Complex Networks Maintenance and Troubleshooting

CCNP TSHOOT: Module 1, 2, 3

Agenda

- **Planning Maintenance for Complex Networks**
- **Troubleshooting Processes for Complex Enterprise Networks**
- **Using Maintenance and Troubleshooting Tools and Applications**

Planning Maintenance

Network Engineer/Admin’s Job

1) Device installation and maintenance
   - Installing devices, creating, backing up configuration
2) Failure response
   - Device or link failure, replacing equipment, restoring backups, supporting users
3) Network performance
   - Capacity planning, performance tuning, usage monitoring
4) Business procedures
   - Documenting, compliance auditing, SLA management
5) Security
   - Implementing security procedures, penetration testing

Structured vs. Interrupt-driven Maintenance

- **Interrupt driven**
  - Usually in smaller networks because overhead of structured network is large
  - Reaction to a problem, not prevention
- **Structured driven**
  - Proactive approach with predefines processes
  - Response to incident is more efficient
- *You cannot avoid interrupt-driven work entirely!*
  - Failures will happen, you cannot plan them
  - Structured driven approach reduce the amount of interrupt-driven work

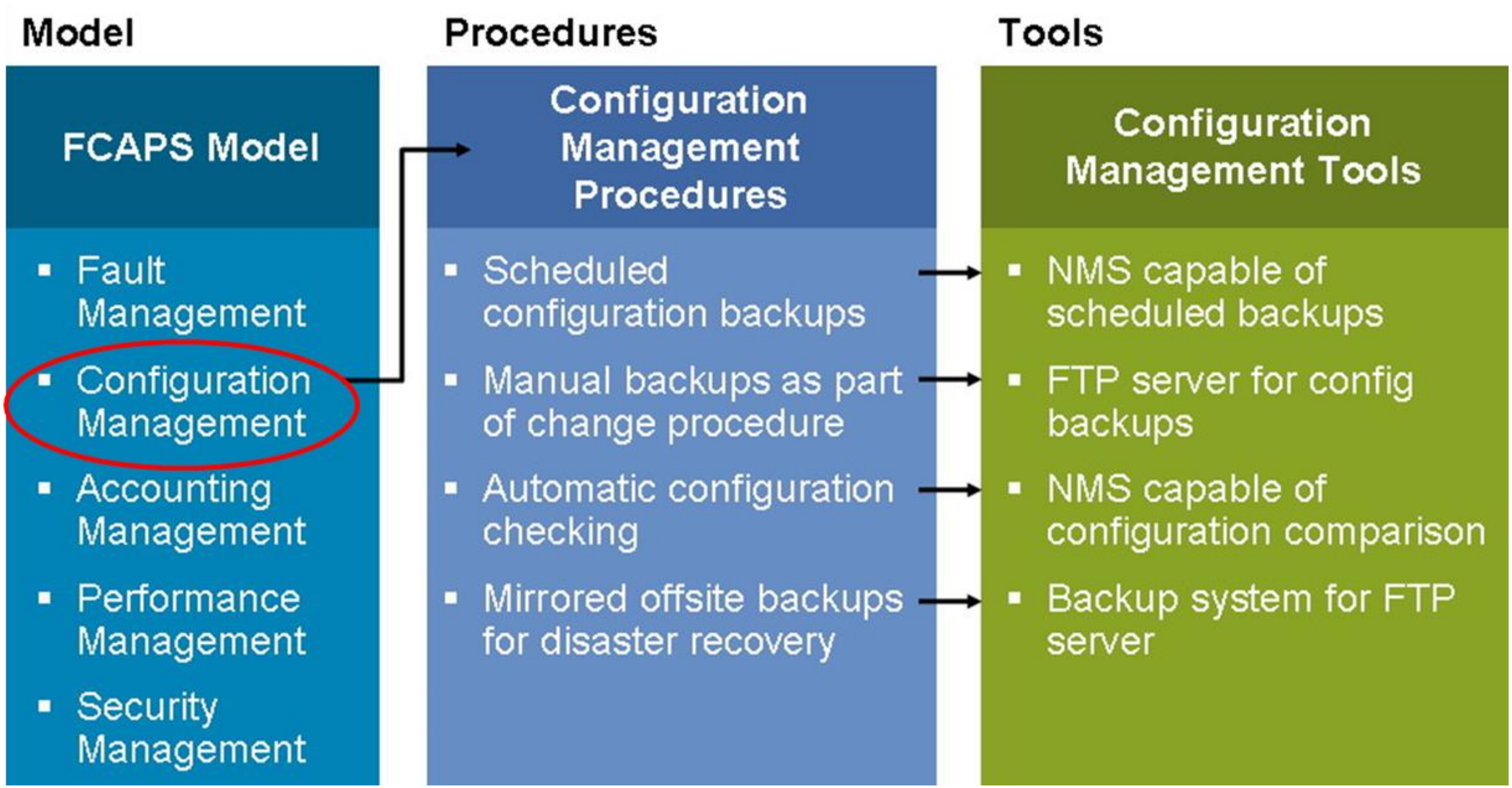
Structured Maintenance Advantages

- **Proactive instead of reactive**
  - Discover and prevent problems before they happen.
- **Reduced network downtime**
  - Maximize mean time between failures (MTBF)
  - Minimize mean time to repair (MTTR)
- **More cost effective**
  - Performance monitoring and capacity planning for budgeting
- **Better alignment with business objectives**
  - Time and resources are allocated to processes based on importance to the business
  - E.g., Upgrades and major maintenance jobs are not scheduled during critical business hours
- **Improved network security**
  - Up-to-date prevention and detection mechanisms

Maintenance Models

- **IT Infrastructure Library (ITIL)**
  - Framework of best practices for IT Service Management
- **ISO – FCAPS**
  - **F**ault management
  - **C**onfiguration management
  - **A**ccounting management
  - **P**erformance Management
  - **S**ecurity Management
  - http://www.ciscopress.com/bookstore/product.asp?isbn=1578701805.
- **ITU-T – Telecommunications Management Network**
  - M.3000 for Bussiness, Service, Network and Element management
- **Cisco Lifecycle Services Phases – PPDIOO**
  - Prepare, Plan, Design, Implement, Operate, and Optimize

FCAPS Model



Model

FCAPS Model

- Fault Management
- Configuration Management
- Accounting Management
- Performance Management
- Security Management

Procedures

Configuration Management Procedures

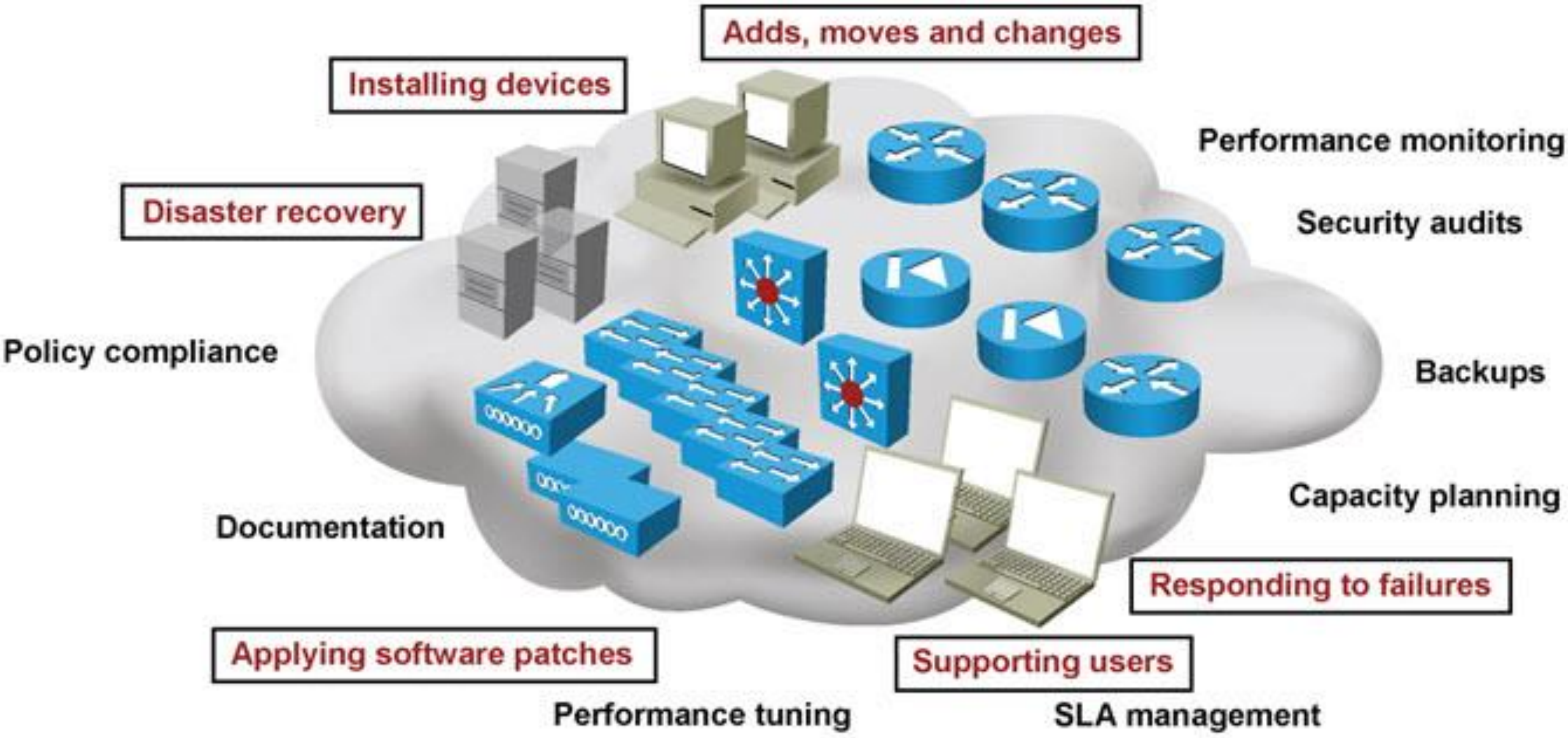
- Scheduled configuration backups
- Manual backups as part of change procedure
- Automatic configuration checking
- Mirrored offsite backups for disaster recovery

Tools

Configuration Management Tools

- NMS capable of scheduled backups
- FTP server for config backups
- NMS capable of configuration comparison
- Backup system for FTP server

Network Maintenance Processes



Network Maintenance Processes

- **Accommodating Adds, Moves, and Changes**
  - Affects users, computers, printers, servers and phones and potential changes in configuration and cabling.
- **Installation and configuration of new devices**
  - Includes adding ports, link capacity and network devices.
- **Replacement of failed devices**
  - Done through service contracts or by in-house support engineers.
- **Backup of device configurations and software**
  - Good backups of both software and configurations can simplify and reduce downtime
- **Troubleshooting link and device failures**
  - Diagnosing and resolving failures related to network components
- **Software upgrading or patching**
  - Requires that you stay informed of available software upgrades or patches and use them if necessary. These can address critical performance or security vulnerabilities.
- **Network monitoring**
  - Using mechanisms such as router, firewall logs or by using sophisticated network monitoring applications
- **Performance measurement and capacity planning**
  - Facilitates planning for upgrades (capacity planning) to help prevent bottlenecks, congestion and failures.
- **Writing and updating documentation**
  - Current network documentation is used for reference during implementation, administration, and troubleshooting is a mandatory network maintenance task.

Network Maintenance Planning

- **Scheduling maintenance**
  - Reduces network downtime. Prevent long-term maintenance tasks from being forgotten. Disruptive maintenance tasks are scheduled during assigned maintenance windows.
- **Formalizing change control procedures**
  - Which changes require authorization and who is responsible? What kind of preparation is needed? What verification is required? Does documentation need to be updated?
- **Establishing network documentation procedures**
  - Includes network drawings, connection documentation, equipment lists, IP address administration, configurations and design documentation.
- **Establishing effective communication**
  - Who is making changes and when? Are affected parties aware of the changes and results? What conclusions can be drawn?
- **Defining templates/procedures/conventions**
  - Examples include: Logging and debug timestamps settings (local time or UTC), access list guidelines (end with explicit "deny any“), IP subnet and address assignment (address allocated to the local gateway).
- **Planning for disaster recovery**
  - Includes replacement hardware, current software and configuration information, tools, licenses (if applicable) and knowledge of the procedures required.

Documentation

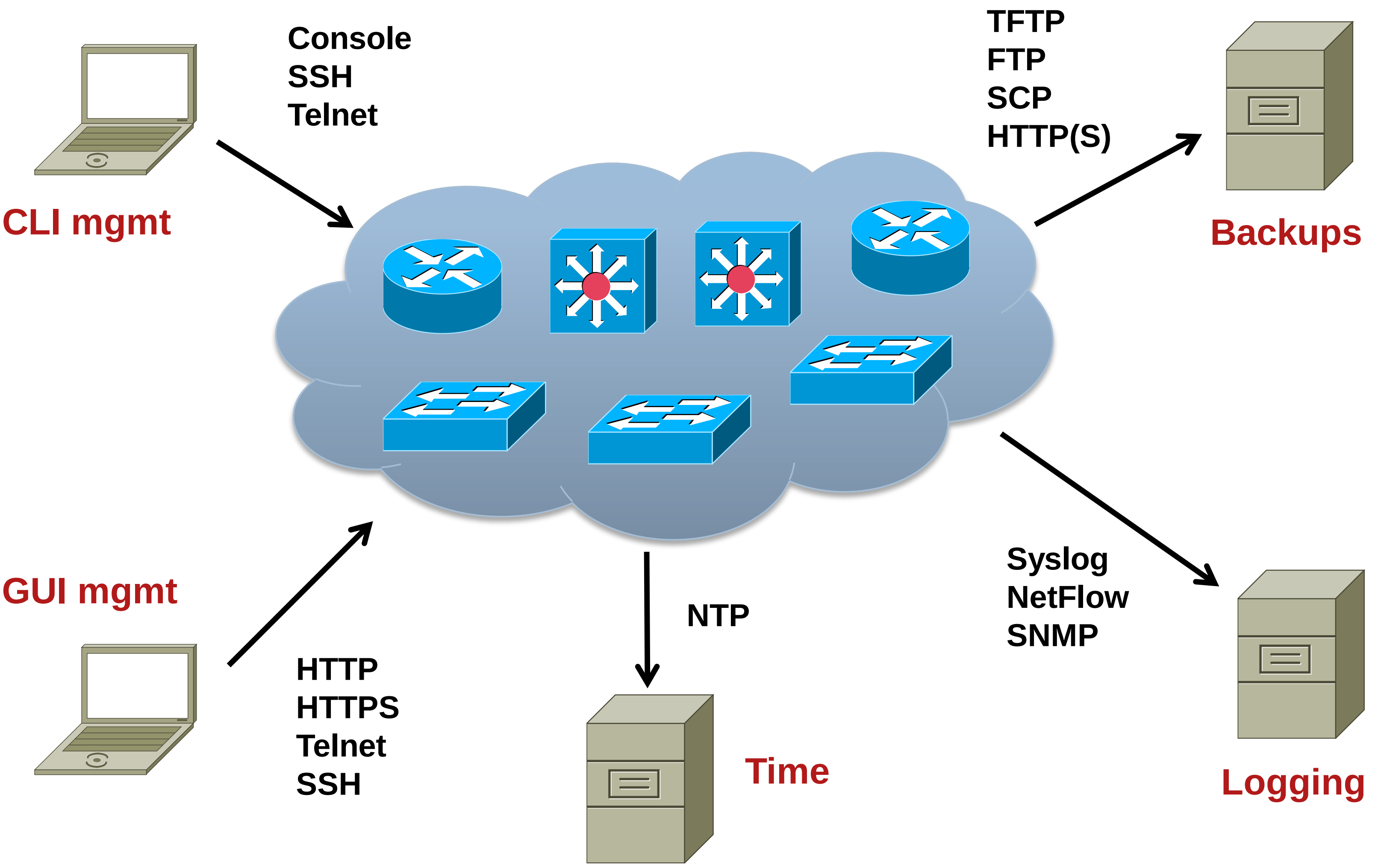
- Accurate documentation is useful for effective troubleshooting
- *Outdated documentation is worse than no documentation!*
  - Documenting the problem and changes during troubleshooting is usually the last things on your mind
- Network diagrams help quickly isolate part of the network
- IP address scheme, patch scheme help to locate devices
- Automated system for backing up configs, diffs, rollback etc. (e.g. rancid)

Network Baseline

- = information about “normal” network behavior
- *Consists of*
  - Link and device performance statistics
    - can include basic performance statistics like
      - the interface load for critical network links
      - the CPU load and memory usage of routers and switches
    - these values can be polled and collected on a regular basis
  - Accounting of network traffic (RMON, NBAR, NetFlow)
  - Measurement of network performance characteristics (IP SLA)
    - measure critical performance indicators like delay and jitter across the network infrastructure

Backup Handling

Fundamental Maintenance Tools



CLI mgmt

Console
SSH
Telnet

GUI mgmt

HTTP
HTTPS
Telnet
SSH

NTP

Time

TFTP
FTP
SCP
HTTP(S)

Backups

Syslog
NetFlow
SNMP

Logging

Cisco Configuration and Documentation Tools

- **Dynamic Configuration Tool**
  - Aids in creating hardware configurations
  - Verifies compatibility of hardware and software selected
  - Produces a Bill of Materials (BoM) with part numbers
  - https://apps.cisco.com/qtc/config/html/configureHomeGuest.html
- **Cisco Feature Navigator**
  - Quickly finds Cisco IOS Software release for required features
  - http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp
- **SNMP Object Navigator**
  - Translates SNMP Object Identifiers (OID) into object names
  - Allows download of SNMP MIB files
  - Verify supported MIBs for a Cisco IOS Software version
  - http://tools.cisco.com/Support/SNMP/do/BrowseOID.do?local=en
- **Cisco Power Calculator**
  - Calculates power supply requirements a PoE hardware configuration
  - Requires CCO login

Network Time Protocol

- NTP specified in the RFC 5905, used to synchronize computer clocks in the Internet
- NTP uses hierarchy of servers. Accuracy of each server is defined by a number called the stratum
  - **Stratum 0**: Reference clock, e.g. atomic (cesium, rubidium) clocks, GPS clocks etc.
  - **Stratum 1**: NTP server whose system clocks are synchronized to within a few microseconds of their attached stratum 0 device
  - **Stratum N**: NTP server synchronized with NTP stratum N-1 server
- NTP is necessary for several reasons:
  - Key-chains - key expiration
  - Certificates – expiration
  - Logs – correlation logs from several devices

Backup and Restore using FTP

- Copy using FTP with stored username and password

```
R1(config)# ip ftp username backup
R1(config)# ip ftp password san-fran
R1(config)# exit
R1# copy startup-config ftp://10.1.152.1/R1-test.cfg
Address or name of remote host [10.1.152.1]?
Destination filename [R1-test.cfg]?
Writing R1-test.cfg !
2323 bytes copied in 0.304 secs (7641 bytes/sec)
```

- Copy using FTP with specified username and password

```
R1# copy startup-config ftp://backup:san-fran@10.1.152.1/R1-test.cfg
Address or name of remote host [10.1.152.1]?
Destination filename [R1-test.cfg]?
Writing R1-test.cfg !
2323 bytes copied in 0.268 secs (8668 bytes/sec)
```

Backup and Restore using HTTP/HTTPS

- Copy using HTTP with stored username and password

```
R1(config)# ip http client username backup
R1(config)# ip http client password san-fran
R1(config)# exit
R1# copy startup-config http://10.1.152.1/R1-test.cfg

! Or

R1# copy startup-config https://10.1.152.1/R1-test.cfg

Address or name of remote host [10.1.152.1]?
Destination filename [R1-test.cfg]?
Writing R1-test.cfg !
2323 bytes copied in 0.304 secs (7641 bytes/sec)
```

- Username or password can specified as a command line argument similarly to FTP

Backup and Restore using Archive

- Setting up the configuration archive

```
R1(config)# archive
R1(config-archive)# path flash:/config-archive/$h-config
R1(config-archive)# write-memory
R1(config-archive)# time-period 10080
```

- Verifying command output

```
R1# show archive
There are currently 3 archive configurations saved.
The next archive file will be named flash:/config-archive/R1-config-4
 Archive #  Name
   0
   1       flash:/config-archive/R1-config-1
   2       flash:/config-archive/R1-config-2
   5       flash:/config-archive/R1-config-3 <- Most Recent
```

Backup and Restore using `configure replace`

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# hostname TEST
TEST(config)# ^Z
TEST# configure replace flash:config-archive/R1-config-3 list
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: yes
!Pass 1
!List of Commands:
no hostname TEST
hostname RO1
end
Total number of passes: 1
Rollback Done
```

Tracking Changes in the Configuration

- Enable logging commands and sending them to syslog server

```
R1(config)# archive
R1(config-archive)# log config
R1(config-archive-log-cfg)# logging size 500
R1(config-archive-log-cfg)# hidekeys
R1(config-archive-log-cfg)# notify syslog
R1(config-archive-log-cfg)# logging enable
```

- Show changes

```
R1# show archive log config all

 idx   sess           user@line      Logged command
    1     1        console@console  |  logging enable
    2     1        console@console  |  exit
    3     1        console@console  |   exit
    4     1        console@console  |interface lo0
    5     1        console@console  | description => Local RID <=
    6     1        console@console  | ip address 192.0.2.1 255.0.0.0
    7     1        console@console  | exit
    8     2        console@console  |no ip domain lookup
```

Resilient Configuration

- Some attacks (and configuration attempts ☺) leads to IOS and configuration corruption
- Resilient configuration is protective feature available since 12.3(8)T
  - Backs up IOS and configuration to “invisible files” on flash
  - These files are not directly accessible via IOS commands and cannot be deleted through **`format`** or **`erase`**
  - They can be used to recover original IOS or configuration
  - Resilient Configuration cannot be remotely deactivated, only through console connection
  - Available on routers

Configuration of RC

- IOS backup:

```
Router(config)# secure boot-image
```

- Config backup:

```
Router(config)# secure boot-config
```

- Veryfing configuration:

```
Router# show secure [bootset]
```

- IOS recovery is done through ROMMON and `no secure boot-image`

- Configuration recovery is done with

```
Router(config)# secure boot-config restore cieľový-súbor
```

Disaster Recovery Tools

- Successful disaster recovery is dependent on the existence of the following:
  - Up to date configuration backups
  - Up to date software backups
  - Up to date hardware inventories
  - Configuration and software provisioning tools

MY COMPUTER DOESN'T WORK! THE HARD DRIVE CRASHED! WHAT DO I DO?!
DID YOU BACK UP?
WHY? IS IT GONNA BLOW?!

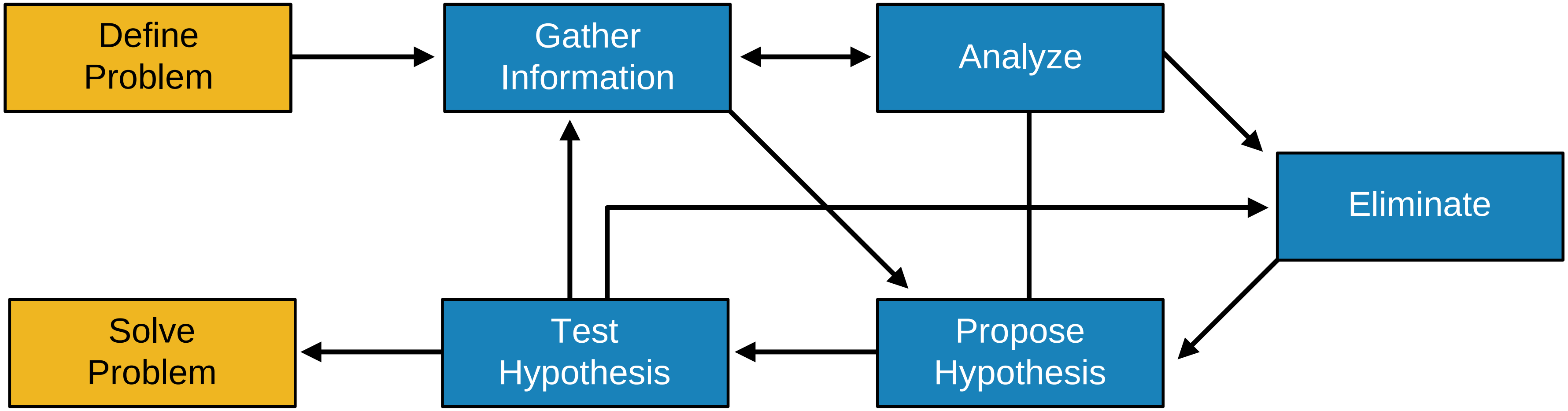
Troubleshooting Processes

Structured Approaches

- IF there is a problem THEN process starts in the head o troubleshooter
- **Top-down**
  - Troubleshoot from the application layer down to the physical layer
- **Bottom-up**
  - Troubleshoot from the physical layer up to the application layer
- **Divide and conquer**
  - Start in the middle of the OSI model, based on findings move up/down
- **Follow-the-path**
  - Follow the path that packets travels through the network
- **Spot the differences**
  - Check differences between working/not working device (e.g. configuration)
- **Move the problem**
  - Change a switch port / device, observe whether the problem moves

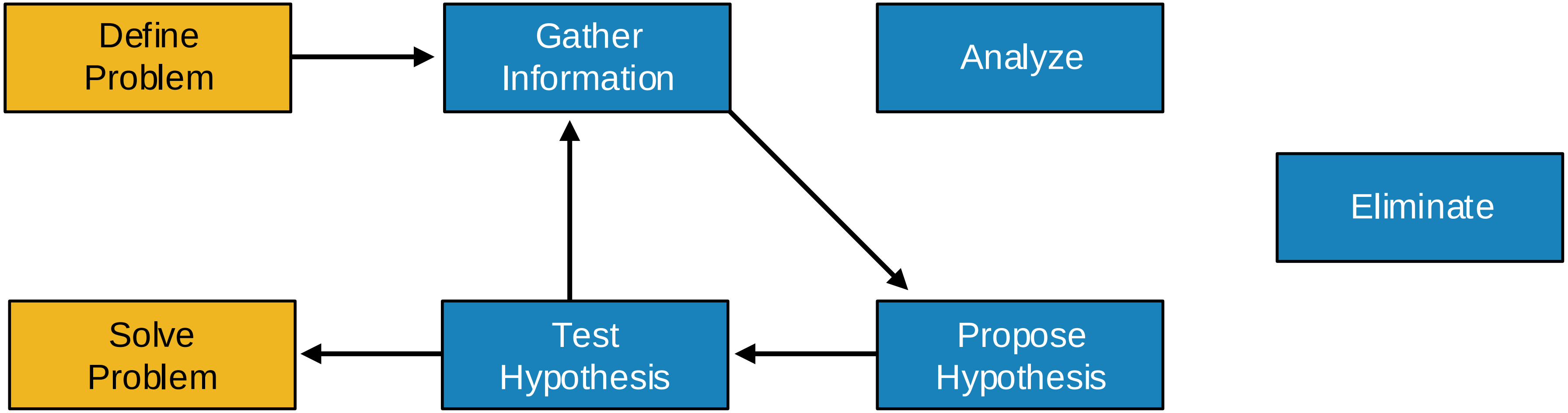
Structured Approach

- Independently on chosen approach it is mandatory to progress structurally and systematically

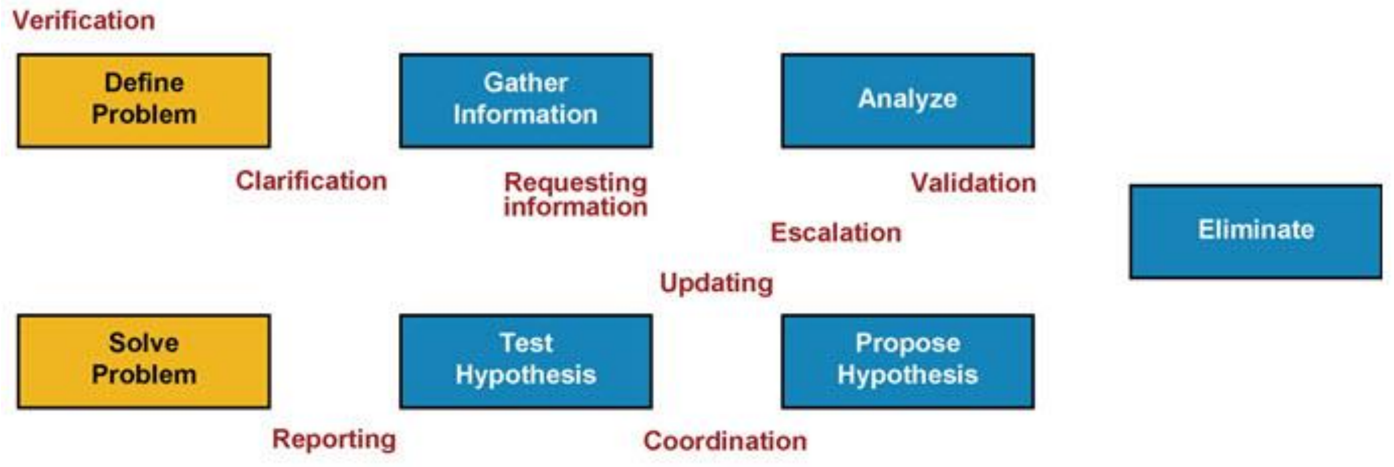


Shoot from the Hip Approach

- Short observation, quick change, observe solution
- Suitable for experienced troubleshooter



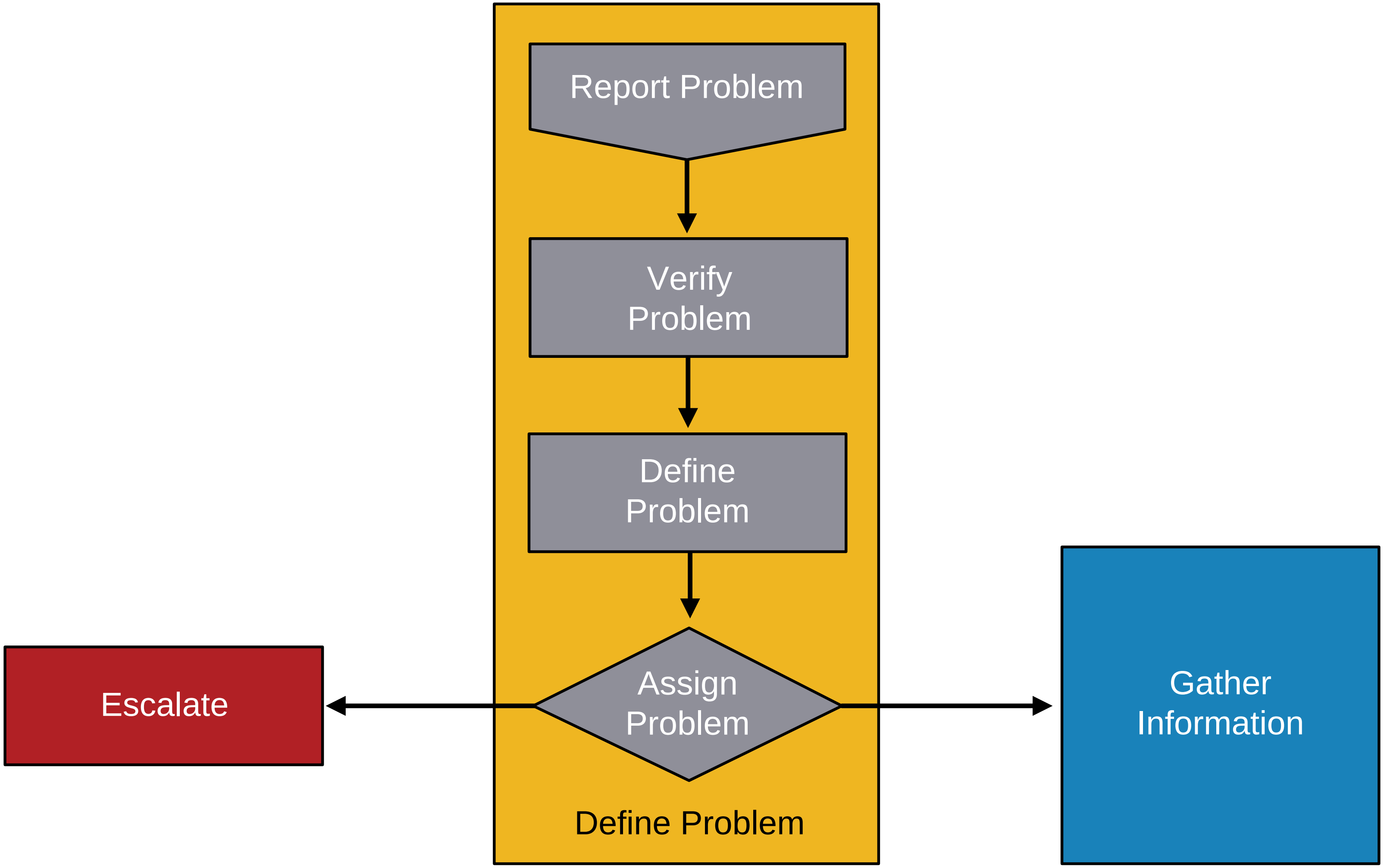
Communication



- Communication is an essential part of structured troubleshooting

1) Define Problem
   - Clarification is necessary. Asking good questions, carefully listening
2) Gather Information
   - Requesting information from others engineers or users
3) Analyze
   - Solitary process, however consultation with more experienced engineers is often useful
4) Propose and Test Hypothesis
   - Changes can be disruptive, users can be impacted. Communicate what you are doing and why you are doing it.
5) Solving Problem
   - Report back to the person who reported the problem.

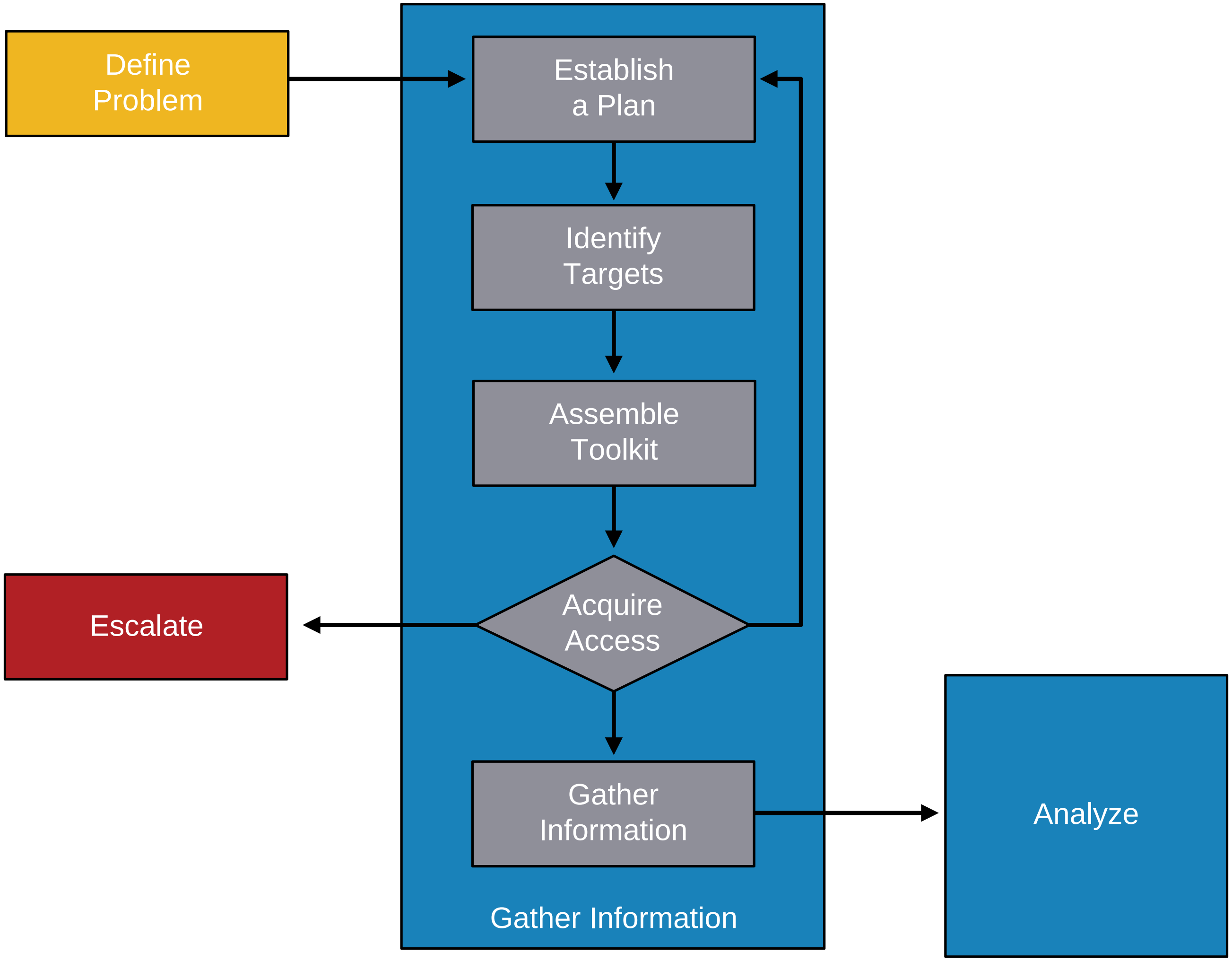
① Define the Problem



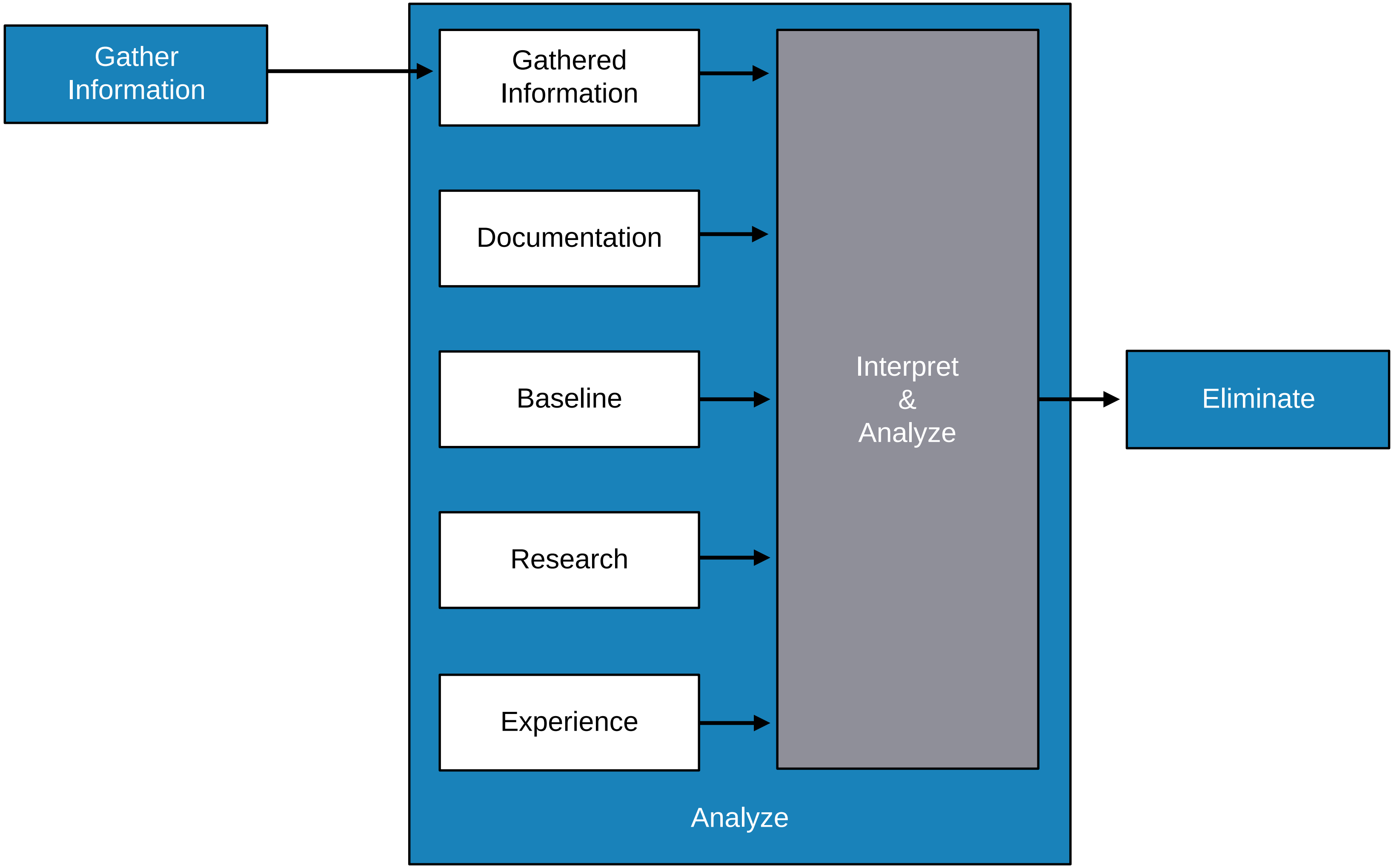
Verify Problem

- User usually reports symptoms not causes of problem
  - Symptom is only external manifestation of problem
  - However, to successfully solve problem means to get rid off the cause
  - Knowledge of protocols and technologies helps a lot
- Following questions are important for verification
  - *When the problem occured first?*
  - *Had it ever worked at all?*

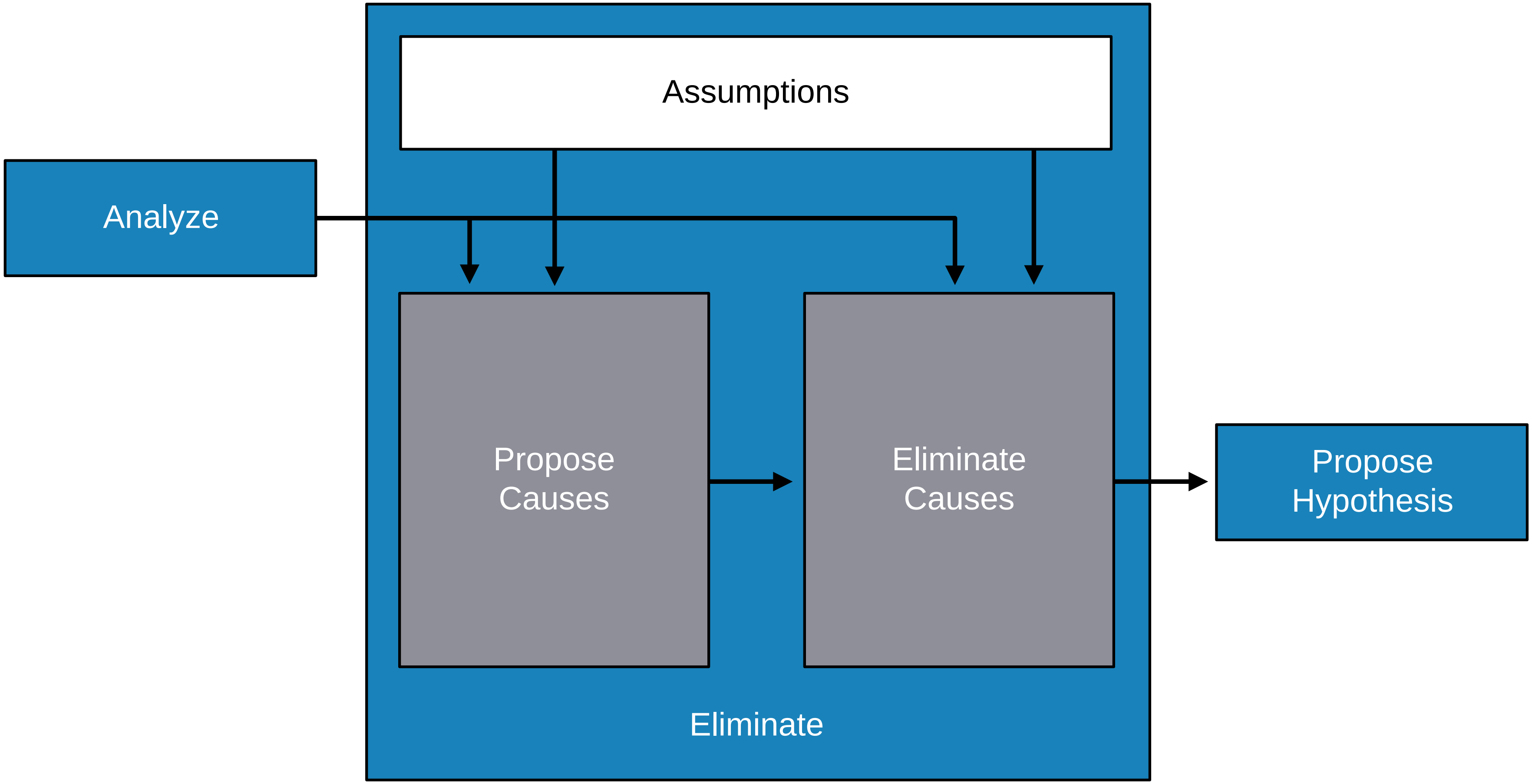
② Gather Information



③ Analyze


Gather Information
Gathered Information
Documentation
Baseline
Research
Experience
Interpret & Analyze
Analyze
Eliminate

④ Analyze



Assumptions

Analyze

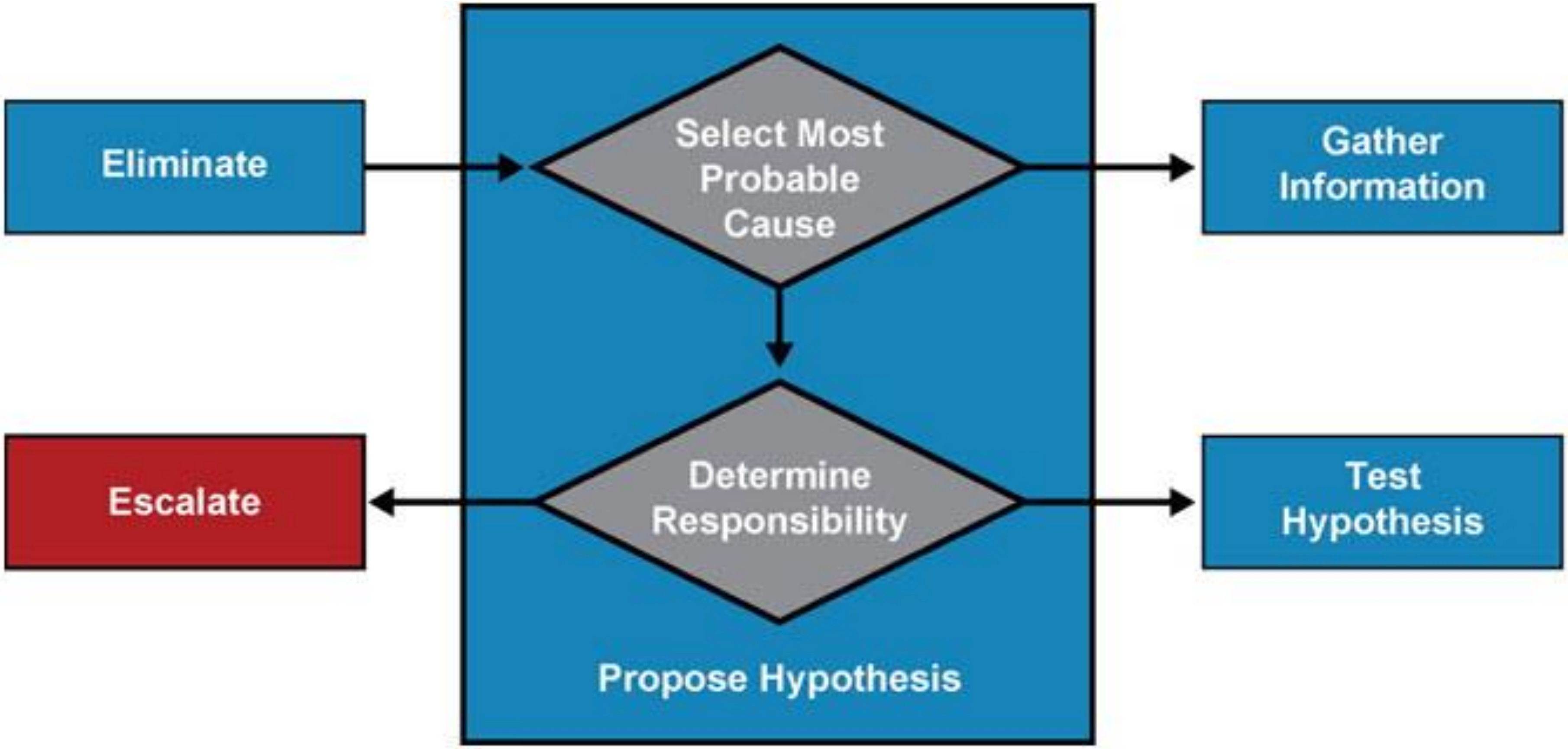
Propose Causes

Eliminate Causes

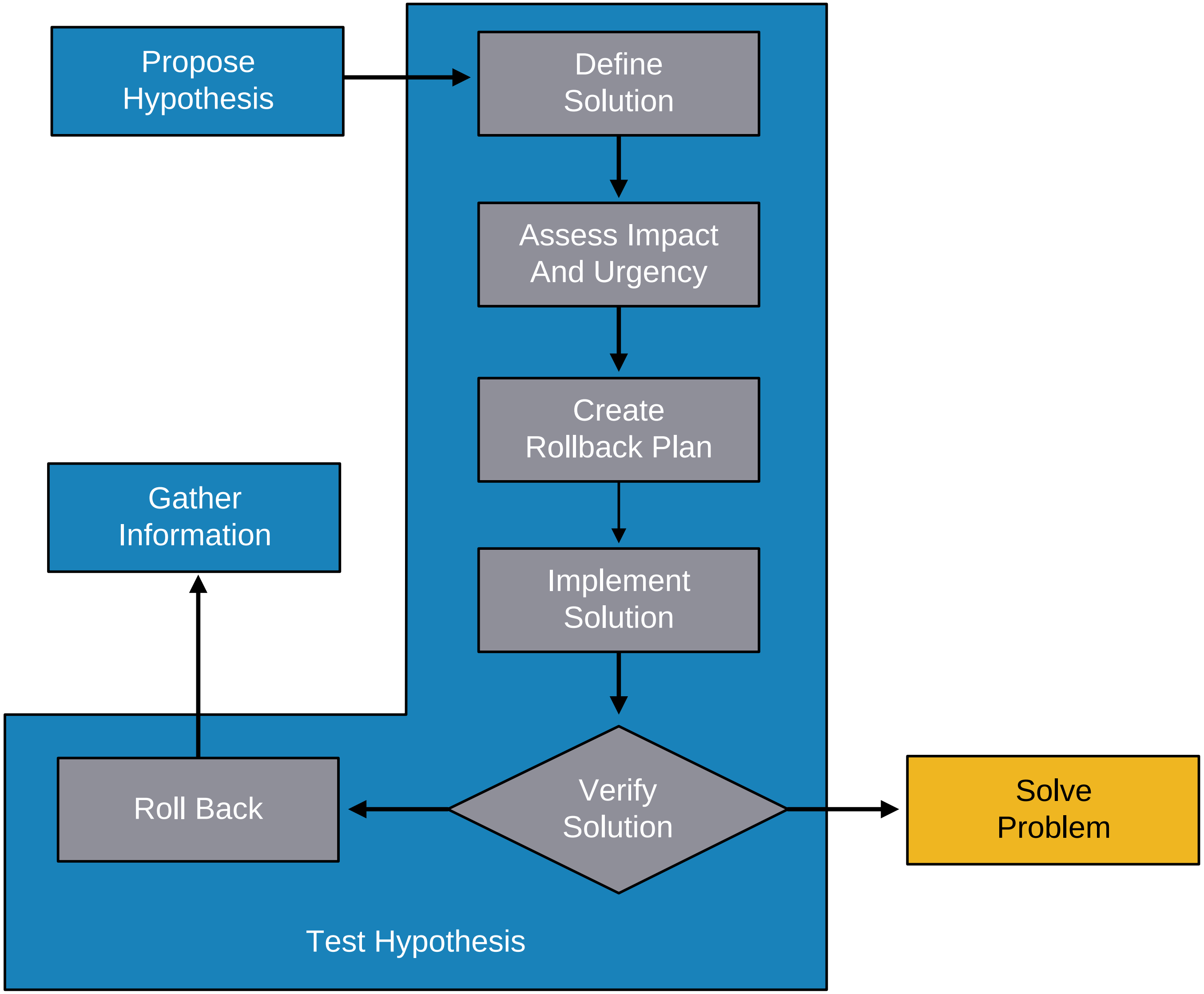
Propose Hypothesis

Eliminate

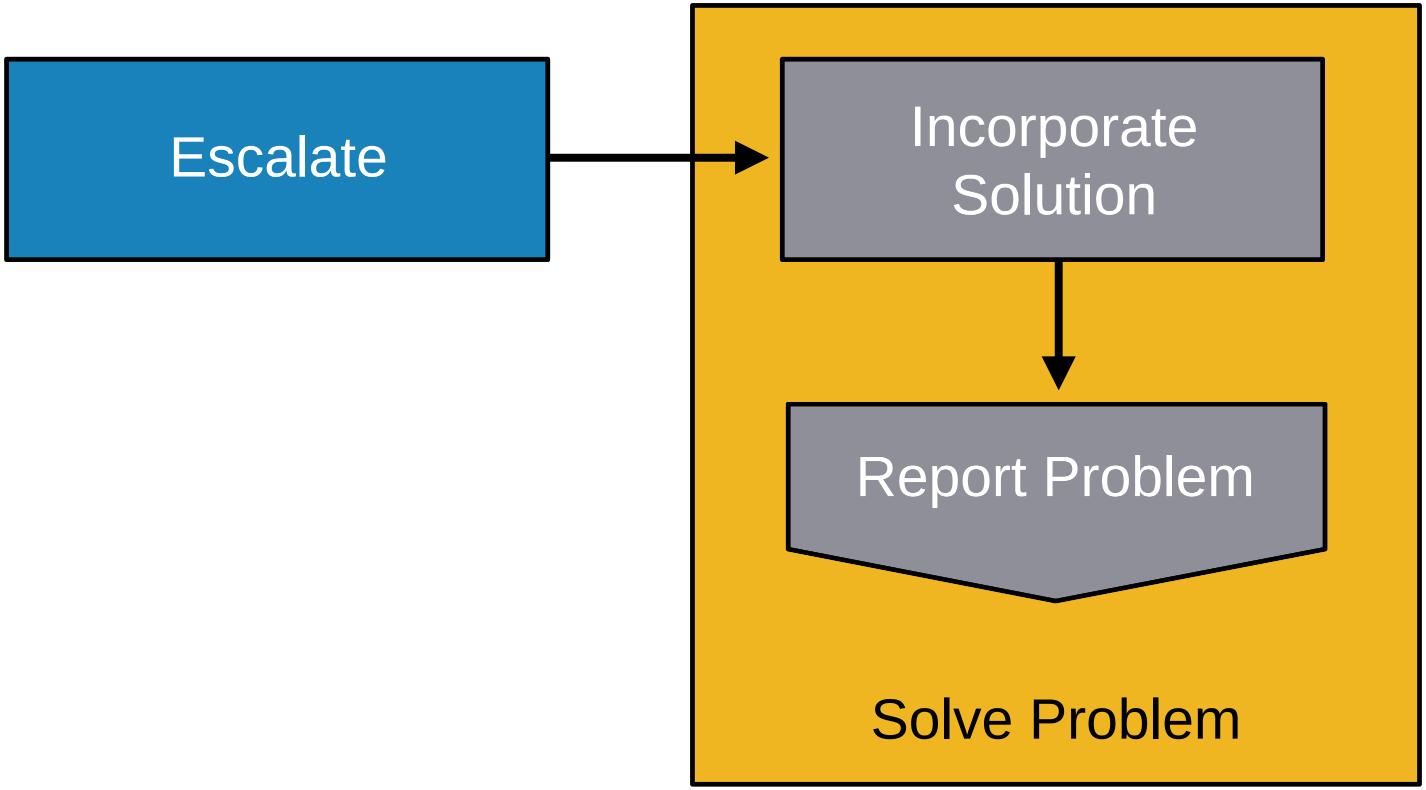
⑤ Propose Hypothesis



⑥ Test Hypothesis



⑦ Solve Problem



Spot the Differences Example

- Branch1 is in good working order

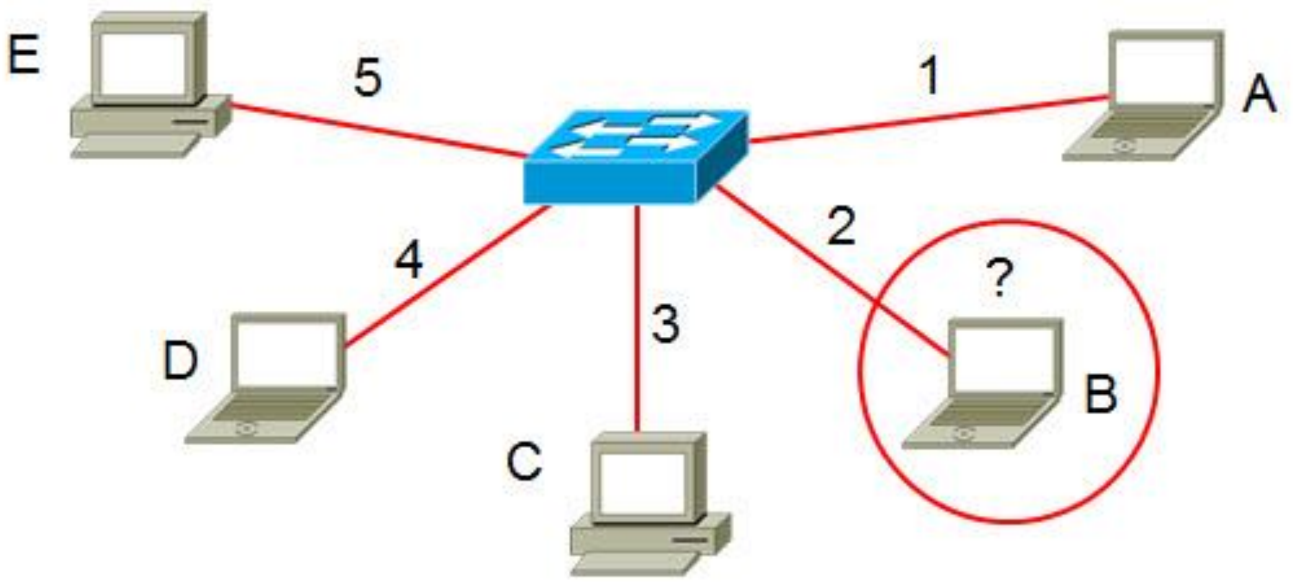
```
Branch1# show ip route
 <output omitted>
     10.0.0.0/24 is subnetted, 1 subnets
C       10.132.125.0 is directly connected, FastEthernet4
C    192.168.36.0/24 is directly connected, BVI1
S*   0.0.0.0/0 [254/0] via 10.132.125.1
```

- Branch2 has connectivity problems

```
Branch2# show ip route
 <output omitted>
     10.0.0.0/24 is subnetted, 1 subnets
C       10.132.125.0 is directly connected, FastEthernet4
C    192.168.36.0/24 is directly connected, BVI1
```

Move the Problem Example

- Laptop B is having network problems
  - Swap cable with the working device (e.g. laptop A)
  - Swap switch port
  - Replace switch



IOS Troubleshooting Tools

Tricks with `show ip route` ①

```
R1# show ip route 10.1.193.2
Routing entry for 10.1.193.0/30
  Known via "connected", distance 0, metric 0 (connected, via
interface)
  Redistributing via eigrp 1
  Routing Descriptor Blocks:
  * directly connected, via Serial0/0/1
      Route metric is 0, traffic share count is 1

R1# show ip route 10.1.193.10
% subnet not in table
```

```
R1# show ip route 10.1.193.0 255.255.255.0 longer-prefixes
< output omitted >
Gateway of last resort is not set
     10.0.0.0/8 is variably subnetted, 46 subnets, 6 masks
C       10.1.193.2/32 is directly connected, Serial0/0/1
C       10.1.193.0/30 is directly connected, Serial0/0/1
D       10.1.193.6/32 [90/20517120] via 10.1.192.9, 2d01h, FastEthernet0/1
                      [90/20517120] via 10.1.192.1, 2d01h, FastEthernet0/0
D       10.1.193.4/30 [90/20517120] via 10.1.192.9, 2d01h, FastEthernet0/1
                      [90/20517120] via 10.1.192.1, 2d01h, FastEthernet0/0
D       10.1.193.5/32 [90/41024000] via 10.1.194.6, 2d01h, Serial0/0/0.122
```

Tricks with show ip route ②

```
R1# show ip route
< output omitted >

     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Loopback0

R1# show ip route 192.168.1.0
Routing entry for 192.168.1.0/30, 1 known subnets
  Attached (1 connections)

C       192.168.1.0 is directly connected, Loopback0

R1# show ip route 192.168.1.0 255.255.255.252
Routing entry for 192.168.1.0/30
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Routing Descriptor Blocks:
  * directly connected, via Loopback0
      Route metric is 0, traffic share count is 1
```

Filtering of `show` Command ①

Using pipes with **`include`**, **`exclude`** and **`begin`**

```
R1# show processes cpu | include IP Input
  71      3149172    7922812         397   0.24%   0.15%   0.05%   0 IP Input

S1# show ip interface brief | exclude unassigned
Interface                  IP-Address       OK? Method Status                Protocol
Vlan128                    10.1.156.1       YES NVRAM  up                    up

S1# show running-config | begin line vty
line vty 0 4
 transport input telnet ssh
line vty 5 15
 transport input telnet ssh
!
End

R1# show processes cpu| include IP Input
                      ^
% Invalid input detected at '^' marker.
```

Filtering of show Command ②

Using pipes with **section** and **^**

```
R1# show running-config | section router eigrp
router eigrp 1
 network 10.1.192.2 0.0.0.0
 network 10.1.192.10 0.0.0.0
 network 10.1.193.1 0.0.0.0
 no auto-summary

R1# show processes cpu | include ^CPU|IP Input
CPU utilization for five seconds: 1%/0%; one minute: 1%; five minutes: 1%
  71   3149424   7923898     397  0.24%  0.04%  0.00%   0 IP Input
```

Collecting with show Command ①

Using the **redirect** and **tee** options

```
R1# show tech-support | redirect tftp://192.168.37.2/show-tech.txt

R1# show ip interface brief | tee flash:show-int-brief.txt
Interface                  IP-Address      OK? Method Status
Protocol
FastEthernet0/0            10.1.192.2      YES manual up                    up
FastEthernet0/1            10.1.192.10     YES manual up                    up
Loopback0                  10.1.220.1      YES manual up                    up

R1# dir flash:
Directory of flash:/
 1 -rw-  23361156  Mar 2 2009 16:25:54 -08:00 c1841-advipservicesk9mz.1243.bin
 2 -rw-       680  Mar 7 2010 02:16:56 -08:00 show-int-brief.txt
```

Collecting with show Command ②

Using the **append** option and the **more** command

```
R1# show version | append flash:show-commands.txt

R1# show ip interface brief | append flash:show-commands.txt

R1# more flash:show-commands.txt
Cisco IOS Software, 1841 Software (C1841-ADVIPSERVICESK9-M), Version 12.4(23),
RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2008 by Cisco Systems, Inc.
Compiled Sat 08-Nov-08 20:07 by prod_rel_team
ROM: System Bootstrap, Version 12.3(8r)T9, RELEASE SOFTWARE (fc1)
R1 uptime is 3 days, 1 hour, 22 minutes
< output omitted >
Interface                  IP-Address      OK? Method Status
Protocol
FastEthernet0/0            10.1.192.2      YES manual up                    up
FastEthernet0/1            10.1.192.10     YES manual up                    up
```

Pinging ①

```
Router# ping ip-address | hostname [repeat repeat-count
size datagram-size source [address | interface] df-bit]
```

Parameter	Description

repeat
repeat-count
	Number of ping packets that are sent to the destination address. The default is 5.

size *datagram-size*	Size of the ping packet (in bytes). Default: 100 bytes.
source **[***address* **|** *interface***]**	The interface or IP address of the router to use as a source address for the probes.
df-bit	Enables the "do-not-fragment" bit in the IP header.

Pinging ②

Using the ping extended option: source

```
R1# ping 10.1.156.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.156.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

R1# ping 10.1.156.1 source FastEthernet 0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.156.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.192.2
.....
Success rate is 0 percent (0/5)
```

Pinging ③

Using the ping extended option: **df-bit**

```
R1# ping 10.1.221.1 size 1476 df-bit
Type escape sequence to abort.
Sending 5, 1476-byte ICMP Echos to 10.1.221.1, timeout is 2 seconds:
Packet sent with the DF bit set
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 184/189/193 ms

R1# ping 10.1.221.1 size 1477 df-bit
Type escape sequence to abort.
Sending 5, 1477-byte ICMP Echos to 10.1.221.1, timeout is 2 seconds:
Packet sent with the DF bit set
M.M.M
Success rate is 0 percent (0/5)
```

Pinging ④

Explanation of ping results characters

- **!** Each exclamation point indicates receipt of a reply.
- **.** Each period indicates a timeout waiting for a reply.
- **U** A destination unreachable ICMP message was received.
- **Q** Source quench (destination too busy).
- **M** Could not fragment (MTU related).
- **?** Unknown packet type.
- **&** Packet lifetime exceeded

Pinging ⑤

Using the **ping** extended prompt mode

```
R1# ping
Protocol [ip]:
Target IP address: 10.1.221.1
Repeat count [5]: 1
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface:
Type of service [0]:
Set DF bit in IP header? [no]: yes
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]: y
Sweep min size [36]: 1400
Sweep max size [18024]: 1500
Sweep interval [1]:
Type escape sequence to abort.
Sending 101, [1400..1500]-byte ICMP Echos to 10.1.221.1, timeout is 2 seconds:
<output omitted>
```

Testing Network Connectivity

Using Telnet to test the Transport and Application Layer

```
R1# telnet 192.168.37.2 80
Trying 192.168.37.2, 80 ... Open
GET
<html><body><h1>It works!</h1></body></html>
[Connection to 192.168.37.2 closed by foreign host]


R1# telnet 192.168.37.2 25
Trying 192.168.37.2, 25 ...
% Connection refused by remote host
```

Collecting Real-time Information

The **debug ip packet** command output

```
R1# debug ip packet
IP: s=172.69.13.44 (Fddi0), d=10.125.254.1 (Serial2), g=172.69.16.2, forward
IP: s=172.69.1.57 (Ethernet4), d=10.36.125.2 (Serial2), g=172.69.16.2, forward
IP: s=172.69.1.6 (Ethernet4), d=255.255.255.255, rcvd 2
IP: s=172.69.1.55 (Ethernet4), d=172.69.2.42 (Fddi0), g=172.69.13.6, forward
IP: s=172.69.89.33 (Ethernet2), d=10.130.2.156 (Serial2), g=172.69.16.2,
forward
IP: s=172.69.1.27 (Ethernet4), d=172.69.43.126 (Fddi1), g=172.69.23.5, forward
IP: s=172.69.1.27 (Ethernet4), d=172.69.43.126 (Fddi0), g=172.69.13.6, forward
IP: s=172.69.20.32 (Ethernet2), d=255.255.255.255, rcvd 2
IP: s=172.69.1.57 (Ethernet4), d=10.36.125.2 (Serial2), g=172.69.16.2, access
denied
```

Collecting Real-time Information

The **debug ip rip** command output

```
R2# debug ip rip
RIP: received v2 update from 10.0.23.3 on FastEthernet0/1
     10.0.3.0/24 via 0.0.0.0 in 1 hops
RIP: received v2 update from 10.0.12.1 on FastEthernet0/0
     10.0.1.0/24 via 0.0.0.0 in 1 hops
RIP: sending v2 update to 224.0.0.9 via FastEthernet0/1 (10.0.23.2)
<output omitted>

R2# debug condition interface fa0/1
Condition 1 set
RIP: sending v2 update to 224.0.0.9 via FastEthernet0/1 (10.0.23.2)
RIP: build update entries
        10.0.1.0/24 via 0.0.0.0, metric 2, tag 0
        10.0.2.0/24 via 0.0.0.0, metric 1, tag 0
        10.0.12.0/24 via 0.0.0.0, metric 1, tag 0
RIP: received v2 update from 10.0.23.3 on FastEthernet0/1
     10.0.3.0/24 via 0.0.0.0 in 1 hops
<output omitted>
```

Traffic Forwarding to the CPU

- Traffic being punted to the CPU is indirect proof of TCAM allocation failures or use of unsupported features
- The **show controllers cpu-interface** displays the statistics for packets that are forwarded by CPU

```
Switch# show controllers cpu-interface
ASIC     Rxbiterr    Rxunder     Fwdctfix    Txbuflos    Rxbufloc    Rxbufdrain
-------------------------------------------------------------------------
ASIC0      0           0           0           0           0           0

cpu-queue-frames  retrieved  dropped    invalid    hol-block  stray
----------------- ---------- ---------- ---------- ---------- ----------
rpc               0          0          0          0          0
stp               1          0          0          0          0
ipc               0          0          0          0          0
routing protocol  28312      0          0          0          0
L2 protocol       0          0          0          0          0
remote console    0          0          0          0          0
sw forwarding     13800556   0          0          0          0
host              7648       0          0          0          0
broadcast         462103     0          0          0          0
cbt-to-spt        0          0          0          0          0
igmp snooping     35916      0          0          0          0
icmp              0          0          0          0          0
logging           0          0          0          0          0
rpf-fail          0          0          0          0          0
dstats            0          0          0          0          0
cpu heartbeat     22302361   0          0          0          0
```

CPU Problems

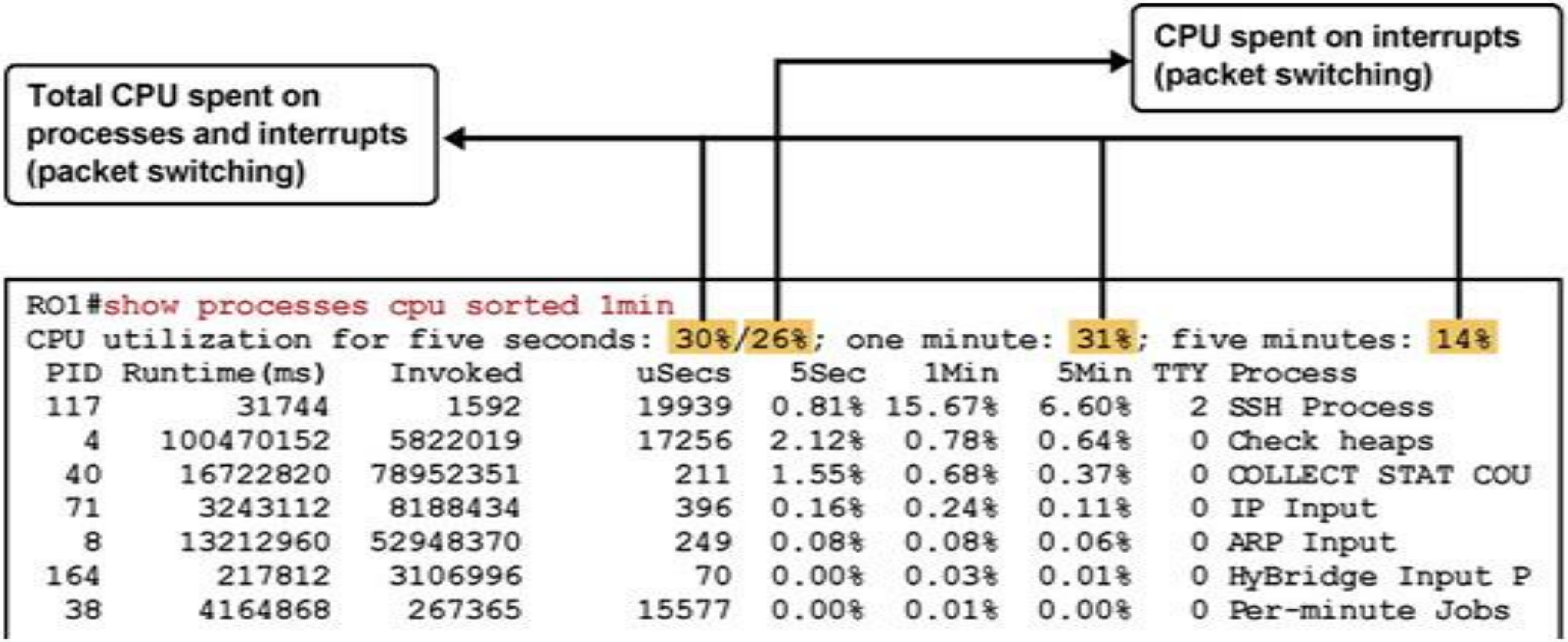
- First, determine whether interrupts or processes are the major cause of the increased CPU load
  - `IF` case of interrupts `THEN` troubleshoot packet forwarding and TCAM
  - `IF` case of processes `THEN` isolate responsible process and troubleshoot based on outcome
- In general, an average CPU load if 50% is not problematic just same as temporary 100% bursts
- Spikes in load could be caused by
  - Processor-intensive commands such as show tech-support, debug, show running, copy run start
  - Routing protocol updates
  - SNMP Polling

Diagnosing Hardware Issues

Checking CPU utilization with **show processes cpu**

Important processes are

- IP Input
- IP ARP
- SNMP Engine
- IGMPSN



Total CPU spent on processes and interrupts (packet switching)

CPU spent on interrupts (packet switching)

```
R01#show processes cpu sorted 1min
CPU utilization for five seconds: 30%/26%; one minute: 31%; five minutes: 14%
 PID Runtime(ms)     Invoked      uSecs   5Sec   1Min   5Min TTY Process
 117       31744        1592      19939  0.81% 15.67%  6.60%   2 SSH Process
   4   100470152     5822019      17256  2.12%  0.78%  0.64%   0 Check heaps
  40    16722820    78952351        211  1.55%  0.68%  0.37%   0 COLLECT STAT COU
  71     3243112     8188434        396  0.16%  0.24%  0.11%   0 IP Input
   8    13212960    52948370        249  0.08%  0.08%  0.06%   0 ARP Input
 164      217812     3106996         70  0.00%  0.03%  0.01%   0 HyBridge Input P
  38     4164868      267365      15577  0.00%  0.01%  0.00%   0 Per-minute Jobs
```

Diagnosing Hardware Issues

Checking memory utilization with the **show memory** command

```
R1# show memory
               Head   Total(b)     Used(b)      Free(b)    Lowest(b)  Largest(b)
Processor  820B1DB4   26534476    19686964      6847512      6288260     6712884
      I/O   3A00000    6291456     3702900      2588556      2511168     2577468
```

Diagnosing Hardware Issues

Checking interfaces with the **show interfaces** command

```
R1# show interfaces FastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
<output omitted>
  Last input 00:00:00, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/1120/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 2000 bits/sec, 3 packets/sec
  5 minute output rate 0 bits/sec, 1 packets/sec
     110834589 packets input, 1698341767 bytes
     Received 61734527 broadcasts, 0 runts, 0 giants, 565 throttles
     30 input errors, 5 CRC, 1 frame, 0 overrun, 25 ignored
     0 watchdog
     0 input packets with dribble condition detected
     35616938 packets output, 526385834 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

Diagnosing Hardware Issues

Additional hardware commands and tools:

- **`show controllers`**
- **`show platform`**
- **`show inventory`**
- **`show diag`**
- Generic Online Diagnostics (GOLD)
- Time Domain Reflectometer

Slides adapted by Vladimír Veselý and Matěj Grégr
partially from official course materials
but the most of the credit goes to CCIE#23527 Ing. Peter Palúch, Ph.D.

Last update: 2017-03-06