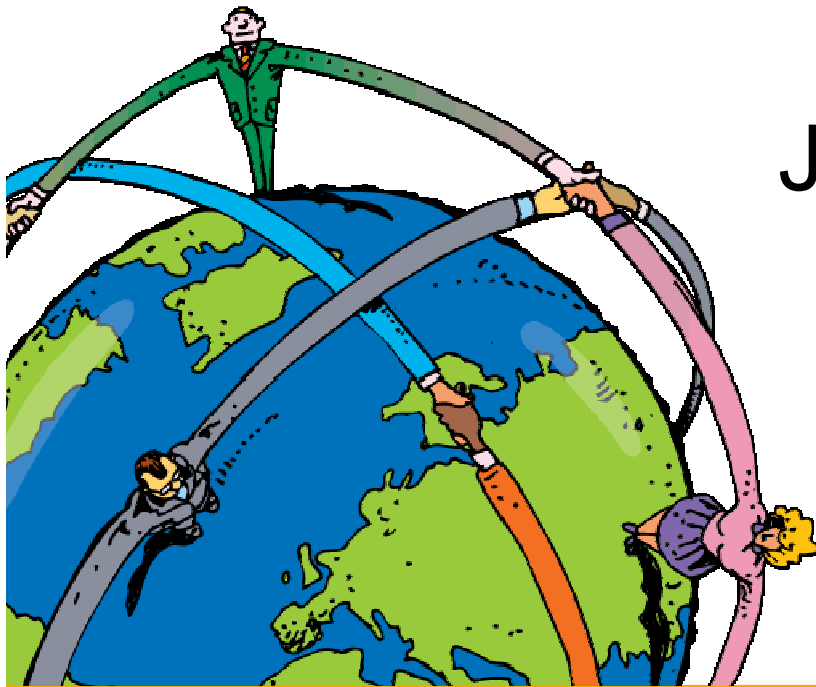


# JUNIPER Networks Routing Workshop



**Amit Kumar Das - SE , India & SAARC**  
**Anand Verma – Senior SE , India & SAARC**

# Agenda

## **Date 11.08.08**

- 17:30 -18:30 Introduction to JUNOS, CLI and basic configuration

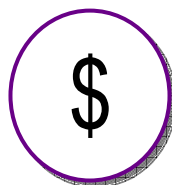
## **Date 12.08.08**

- 9:00 -11:00 OSPF Protocol overview with hands-on  
  
Tea Break
- 11:15-13:30 BGP – Protocol overview with hands-on  
  
Lunch Break
- 14:30-17:00 MPLS-TE-VPN –Technology overview with hands-on
- 17:00-18:00 High Availability

# Leader in High-Performance Networking

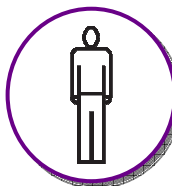
*Well  
Positioned  
for  
Growth*

## Operational Excellence



**\$2.7B**

**Cash and Investments**



**5,000+**

**Dedicated Employees**



**\$120M**

**Quarterly R&D Engine**

***Partner and Shareholder Value***

# Global Presence, Follow-the-Sun Support



 15 Support Centers, 200+ Engineers

 89 Offices in 38 Countries

# Serving Enterprises and Service Providers

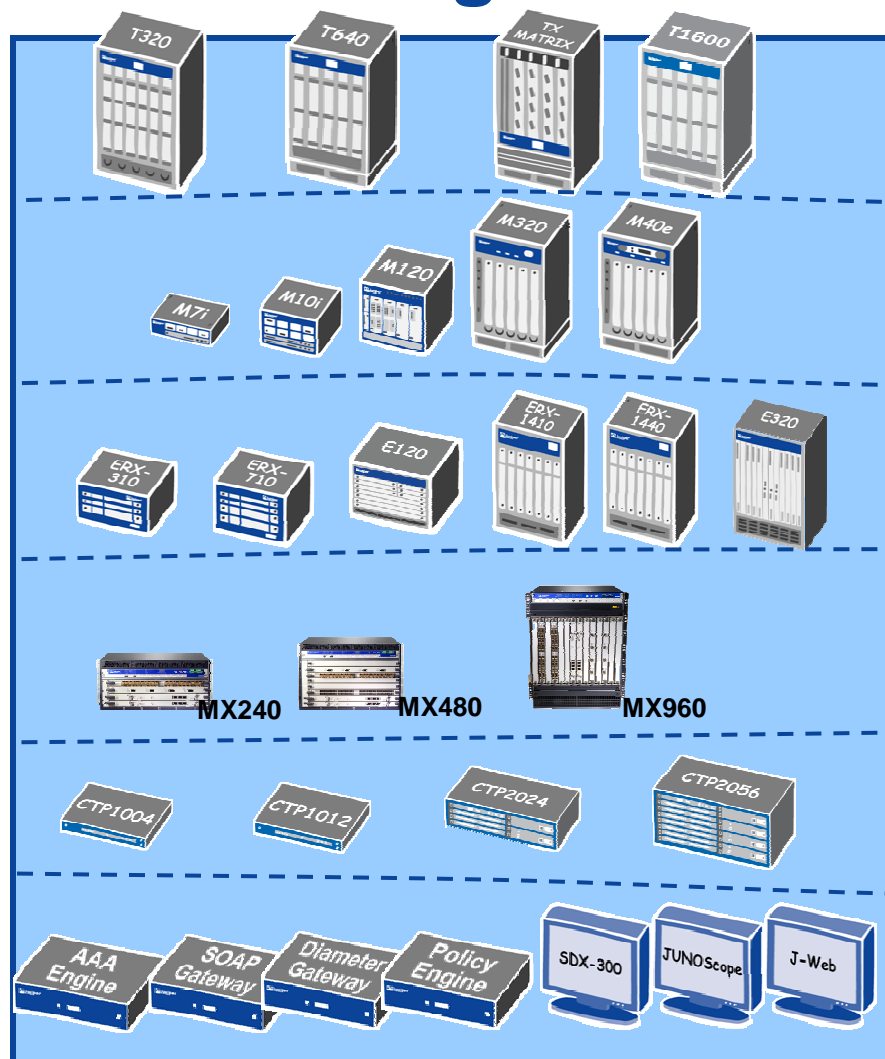
## Service Providers



## Enterprises



# Networking Platform Portfolio



## T-series Core Routing Solutions

Highly available and resilient core routers that scale through multi-terabit capacities

## M-series Multi-Service Edge Solutions

Combines best-in-class IP/MPLS capabilities w/ unmatched reliability and service richness

## E-series Broadband Service Solutions

Carrier class routing platforms for IP-Edge and Broadband Service applications, such as multiplay

## MX Carrier Ethernet Solutions

High-Density Carrier Ethernet edge services in a highly scalable and reliable platform

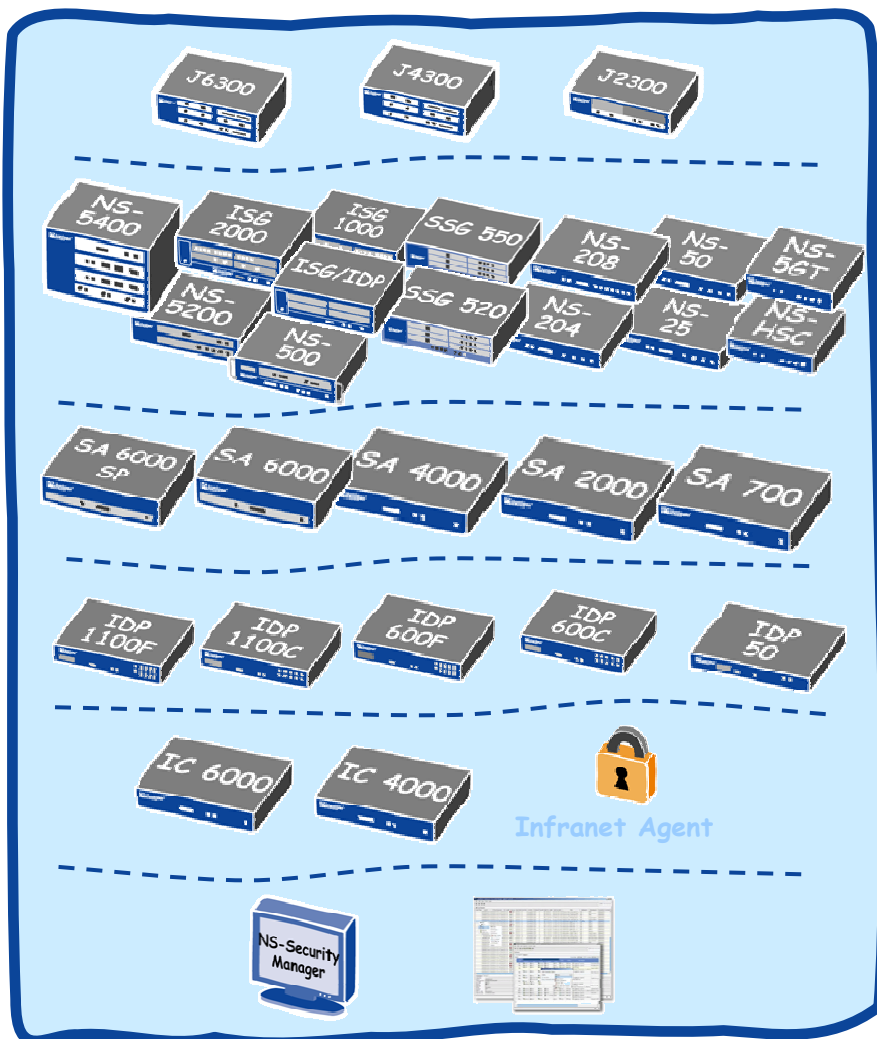
## Circuit to Packet Solutions

Provides technology and features to reliably transport circuit based apps across IP networks

## Management and Policy Solutions

Provides admin and control of solution portfolio, enabling the rapid service creation and mgt

# Security Product Solutions



## Enterprise Routing Solutions

Service provider quality routers for the enterprise designed for remote, branch or regional offices

## Integrated Firewall/IPSec VPN Solutions

Purpose-built security appliances with WAN & LAN interface flexibility and performance capabilities to protect enterprise and service provider networks

## SSL VPN Solutions

Product lines for secure LAN, extranet and intranet access to mobile employees, customers and partners with no client software deployment

## Intrusion Detection and Prevention Solutions

Intrusion prevention appliances that help protect networks and critical resources from attacks

## Unified Access Control Solutions

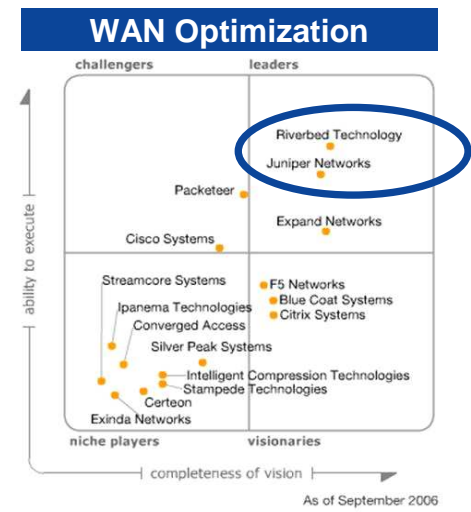
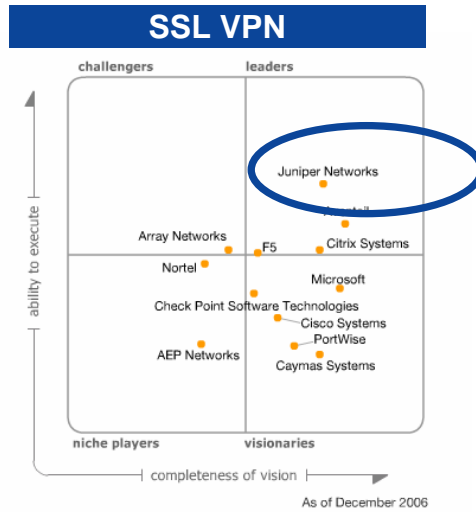
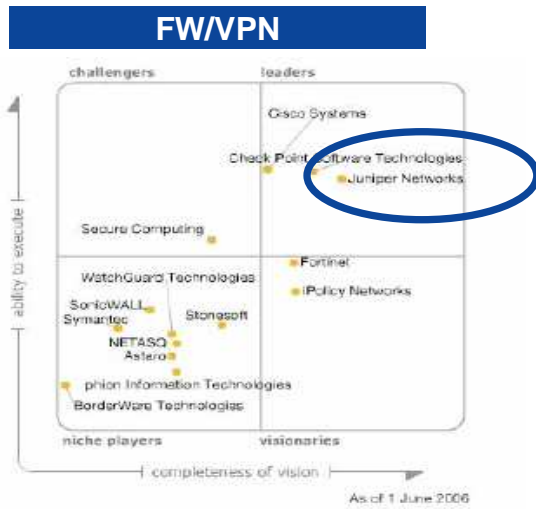
Product line consisting of three components that work together to enable a cost-effective, pragmatic solution solving endpoint security as it effects the LAN

## Central Policy-based Management Solution

3-tier system providing role-based administration and central control and logging of all FW/VPN solutions

# Gartner Magic Quadrants

## Juniper, a proven leader in all categories



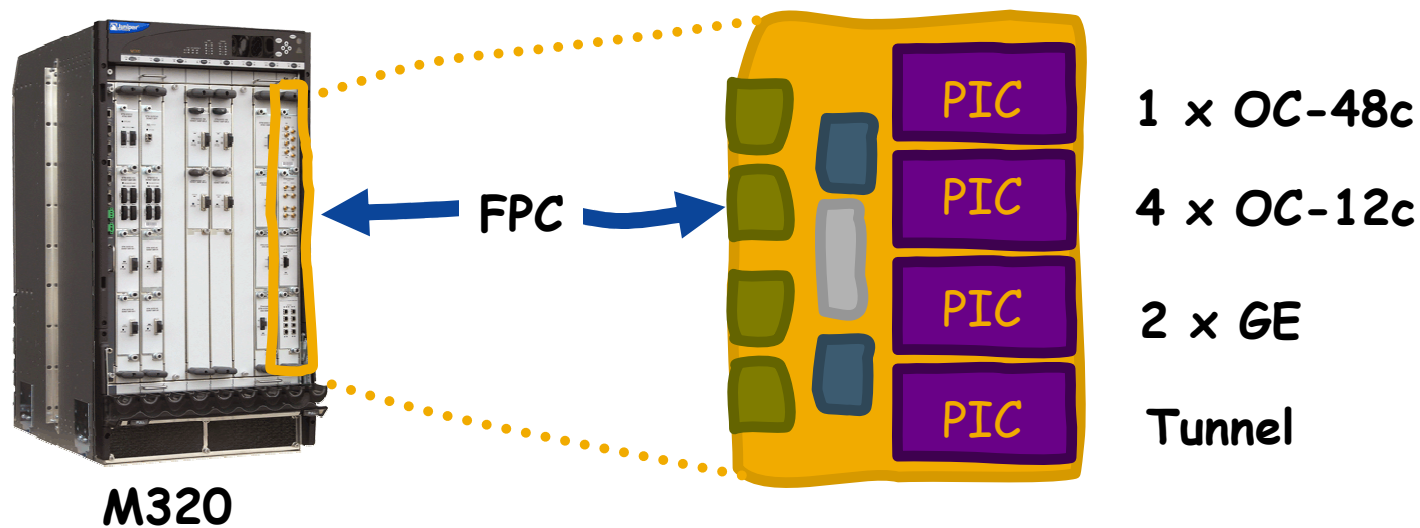




# JUNOS and Platforms Training

## Modular interface architecture

- Physical Interface cards known as “PIC”s
- PICs plug into Flexible PIC Concentrators (“FPC”s)
- Advantages:
  - Mix and match interface media per FPC slot
  - PIC portability (i.e. migration from one platform to another)
  - PIC hot insert/removal



## PFE Overview

- **FPCs**
  - Hardware platform which accepts PICs
- **PICs**
  - Physical Interface Card
  - Contains physical layer components
- **Control board with internet Processor ASIC**
  - CFEB M7i/M10i
  - SFM M40e
  - FEB M120
  - SIB M320/T320/T640/TX Matrix

## Routing Engine Overview

- **JUNOS software resides in flash memory**
  - Alternate copy available on hard drive
- **Provides routing protocol intelligence to PFE**
  - Not directly involved with packet forwarding
- **Implements command-line user interface**
  - Operations
  - Administration
  - Maintenance
  - Provisioning
- **Manages care and feeding of PFE**

## Interface Names

- **Physical interfaces have standard names**

- Type
- FPC slot
- PIC slot
- Port number



**so-5/2/3**

## Interface Names

- Logical interfaces are used to set up Frame Relay DLCIs or ATM virtual circuits

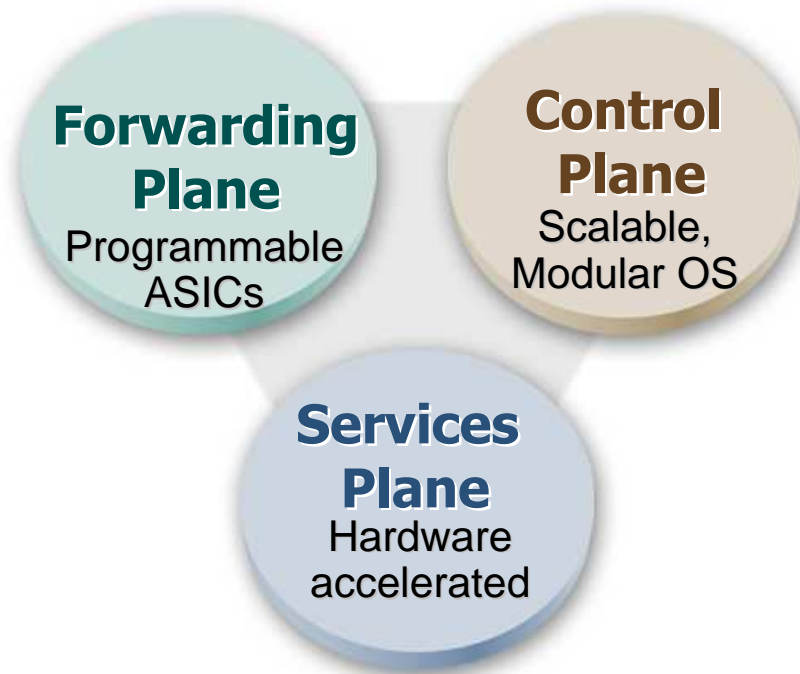
**so-5/2/3.43**



- Interface number is separate in meaning from the actual DLCI or ATM VC and can be any arbitrary value
- Suggested convention is to keep them the same whenever possible
- Router has two permanent interfaces
  - Out-of-band management interface is called `fxp0`
  - Internal Routing Engine to PFE connection is called `fxp1`

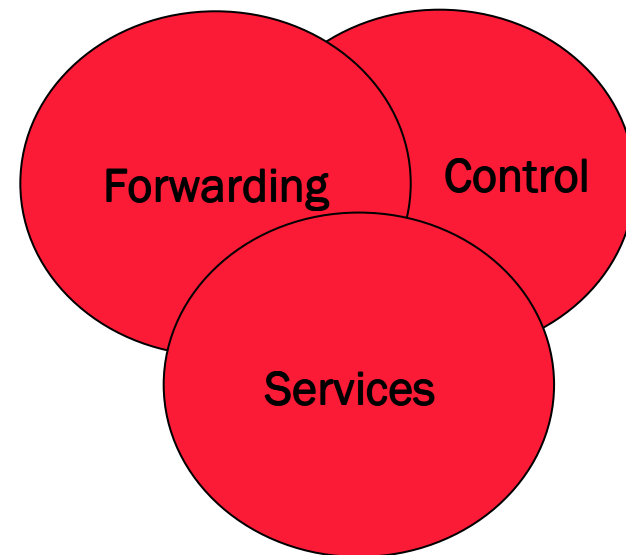
# Service Built Architecture

## Juniper platforms



- Linear, High performance with features
- Secure – DOS attack resistant
- Modular OS - Inherently stable

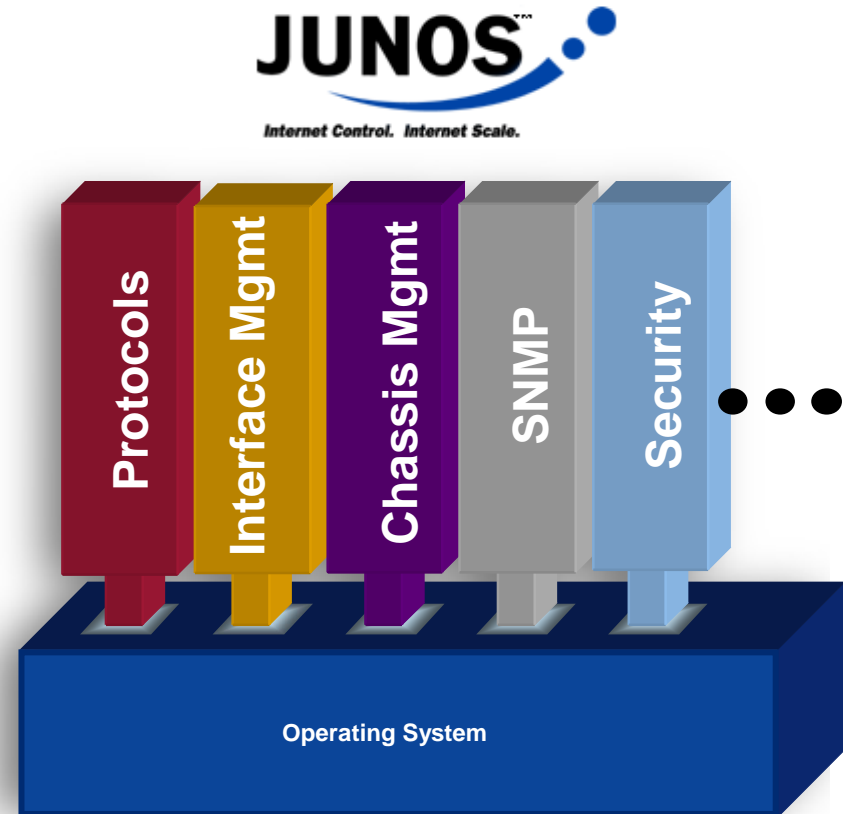
## Software / CPU based platforms



- Performance suffers with features
- DOS attack vulnerable
- Monolithic OS – prone to instability

# JUNOS Software

- Deployed since early 1998
- Purpose-built for reliability and scalability
- Modular design
  - Failure protection
  - Independent restart
  - Individual daemons
  - Memory protection

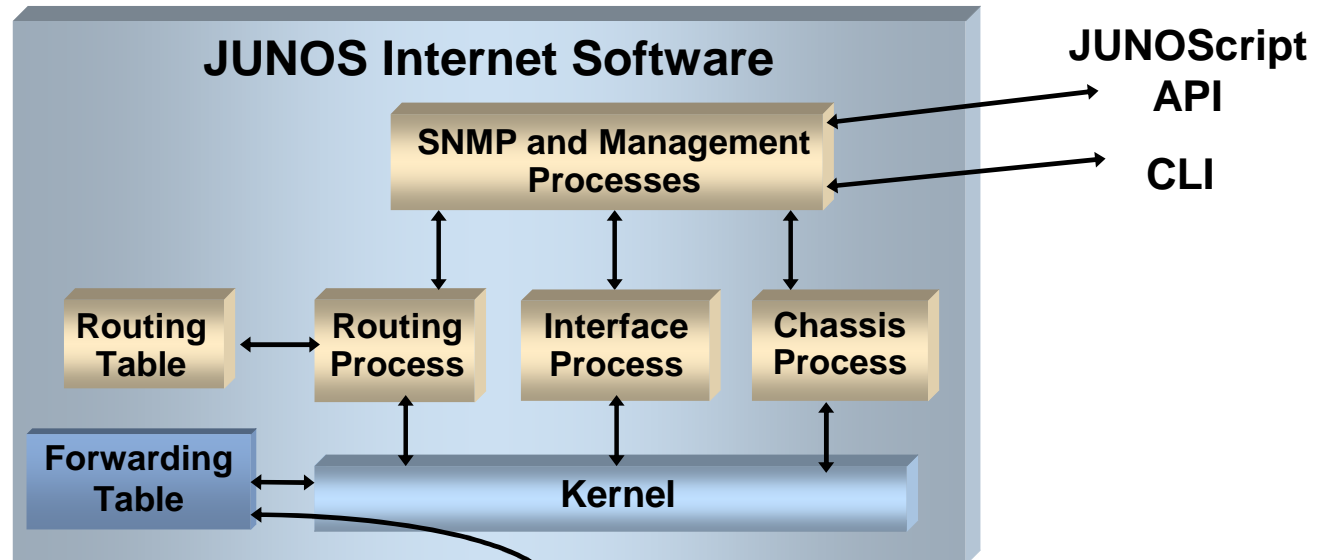




# JUNOS Architecture

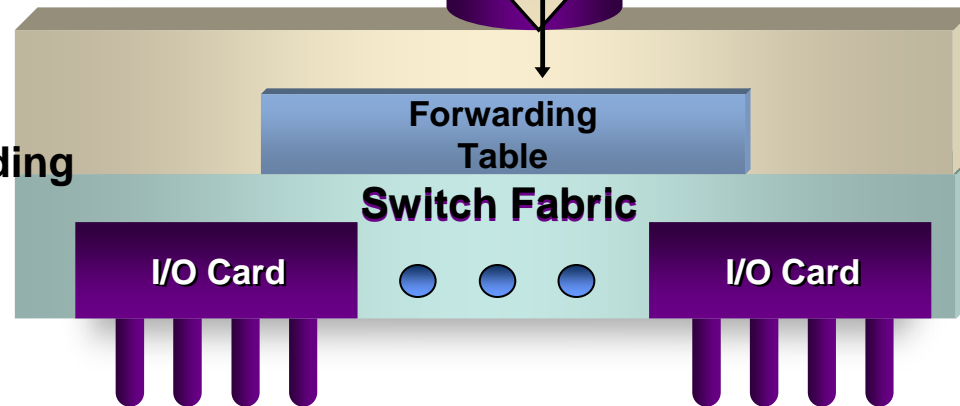
## Control

### Routing Engine



## Forwarding

### Packet Forwarding Engine

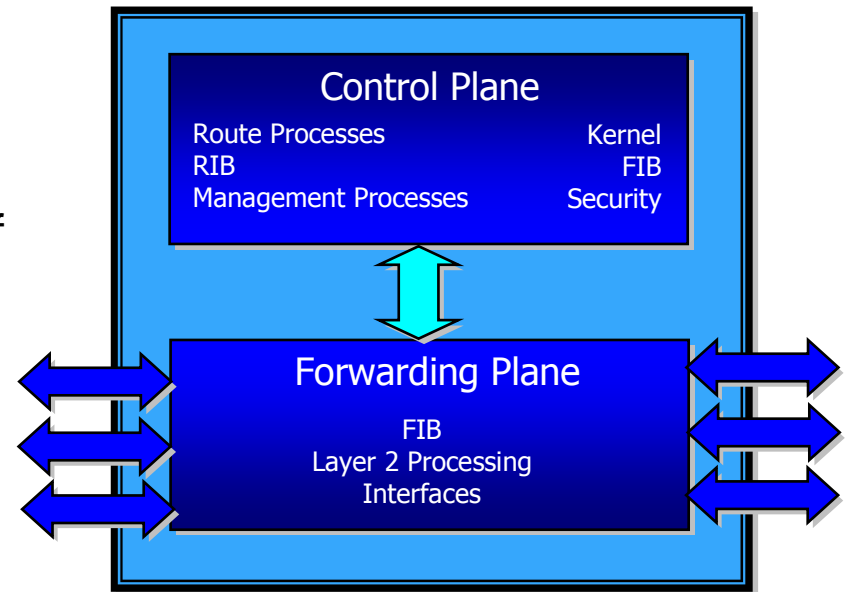


## What Makes an OS Carrier Class?

- **STABILITY**
- **SCALABILITY**
- **SECURITY**
- **PRECISION**
- **HIGH AVAILABILITY**
- **FEATURE RICHNESS**
- **CONSISTENCY**
- **PREDICTABILITY**
- **PROCESS**

# Stability

- **Separation of control and forwarding planes**
  - Protects control plane during periods of heavy traffic
  - Protects forwarding plane during times of network instability
- **Separate control plane and forwarding plane processors**
  - Separation of functions is no good if they are controlled by the same processor
- **Modular processes running in protected memory**
  - Failure of one process does not affect other processes



- **Disciplined control and regression testing of code is fundamental to network survival**
  - System failures are highly disruptive
  - System failures are slow to recover
  - System failures tend to cause cascading failures through the network

## Scalability

- **Scalability in component expansion**
  - Wide diversity of interface types in same chassis
  - Memory flexibility to handle network growth
- **Scalability in features**
  - Wide diversity of available features
  - All currently supported features in one image
  - No “feature specific” packages
- **Scalability in protocol capabilities**
  - Wide diversity of supported IP protocols

# Security

## ■ Self protection

- Powerful firewall capability between forwarding and control planes
  - Allows strict specification of what packets are allowed to reach control plane
- Rate limiting between forwarding and control planes
  - Allows rate limitation of packets that must be accepted but that might be exploited for DoS attacks

## ■ Network protection

- Protocol security
  - Strong authentication between communicating peers
- Attack prevention/alleviation tools
  - Allows traceback to attack entry points
  - Allows throttling of attack traffic

# Precision

- **Incorrect route calculations can lead to:**
  - Routing loops
  - Black holes
- **Carrier-class route calculations must be right *every time***
- **If you don't have precision, you don't have:**
  - Stability
  - Scalability
  - Security

## High Availability: Minimizing Downtime

- **Minimizing downtime from software failures:**
  - Intelligently modular software architecture
  - Graceful switchover to backup control plane
  - Graceful switchover to backup forwarding plane
- **Minimizing downtime from configuration (human) errors:**
  - No “automatic” commits during configuration
  - Strong error checking features
  - Customizable configuration rules (commit scripts)
- **Minimizing planned downtime:**
  - Uninterrupted switchover of routing protocols
  - In-service software upgrades

# Modularity

- **Modularity is an engineering tool**
  - For creating reliable software
  - By dividing the job into manageable chunks
- **Sizing the modules:**
  - Large enough to reasonably contain interdependencies
  - Small enough so engineering teams can reasonably understand and manage the code
- **Allows same teams to manage same modules release after release**
  - No separate “Release A” and “Release B” teams
- **Modularity benefits the customer in terms of reliability and performance**



## Why Is This Interesting ?

“The possibility of failures would be much reduced if you consider that changing device configuration causes **60% of downtime due to human error.**”

--Jeffrey Nudler  
Senior Analyst \*

\* <http://www.networkworld.com/news/2005/101005-ietf.html>

## Prevent Configuration Error

- **JUNOS: offline configuration model**
  - Candidate ← CLI changes
  - Check-out ← 'commit check'
  - Committed ← active config
- **Versioning, in-line warnings, commit-confirmed, rollback, activate/deactivate**



# JUNOS

## Command Line Interface

## 10,000–Foot View

- **Move around statement hierarchy using `edit` command**
  - Like UNIX `cd` command
- **Alter configuration using `set` command**
- **Activate configuration using `commit` command**

# Enter Configuration Mode

- **Type `configure` at the CLI prompt**

```
root@lab2> edit
Entering configuration mode
[edit]
root@lab2#
```

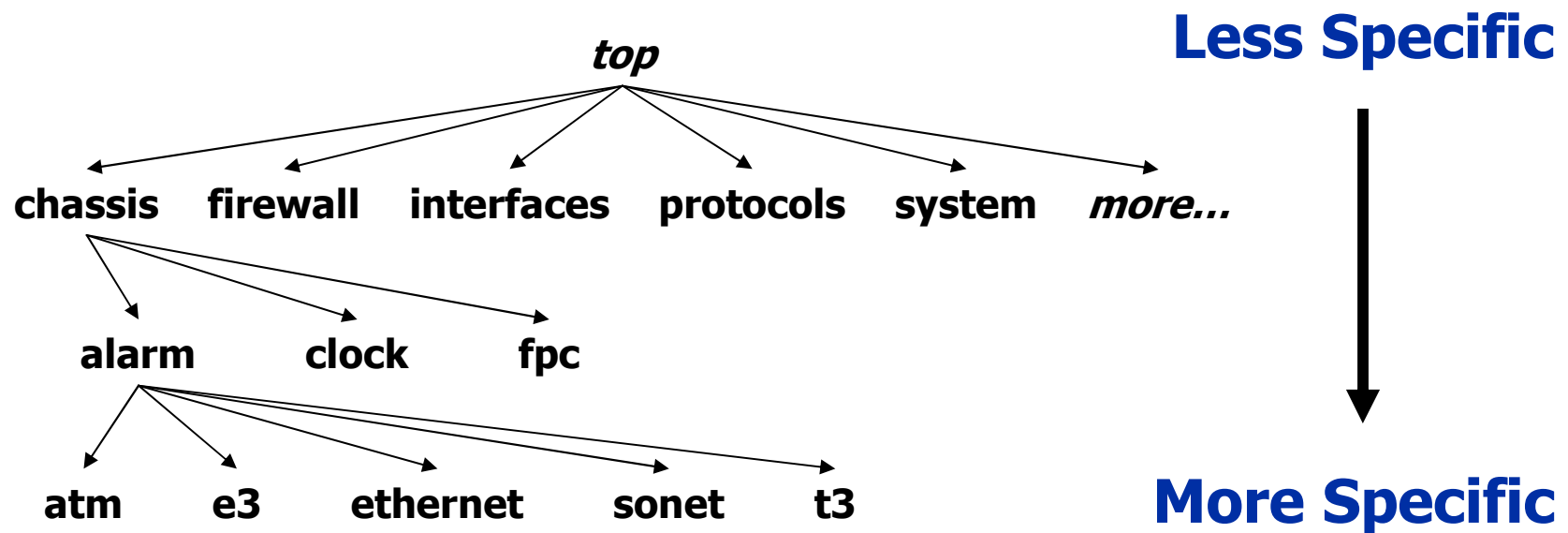
- **Other users in configuration mode**

```
root@lab2> edit
Entering configuration mode
Current configuration users:
  diana terminal d0 on since 1999-10-14 07:11:29 UTC,
    idle 00:00:49 [edit protocols ospf]
The configuration has been changed but not committed

[edit]
root@lab2#
```

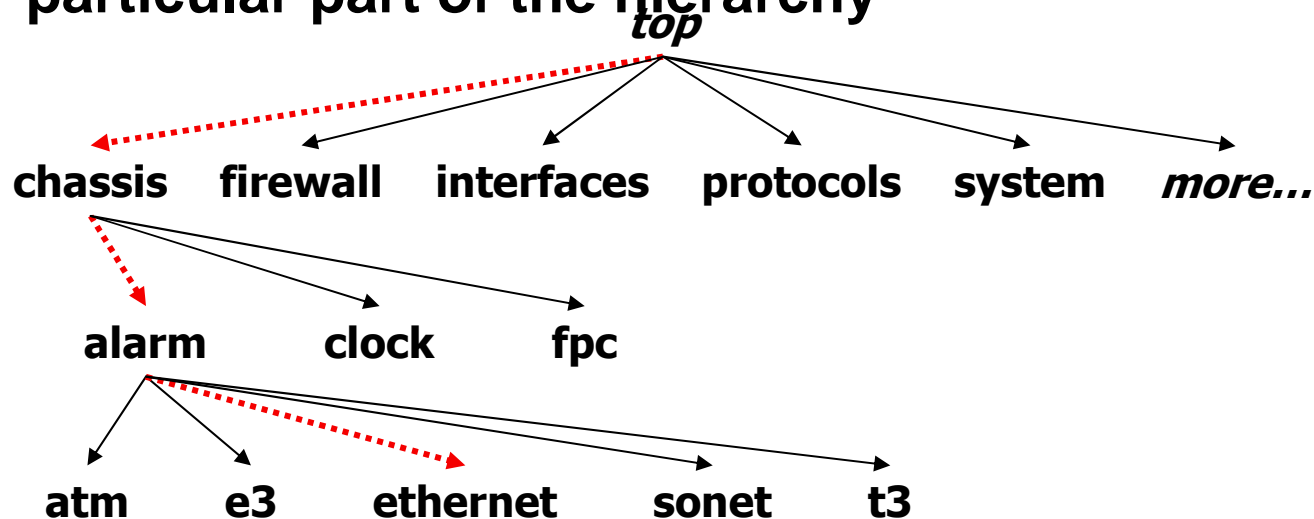
## Move around the Hierarchy

- Configuration statements organized as a tree
  - Similar to UNIX/Windows–style directories



## Move around the Hierarchy

- Use the `edit` command to focus your attention on a particular part of the hierarchy



```

user@host# edit chassis alarm ethernet
[edit chassis alarm ethernet]
  
```

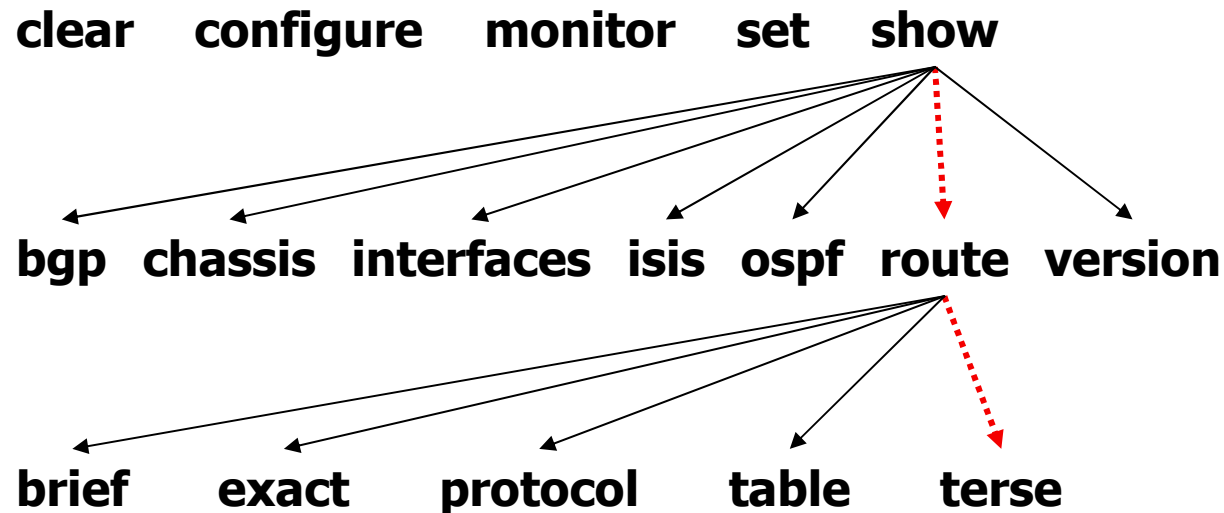
## Move around the Hierarchy

- **Use the `exit` command to move back to where you just were**
  - `exit` at the top level exits configuration mode and puts you back into operational mode
  - `exit configuration-mode` exits no matter where you are
- **Use the `up` command to move up a level**
- **Use the `top` command to move to the top of the hierarchy**



# Overview

- **Command hierarchy**



**Less Specific**



**More Specific**

## Log In

- **Special treatment for “root” login**
  - Can only log in as root from console port
  - Must create additional user with superuser privileges to log in via network ports
  - Be sure to review security implications

## Edit Lines

- **Move the cursor**

Ctrl-B

Back one character

Ctrl-F

Forward one character

Ctrl-A

To beginning of line

Ctrl-E

To end of line

- **Delete characters**

Delete or

backspace key

Delete character before cursor

Ctrl-D

Delete character under cursor

Ctrl-K

Delete from cursor to end of line

Ctrl-U

Delete all characters

Ctrl-W

Delete entire word to left of cursor

- **Other keys**

Ctrl-L

Redraw the current line

Ctrl-P

Move backwards through command  
history

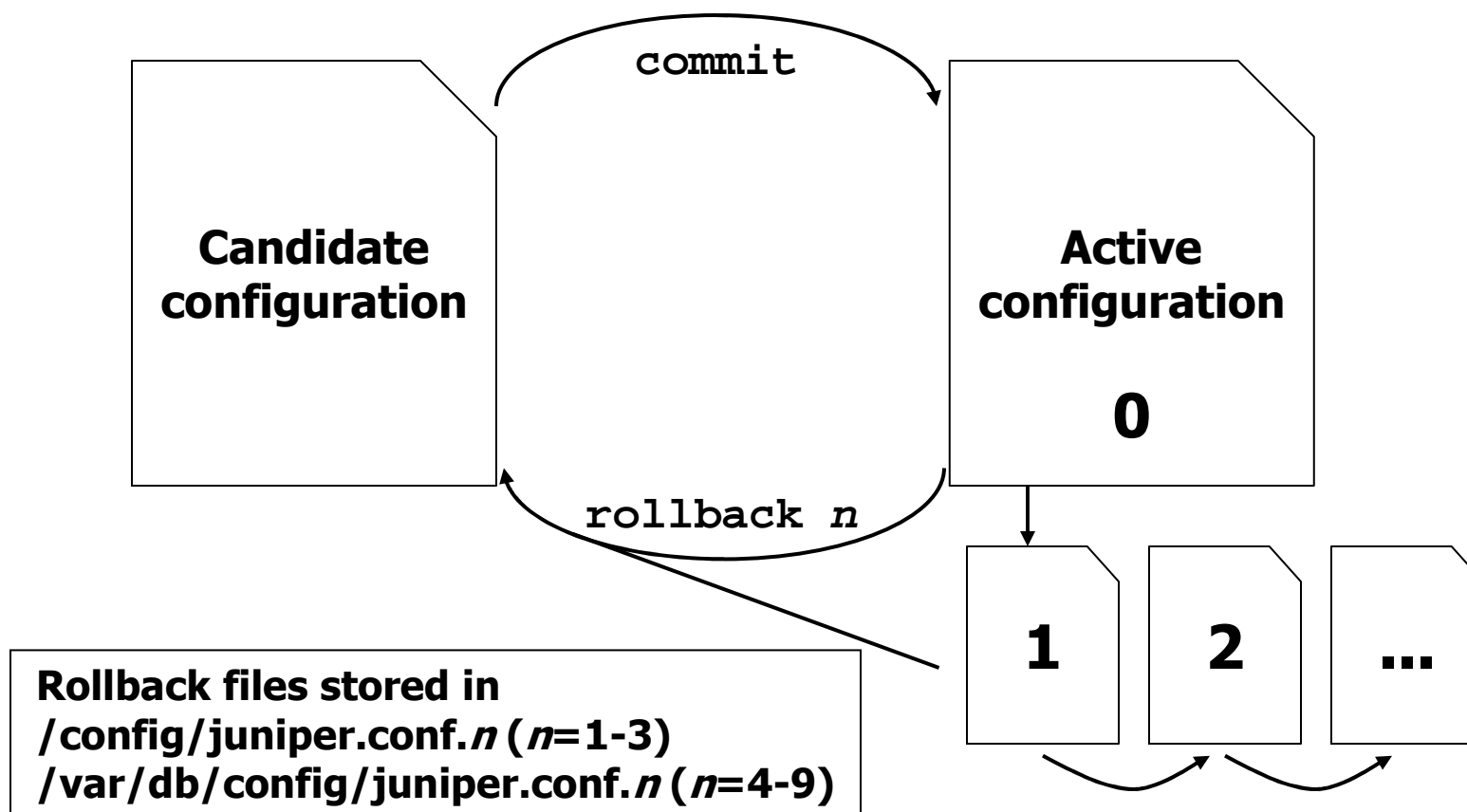
Ctrl-N

Move forward through command  
history

## Configure the Router: Overview

- CLI has separate configuration mode
- You edit a copy of current configuration called the **candidate** configuration
- Changes you make are visible to other CLI users
  - Changes they make might conflict with your changes
- Changes do not take effect until you commit them
- When committed, candidate configuration becomes active and a new candidate is created

# Configure the Router: Overview



## Activate a Configuration

- **Activate configuration changes using the `commit` command**

```
[edit]  
user@host# commit  
commit complete  
[edit]  
user@host#
```

- **Checks configuration before activating it**
- **System never commits for you**

## Activate a Configuration

- **Inband configuration has disadvantages**
  - Might disrupt connectivity to router
  - Might disrupt inband session
- **Avoid disadvantages using `commit confirmed` command**
  - Activates configuration for a few minutes (default is 10 minutes)
  - If configuration is not confirmed, router returns to previous configuration automatically
  - Confirm configuration by issuing a second `commit` command

## Back out Changes

- Use the `rollback` command to restore one of the last nine previously committed configurations
- `rollback` or `rollback 0` resets the candidate configuration to the currently running configuration, which is the last version committed.
- `rollback 1` loads the configuration before that
- and so on



## Commit Dual RE

- **When commit is entered, the system will only activate the changes in the local RE.**
- **Use `commit sync` command to make changes to activate in both RE.**

## Save Configuration Files

- Current candidate configuration **from current hierarchy level and below** can be saved to ASCII file using **save** command

```
[edit]
cli# save filename
[edit]
cli#
```

- File is saved to user's home directory unless full path name is specified
- Filename can be URL or in user@host notation

## Load a Configuration File

- **Syntax**
  - `load (replace | merge | override) filename`
- **Changes candidate configuration only**
- **You must `commit` to activate**
- **Can take input from the terminal**
- **Use the `load` command to**
  - ***Override*** an existing configuration
  - ***Merge*** new statements into existing configuration
  - ***Replace*** existing statements in current configuration

# Command Summary

- **Add and modify configuration statements**
  - `edit`, `set`, `rename`, and `insert` commands
- **Display current configuration**
  - `show` command
- **Save, validate, and activate a complete configuration**
  - `commit` command
- **Return to previously saved configuration**
  - `rollback` command
- **Remove configuration statements**
  - `delete` command
- **Display other users configuring router**
  - `status` command

## View Log Files

- **Additional logging can be turned on on a per-module basis**
  - use `traceoptions flag keywords`
  - specify file name with `traceoptions file file-name command`

- **Example**

```
[edit protocols ospf]
```

```
cli# set traceoptions flag errors
```

```
cli# set traceoptions file ospf-log
```

## View Log Files

- **System keeps log files in /var/log**
  - messages file contains running commentary about system operation
  - Can be tuned to provide minimal to extensive logging
- **View with**  
`show log file-name`
- **View in real time with**  
`monitor start file-name`

# Powerup and Powerdown

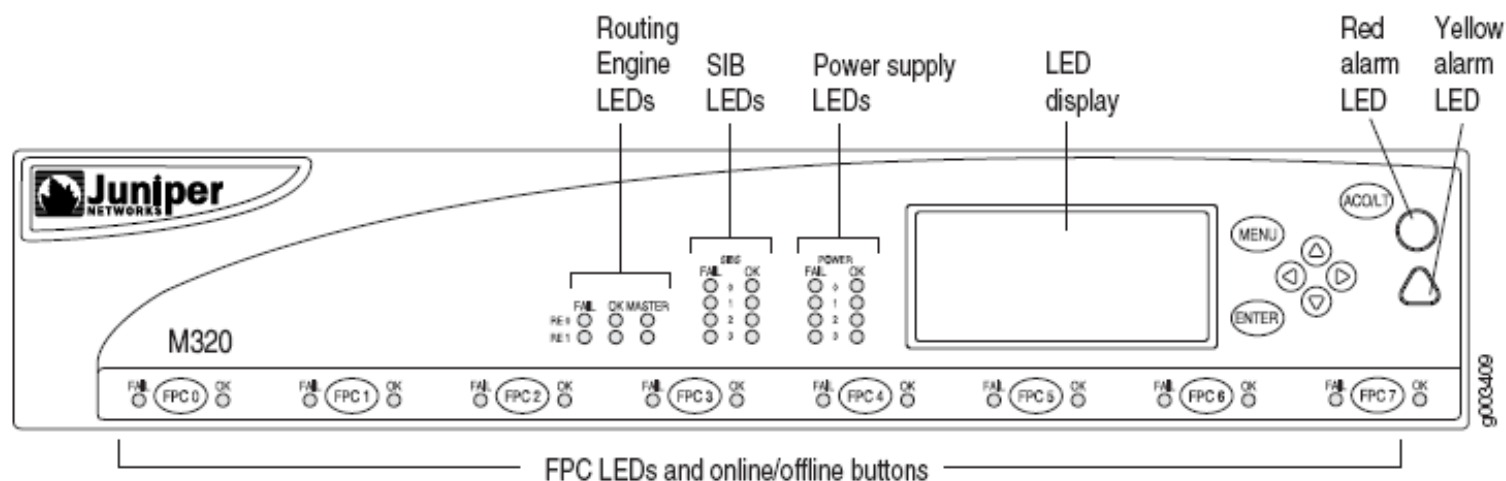
## ■ Powerup

- Connect all cables
- Turn on one power supply
- Turn on second power supply

## ■ Powerdown

- Shut down JUNOS routing software
- CLI request `system halt` command
- Turn off power supplies

# Visible Activity at Startup

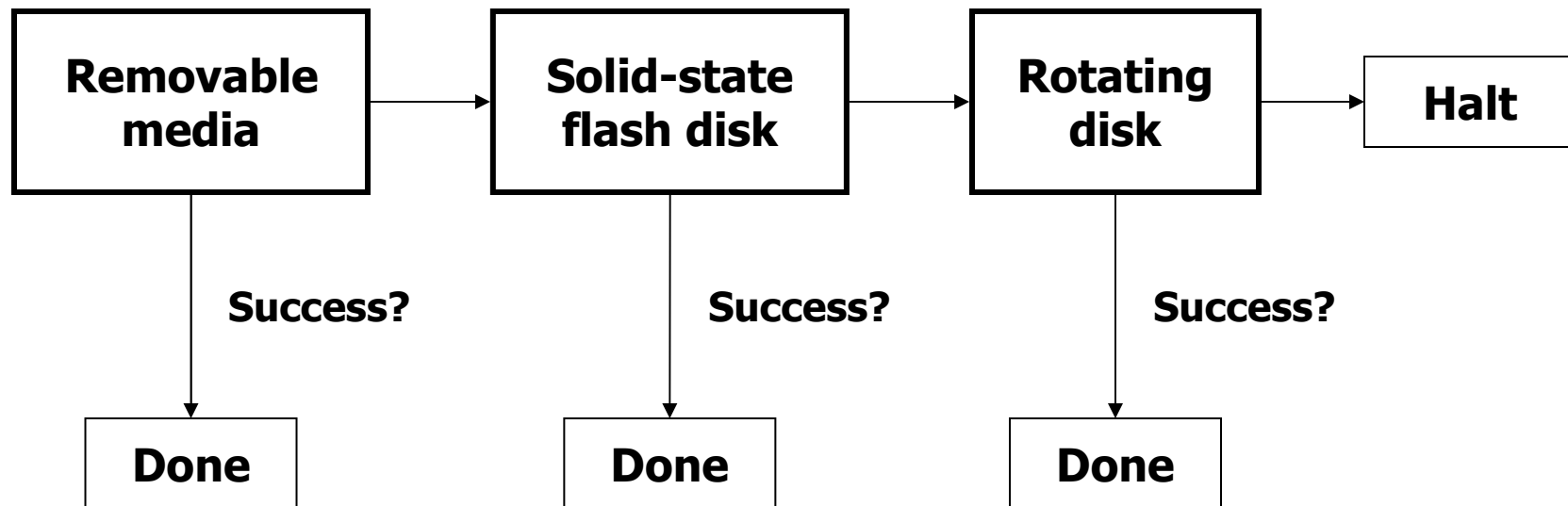


- **Craft interface displays:**
  - Starting Routing Engine
  - Starting PFE
  - Starting Cards
- **FPC LEDs**
  - Blink green while testing
  - Become solid green when tests pass
- **Alarm LEDs light as needed**



## Boot Sequence

- **Hardware controlled**
  - Software notifies hardware when boot completes



## Initial Configuration

- **Root password**
  - Root password **not** set at factory
  - Must use console to configure root password
- **Router and domain name**
- **Management interface IP address and prefix length**
- **Default router IP address**
- **DNS server IP address**

# Initial Configuration

## ■ Enter configuration mode

```
root@> configure
[edit]
root@#
```

## ■ Set root password

### ● Plain text known

```
root@# set system root-authentication
      plain-text-password
```

### ● Pre-encrypted password

```
root@# set system root-authentication
      encrypted-password encrypted-password
```

### ● SSH (secure shell) key

```
root@# set system root-authentication
      ssh-rsa key
```

# Initial Configuration

- **Set router name**

```
[edit]  
root@# set system host-name lab2
```

- **Set router domain name**

```
[edit]  
root@# set system domain-name juniper.net
```

- **Commit changes so far**

```
[edit]  
root@# commit  
commit complete
```

```
[edit]  
root@lab2#
```

## Initial Configuration

- **Set management Ethernet IP address and prefix**

[edit]

```
root@lab2# set interfaces fxp0 unit 0 family inet address ip-  
address/prefix-length
```

- **Set default route**

[edit]

```
root@lab2# set system backup-router gateway-address  
root@lab2# set routing-options static route default nexthop  
gateway-address retain no-readvertise
```

- **Set name server address**

[edit]

```
root@lab2# set system name-server ns-address
```

## Full Installation

- **Reinstall JUNOS software if storage media fails or is corrupted**
- **Future major software revisions may require full installation**
- **Three steps**
  - Prepare to reinstall JUNOS software
  - Reinstall JUNOS software
  - Configure JUNOS software

# Full Installation: Software Configuration

- Log in as root

```
no-name (ttyd0)
```

```
login: root
```

```
Last login: date on ttyd0
```

```
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
```

```
The Regents of the University of California. All rights reserved.
```

```
---JUNOS 4.0R1 built 2000-02-10 09:29:44 UTC
```

```
#
```

- Start CLI

```
# cli
```

```
root@no-name>
```

## Software Update Packages

- **JUNOS software updates are contained in four packages**
  - jkernel—Operating system
  - jroute—Routing Engine software
  - jpfe—Packet Forwarding Engine software
  - jdocs—On-line documentation
  - jbundle—All four upgrade packages
- **Packages can be upgraded individually**
- **CLI `show system software` command displays installed packages**



# Package Naming Convention

- **Software packages have standard names**

*package-m.nZnumber.tgz*

- *m.n* is the major version number
- **Z** is a single uppercase letter
  - A–Alpha
  - B–Beta
  - R–Release
  - I–Internal
- *number* is the release number, which might include the build number for that release

- **For example**

*jbundle-4.0R1.2.tgz*

# Upgrade Software Packages

- Download current package from software download page at [www.juniper.net](http://www.juniper.net)

- Add new package

```
root@lab2> request system software add new-package-name  
Checking available free disk space...11200k available,  
6076k suggested.
```

- If JTAC requests, reboot router

```
root@lab2> request system reboot
```

## Back Up Existing Software

- **System software and configuration can be backed up to rotating disk**
- **Best used**
  - Before major upgrade to ensure system recovery if necessary
  - When system is known stable
- **CLI request `system snapshot` command**

## Interface Port Type

- **at**— ATM over SONET/SDH ports
- **e3**— E-3 ports
- **fe**— Fast Ethernet ports
- **so**— SONET/SDH ports
- **t3**— DS-3 ports
- **ds**- Nx64k interfaces
- **ge**— Gigabit Ethernet ports
- **ml**- multilink
- **ls** – link services
- **sp** – adaptive services pic
- **vt**- virtual interface

# Configure Interfaces

- Physical properties
  - Clocking
  - Scrambling
  - Frame check sequence (FCS)
  - Maximum transmission unit (MTU)
  - Keepalives
  - Other link characteristics
- Logical properties
  - Protocol family (Internet, ISO, MPLS)
  - Addresses (IP address, ISO NET address)
  - Virtual circuits (VCI/VPI, DLCI)
  - Other characteristics

## Configure Interfaces

- **Standard configuration statement hierarchy**

```
interfaces {  
    interface-name {  
        physical-properties;  
        [...]  
        unit unit-number {  
            logical-properties;  
            [...]  
        }  
    }  
}
```

# Configure Physical Properties

- **Configure physical properties of the interface using the set command:**
- **Or park yourself in the interfaces section of the hierarchy and set many options**

```
set interface so-1/0/3 no-keepalives
```

```
lab@omaha> configure
[edit]
lab@omaha# edit interfaces so-1/0/3
[edit interfaces so-1/0/3]
lab@omaha# set no-keepalives
lab@omaha# set sonet-options fcs 32
lab@omaha# commit
```

# Logical Interface Settings

## ■ Logical settings

- Protocol family (Internet, ISO, MPLS)
  - Protocol MTU
  - IP address
  - Other protocol options
- Virtual circuit identifiers (VPI.VCI, DLCI)
- Other according to-circuit characteristics



## Configure Logical Interfaces

- Use the `set` command to configure a logical interface, using the unit number

- For example

```
set interface so-1/0/3 unit 40 dlci 40
```

- Or park yourself at the `unit` level

```
lab@omaha> configure
```

```
[edit]
```

```
lab@omaha# edit interfaces so-1/0/3 unit 40
```

```
[edit interfaces so-1/0/3 unit 40]
```

```
lab@omaha# set dlci 40
```

```
lab@omaha# set family inet address 10.0.20.1/24
```

```
lab@omaha# commit
```

## Configure Protocol Families

- Each major protocol is called a family
- Internet protocol has TCP, UDP, and ICMP as family members
- Most common protocol families are
  - Internet (`inet`)
  - International Standards Organization (`iso`)
  - Traffic engineering (`mpls`)
  - Multiple families can live on one logical interface

## Configure Protocol Families

- Internet protocol family (`inet`)
- Allows you to set
  - IP address: `address A.B.C.D/prefix_length`
  - Remote address on point-to-point links: `destination A.B.C.D`
  - Broadcast address: `broadcast A.B.C.D`
  - MTU size: `mtu bytes`
  - ICMP redirect control: `no-redirects`

# Configure Protocol Families

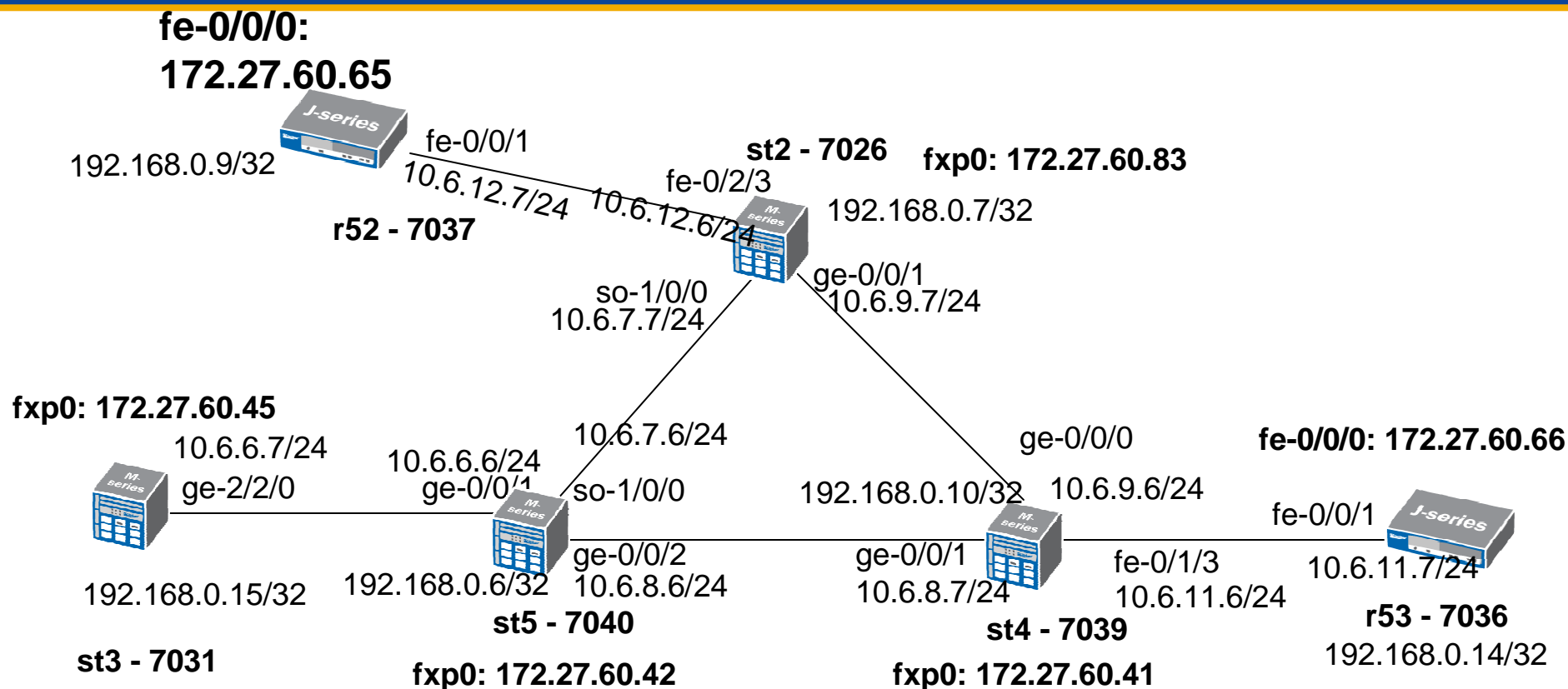
- **Minimal sample configuration**

```
lab@omaha> configure
[edit]
lab@omaha# edit interfaces so-1/0/3
[edit interfaces so-1/0/3]
lab@omaha# set unit 0 family inet address 10.0.20.1/24
lab@omaha# commit
```

- **Displayed as**

```
interfaces {
  so-1/0/3 {
    unit 0 {
      family inet {
        address 10.0.20.1/24;
      }
    }
  }
}
```

# Hands-On Session



All user/pw pair is lab/sanogLab.

You can access r52 from Internet by telnet to 218.189.73.130.

To access via fxps, please use the IP 172.27.60.xx after logging in to the r52 platform.

**\*\*\*\* Please do not delete apply-groups and group configurations \*\*\*\***

**Juniper**  
**your**  
**Net**<sup>™</sup>

