

Workbook Overview

Troubleshooting becomes integral part of the updated CCIE R&S lab exam. The new section is a group of loosely correlated trouble tickets. Every ticket has a point value associated with it, and the candidate must obtain 80% of the total section score in order to succeed in this section. Troubleshooting scenario uses a topology separate from the configuration part of the exam and has its own L2 configuration and IP addressing. Section grading is based on the automatic script along with a human hand to confirm the script results. From this information, one may conclude that mastering troubleshooting techniques becomes vital for succeeding in the new exam.

In this new IEWB-RS VOL4 workbook we present you with ten troubleshooting scenarios, each having ten trouble tickets. This amount should be approximately equal to the number of the troubleshooting tasks you will encounter in the actual exam. The topology used for every scenario is the same that we use for all our RS products, including VOL1 (technology-focused labs), VOL2 (configuration mock lab scenarios) and VOL3 (core technologies scenarios).

However, unlike our previous workbooks, we **restrict** access to some of the devices in the lab topology. For every scenario this “restricted” set may be different and it is clearly outlined in the scenario’s baseline. Using this technique we increase the scenario complexity by allowing candidates to see only “one” side of the problem. When looking at the lab diagram, you will clearly see routers not under your control as being displayed in orange color. Also, when you log onto the “restricted” device, it will warn you using a banner message.

In addition to the above restriction, we highly encourage you not using the **show running-configuration**, **show startup-configuration** commands or any other command that shows you the textual representation of the router’s configuration. This requirement makes you focus on using the show and debugging commands, which is invaluable when troubleshooting the real-world scenarios.

Our ultimate goal is not only prepare you for passing the Troubleshooting section of the CCIE R&S lab exam, but also to teach you a structured troubleshooting approach. As opposed to simple guessing and peeking at the routers running configurations you should learn using the debugging commands and interpreting various show commands output. For every ticket, we are going to follow the same structured procedure to resolve the issue. Here is an outline of this procedure:

1. Build and Analyze the Baseline
2. Analyze the Symptoms (propose hypothesis)
3. Isolate the issue (gather more symptoms)
4. Fix the Issue (by comparing to the Baseline)

We are now going to discuss all these steps in details to give you the basic understanding of the fundamental procedure.

Structured Troubleshooting

Build and Analyze the Baseline.

Since all tickets in a scenario share the same topology, you need to perform this step only once per the whole scenario. Baseline is essentially a picture of the healthy network, which serves as the starting point of any troubleshooting process. In real life, your baseline is the snapshot of your network under “normal” conditions – stable topology, interfaces under normal utilization, devices responding to management requests, users happy etc. In the lab, all you have is the diagram and possibly some additional network description. Additional information might be provided in the trouble ticket itself, but the initial starting point is the diagram.

We recommend making your own diagrams, including the following information:

- IP addressing + IGPs.
- Layer 2 topology.
- BGP diagram.
- IPv6 topology.
- Multicast and Redistribution diagram.

You may enhance your diagrams with any extra information provided, e.g. hints on the network pre-configuration and applications deployed, such as WWW, FTP, SMTP, VoIP and so on. This will help you analyzing symptoms later. Your goal at this stage is to get clear picture of the network and discover any potential caveats. Try not using any IOS commands at this point, as this may consume your valuable time and add unneeded information. Overall, don't spend too much time building the baseline – the goal is to spend around 20 minutes. By the end of the baseline analysis phase, you should have clear understanding of the protocols and applications deployed in your network.

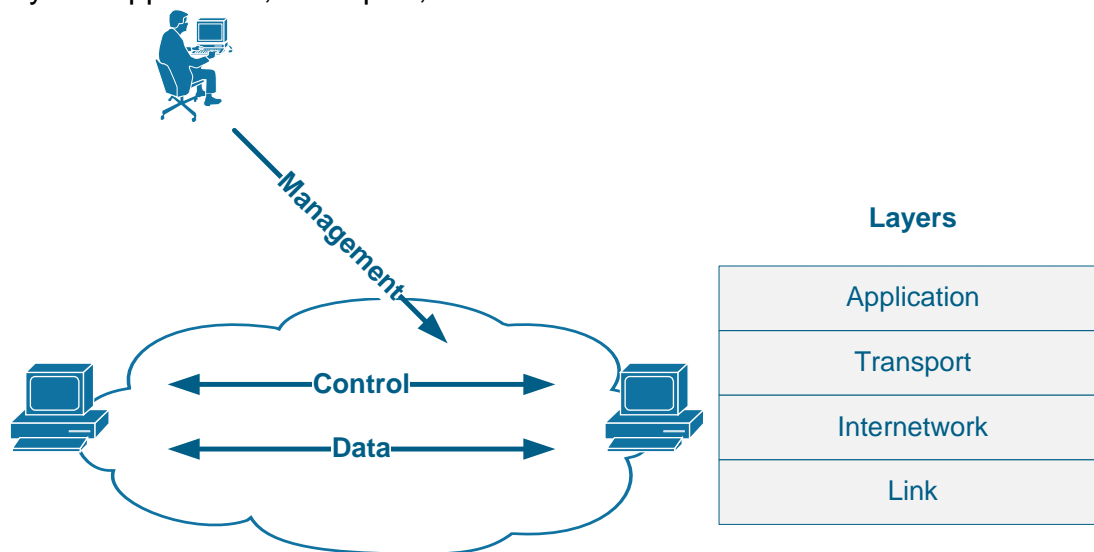
When you finished with building the baseline, take a quick look over all tickets in sequence. See if you can make any conclusions based on the ticket information,

like marking the potentially broken links or missing information flows. Sometimes this may be obvious from the ticket text and give you extra hint and help when dealing with other tickets.

Analyze the Symptoms.

The ultimate goal of this step is coming up with the initial scope of the problem area and the initial set of hypothesis identifying the root cause. With respect to the CCIE lab, the primary source of the information is trouble ticket itself. The ticket might be formatted in a very simple manner, such as “there is an issue that prevents R1 from communicating with R2” or contain a detailed situation description, for example “at 10:00am this morning customers at Branch 1 site started complaining of poor HTTP performance. Analyzing the NOC action logs, you have noticed that someone was modifying R3’s configuration yesterday, but the change log entry is missing” and so on.

When trying to narrow the initial problem scope, it is helpful to use a reference network model, based on the classic TCP/IP protocol layers. The minimal working network consists of two communicating nodes and communication substrate connecting them. The substrate might be a direct link or a set of other nodes/routers. The network functions in three general planes: data, control and management. The first plane is responsible for forwarding data between the two endpoints, based on the programmed tables. The control plane is responsible for negotiating the data paths and establishing end-to-end connectivity. Example control plane protocols are OSPF, RIP, BGP and so on. The management plane is responsible for enforcing certain policy on the network and monitoring its performance. For example, SNMP and SSH used to carry CLI commands are both management protocols. All planes could be subdivided into four classic layers: Application, Transport, Internetwork and Link.



When analyzing the symptoms, you should select the network elements (e.g nodes, links) that you suspect to belong to the problematic area: e.g. routers on the path between two nodes failing to communicate. At the same time, you should mark the planes and the layers that you suspect to malfunction and possibly identify the protocol names at every layer. You need to be aware of the symptoms typical for every layer and remember that an issue at lower layer

affects all other overlying layers as well (cascading effect).

Lastly, the most helpful thing to identify the initial problematic area is finding out what prior changes might have been made to the network, if this is mentioned in the ticket. Remember, every change is a potential problem!

Isolate the issue.

The goal of this phase is finding the device/devices and the specific configuration area that might be causing the issue(s). This is the core of the troubleshooting process. You start verifying your initial hypothesis, by trying to narrow the problematic area as small as possible. To start with the process, you need to select either of the three approaches:

Top-down approach

Test application layers across the path that you suspect to be causing the issues. Usually this approach works well when the issue lying in application misconfiguration (e.g. improper IMAP4 settings). This is very helpful in real-life scenarios; however, from the lab perspective this approach is not very useful as most issues will probably be related to the network configuration.

Bottom-up approach

You start by testing physical layer issues of every node in the problematic area. If you don't find any issues, you proceed to the next layer (i.e. networking) and see if there are any deviations from the baseline there. This is the most universal approach, as it starts with the fundamental layer and moves up. However, executing bottom-up search might be routine and time consuming, and thus inappropriate for a small issue.

Divide-and-Conquer approach

This method attempts to reduce the amount of work required by bottom-up search by making a "guess" – picking up the network layer that you suspect to be malfunctioning and testing the devices in the problematic area at this layer. It is common to start with the Internetwork layer and test end-to-end connectivity using the `ping` and `tracert` commands. If this layer is healthy, then any underlying layer should be healthy as well, and you may continue searching in the "up" direction. Otherwise, using the above mentioned commands you may further isolate the problematic area and find the specific devices that might be causing the issue.

It is important to remember that during the issue isolating phase you will learn more information and may have to change your initial hypothesis, based on the results. Effectively, the Analyze and Isolate phases are deeply interconnected and depend on each other.

Fix the Issue

At the end of the previous stage you should be dealing with the “hot” area of the problem – devices/links that are malfunctioning or improperly configured. Of course, you should have facts on hands to prove that your hypothesis/guess was valid. Your next step is developing a plan to resolve the problem. Resist the urge or simply going ahead and changing the running configuration – you may effectively introduce more issues than there originally was. Save the original configuration, and type in your “fixup” in the notepad. Implement the “fixup” step by step – don’t apply changes to many devices at the same time, if you suspect many devices being affected After every change, run verifications to see if the issue has been eliminated or not.

When you’re done, compare the results to the baseline you have built at the first step. If everything seems to match and the symptoms outlined in the ticket no longer persist you may consider the ticket to be resolved. If not, you should re-analyze the initial symptoms and the additional information gathered during the previous steps. The last step could be named as “Verification” step.

Workbook Solutions

Every solution document is formatted in structured manner to show you the flow of the actual troubleshooting process. You will find the sections corresponding to the in-depth analysis of the scenario baseline, diagram drawing and detailed step-by-step troubleshooting for every ticket presented in the scenario. Here is an outline for the solution document structure:

Build and Analyze the Baseline

- Layer 2 Diagram.
- BGP Diagram.
- Multicast and Redistribution.
 - Redistribution Loops Analysis.
 - Multicast Propagation Analysis.
- IPv6 Diagram
- Read over the Lab

Solutions

- Ticket 1
 - Analyze the Symptoms
 - Isolate the Issue
 - Fix the issue
 - Verify
 - ...

Ticket 10

Analyze the Symptoms

Isolate the Issue

Fix the issue.

Verify

As you can see, the document follows the exact same path for the troubleshooting process that we outlined before. Every solution is about 50-60 pages long and provides enough details for every ticket, so that you'll have plenty of material to learn from.

Sample Ticket

Ticket 2: Load-Balancing

- Three switches: SW2, SW3 and SW4 were configured so that traffic from SW1 load-balances to VLAN9 across the links connecting SW2 to SW3 and SW4.
- However, recently you have found that only the path via SW4 is being utilized.
- Accessing the devices under your control only, return the network to the baseline and ensure proper load-balancing.

3 Points

Ticket 2 Solution

Analyze the Symptoms

This is a typical end-to-end problem, where one node cannot communicate to another in proper manner. Let's see what our routing table at SW2 shows up:

```
Rack1SW2#show ip route 164.1.9.0
Routing entry for 164.1.9.0/24
  Known via "ospf 1", distance 110, metric 21, type intra area
  Redistributing via eigrp 100
  Advertised by eigrp 100 metric 10000 1000 1 255 1500 route-map OSPF-
>EIGRP
  Last update from 164.1.24.10 on FastEthernet0/21, 00:05:04 ago
  Routing Descriptor Blocks:
  * 164.1.24.10, from 150.1.9.9, 00:05:04 ago, via FastEthernet0/21
    Route metric is 21, traffic share count is 1
```

Apparently, the link is being advertised by SW3 and SW2 uses the path via SW4 to reach it. The problem scope includes three devices: SW2, SW3 and SW4. There are two things that come to mind immediately: something is wrong with the link between SW2 and SW4 or link metric are not configured properly.

Hypothesis 1: Issues with the link between SW2 and SW4.

Hypothesis 2: Improper metric configuration.

Initial problem scope: SW2, SW3 and SW4.

Isolate the Issue

Using the “divide-and-conquer” approach we start by testing the EtherChannel link health. This will probe our first hypothesis.

```
Rack1SW2#ping 164.1.23.3
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 164.1.23.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 109/114/118
ms
```

OK this rules out most physical issues. The next obvious check is seeing if we have OSPF neighbors on the link:

```
Rack1SW2#show ip ospf neighbor
```

```
Neighbor ID      Pri   State                    Dead Time   Address
Interface
150.1.9.9        1    FULL/BDR                 00:00:32   164.1.32.9   Port-
channel23
150.1.10.10     1    FULL/DR                  00:00:31   164.1.24.10  FastEthernet0/21
```

The first hypothesis does not appear to be valid so far as even the neighbors came up. Let’s see if we have any luck researching the second guess – the mismatched metrics. Let’s see the OSPF metric over the active path:

```
Rack1SW2#show ip route 164.1.9.0 | inc metric
  Known via "ospf 1", distance 110, metric 21, type intra area
  Advertised by eigrp 100 metric 10000 1000 1 255 1500 route-map OSPF-
>EIGRP
    Route metric is 21, traffic share count is 1
```

We record down the number of 21. Now let’s check what path SW4 chooses to reach VLAN9:

```
Rack1SW2#traceroute 164.1.9.9
```

```
Type escape sequence to abort.
Tracing the route to 164.1.9.9

 1 164.1.24.10 0 msec 8 msec 0 msec
 2 164.1.43.9 0 msec * 0 msec
```

This appears to be good, as SW4 prefer the direct link to SW3. Now let’s compare the cost of the direct link between SW2 and SW4 to the cost of the path being used now:

```
Rack1SW2#show ip ospf interface port-channel 23 | inc Cost
  Process ID 1, Router ID 150.1.8.8, Network Type BROADCAST, Cost: 20
```

So it's 20 and the cost to reach VLAN9 via SW4 is 21. Most likely the cost of VLAN9 interface is simply "1". To make sure it is, we do the following check – look for all links advertised by SW3 and see their metrics

```
Rack1SW2#show ip ospf database router adv-router 150.1.9.9
```

```
      OSPF Router with ID (150.1.8.8) (Process ID 1)
```

```
          Router Link States (Area 38)
```

```
LS age: 1952
Options: (No TOS-capability, DC)
LS Type: Router Links
Link State ID: 150.1.9.9
Advertising Router: 150.1.9.9
LS Seq Number: 80000011
Checksum: 0xF5A
Length: 96
Number of Links: 6
```

```
<snip>
```

```
Link connected to: a Stub Network
(Link ID) Network/subnet number: 164.1.9.0
(Link Data) Network Mask: 255.255.255.0
Number of TOS metrics: 0
TOS 0 Metrics: 1
```

```
<snip>
```

It appears to be that both paths – via SW4 and direct - have the same cost to reach VLAN9 from SW2. Thus, our both hypothesis appear to be invalid. Here is the situation that we have now:

- No link errors preventing OSPF neighbor adjacencies from coming up.
- No mismatching link metrics that prevent equal cost load-balancing
- OSPF selects just one path during SPF for some reason

The next step would be trying to look at the OSPF topology and see why it does not include direct path into calculations or selection. We are going to look at every router's (SW2, SW3 and SW4) router type LSA and see if there is anything wrong with those. We start with SW2 – notice that in the output that follows only relevant links are shown, the links that connect SW2 to SW3 and SW4:

```
Rack1SW2#show ip ospf database router self-originate
```

```
OSPF Router with ID (150.1.8.8) (Process ID 1)
```

```
Router Link States (Area 38)
```

```
LS age: 887
Options: (No TOS-capability, DC)
LS Type: Router Links
Link State ID: 150.1.8.8
Advertising Router: 150.1.8.8
LS Seq Number: 8000000C
Checksum: 0xFA86
Length: 48
AS Boundary Router
Number of Links: 2
```

```
Link connected to: a Transit Network
(Link ID) Designated Router address: 164.1.32.8
(Link Data) Router Interface address: 164.1.32.8
Number of TOS metrics: 0
TOS 0 Metrics: 20
```

```
Link connected to: a Transit Network
(Link ID) Designated Router address: 164.1.24.10
(Link Data) Router Interface address: 164.1.24.8
Number of TOS metrics: 0
TOS 0 Metrics: 10
```

Here we have two transit networks connected to SW3 and SW4. When looking at the database, notice the link type. In our case, both links are transit, meaning this router is not along on those and see some neighbors. Now it's time to check the relevant entries in SW4's database:

Note

The router-ID is usually the loopback IP address. If you don't know the router ID you may find it using the command `show ip ospf neighbor` or peek it from the database output.

```
Rack1SW2# show ip ospf database router adv-router 150.1.10.10
```

```
OSPF Router with ID (150.1.8.8) (Process ID 1)
```

```
Router Link States (Area 38)
```

```
LS age: 1480
Options: (No TOS-capability, DC)
LS Type: Router Links
Link State ID: 150.1.10.10
Advertising Router: 150.1.10.10
LS Seq Number: 8000000D
Checksum: 0x55C4
Length: 72
Number of Links: 4
```

```
Link connected to: a Transit Network
```

```
(Link ID) Designated Router address: 164.1.43.10
(Link Data) Router Interface address: 164.1.43.10
Number of TOS metrics: 0
TOS 0 Metrics: 10
```

```
Link connected to: a Transit Network
```

```
(Link ID) Designated Router address: 164.1.24.10
(Link Data) Router Interface address: 164.1.24.10
Number of TOS metrics: 0
TOS 0 Metrics: 10
```

```
<snip>
```

At networks appear to be in order – both are marked as transit an OSPF SPF should account for these links as valid entries in the topology. Let's check the database of SW3 once again:

```
Rack1SW2# show ip ospf database router adv-router 150.1.9.9
```

```
OSPF Router with ID (150.1.8.8) (Process ID 1)
```

```
Router Link States (Area 38)
```

```
LS age: 1323
Options: (No TOS-capability, DC)
LS Type: Router Links
Link State ID: 150.1.9.9
Advertising Router: 150.1.9.9
LS Seq Number: 80000012
Checksum: 0xD5B
Length: 96
Number of Links: 6
```

```
Link connected to: a Transit Network
```

```
(Link ID) Designated Router address: 164.1.43.10
(Link Data) Router Interface address: 164.1.43.9
Number of TOS metrics: 0
TOS 0 Metrics: 10
```

```
Link connected to: another Router (point-to-point)
```

```
(Link ID) Neighboring Router ID: 150.1.8.8
(Link Data) Router Interface address: 164.1.32.9
Number of TOS metrics: 0
TOS 0 Metrics: 20

Link connected to: a Stub Network
(Link ID) Network/subnet number: 164.1.32.0
(Link Data) Network Mask: 255.255.255.0
Number of TOS metrics: 0
TOS 0 Metrics: 20
```

<snip>

The above shown are the links connecting to SW2 and SW4. Notice the link to SW4 looks normal – as a regular transit network. However, the link to SW2 is marked as “point-to-point” plus it generates a “stub network” entry. Even though the adjacency is UP, the link types at every end seem to mismatch, because SW2 treats the same link as “transit” subnet. The reason why is quickly deduced from the “point-to-point” keyword found in the first entry – SW3 must be configured for link type point-to-point on this interface! This is what dangerous about OSPF adjacencies – they come up as soon as timer values match. However, if the topological views of the link aren’t the same, the OSPF will never be able to have a complete graph of the network connection.

Conclusion: Don’t trust the OSPF adjacencies! If you see some path being ignore while adjacencies are up, there might be something wrong with the topology!

Fix the Issue

SW2:

```
interface Port-Channel 23
 ip ospf network point-to-point
```

Verify

Check that SW2 now has two equal-cost paths to reach VLAN9. One path via SW3 and another via SW4:

```
Rack1SW2#show ip route 164.1.9.9
```

```
Routing entry for 164.1.9.0/24
```

```
  Known via "ospf 1", distance 110, metric 21, type intra area
```

```
  Redistributing via eigrp 100
```

```
  Advertised by eigrp 100 metric 10000 1000 1 255 1500 route-map OSPF-
```

```
>EIGRP
```

```
  Last update from 164.1.24.10 on FastEthernet0/21, 00:03:13 ago
```

```
  Routing Descriptor Blocks:
```

```
    164.1.32.9, from 150.1.9.9, 00:03:13 ago, via Port-channel23
```

```
      Route metric is 21, traffic share count is 1
```

```
  * 164.1.24.10, from 150.1.9.9, 00:03:13 ago, via FastEthernet0/21
```

```
      Route metric is 21, traffic share count is 1
```