Network Access Layer Issues

The output from the **show interfaces** command is useful for detecting common media issues. One of the most important parts of this output is the display of the line and data link protocol status, as shown in the example.

The first parameter (FastEthernet0/18 is up) refers to the hardware layer and indicates whether the interface is receiving a carrier detect signal. The second parameter (line protocol is up) refers to the data link layer and indicates whether the data link layer protocol keepalives are being received.

Based on the output of the **show interfaces** command, possible problems can be fixed as follows:

- If the interface is up and the line protocol is down, a problem exists. There could (1)be an encapsulation type mismatch, the interface on (2) the other end could be error-disabled, or (3) there could be a hardware problem.
- If the line protocol and the interface are both down, a (1) cable is not attached, or some other interface problem exists. For example, in a back-to-back connection, (2) the other end of the connection may be administratively down.
- If the interface is administratively down, it has been manually disabled (the **shutdown** command has been issued) in the active configuration.

```
S1# show interfaces fastEthernet 0/18
FastEthernet0/18 is up, line protocol is up (connected)
Hardware is Fast Ethernet, address is 0025.83e6.9092 (bia 0025.83e6.9092)MTU 1500 bytes, BW
100000 Kbit/sec, DLY 100 usec,
```

Configure Switch Ports

Network Access Layer Issues (Cont.)

The **show interfaces** command output displays counters and statistics for the FastEthernet0/18 interface, as shown here:

```
S1# show interfaces fastEthernet 0/18
FastEthernet0/18 is up, line protocol is up (connected)
  Hardware is Fast Ethernet, address is 0025.83e6.9092 (bia 0025.83e6.9092)
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, media type is 10/100BaseTX
  input flow-control is off, output flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     2295197 packets input, 305539992 bytes, 0 no buffer
     Received 1925500 broadcasts (74 multicasts)
     0 runts, 0 giants, 0 throttles
     3 input errors, 3 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 74 multicast, 0 pause input
     0 input packets with dribble condition detected
     3594664 packets output, 436549843 bytes, 0 underruns
     8 output errors, 1790 collisions, 10 interface resets
     0 unknown protocol drops
     0 babbles, 235 late collision, 0 deferred
```

Network Access Layer Issues (Cont.)

Some media errors are not severe enough to cause the circuit to fail but do cause network performance issues. The table explains some of these common errors which can be detected using the **show interfaces** command.

Error Type	Description
Input Errors	Total number of errors. It includes runts, giants, no buffer, CRC, frame, overrun, and ignored counts.
Runts	Packets that are discarded because they are smaller than the minimum packet size for the medium. For instance, any Ethernet packet that is less than 64 bytes is considered a runt.
Giants	Packets that are discarded because they exceed the maximum packet size for the medium. For example, any Ethernet packet that is greater than 1,518 bytes is considered a giant.
CRC	CRC errors are generated when the calculated checksum is not the same as the checksum received.
Output Errors	Sum of all errors that prevented the final transmission of datagrams out of the interface that is being examined.
Collisions	Number of messages retransmitted because of an Ethernet collision.
Late Collisions	A collision that occurs after 512 bits of the frame have been transmitted

Interface Input and Output Errors

"Input errors" is the sum of all errors in datagrams that were received on the interface being examined. This includes runts, giants, CRC, no buffer, frame, overrun, and ignored counts. The reported input errors from the **show interfaces** command include the following:

- **Runt Frames** - Ethernet frames that are shorter than the 64-byte minimum allowed length are called runts. Malfunctioning NICs are the usual cause of excessive runt frames, but they can also be caused by collisions.
- **Giants** - Ethernet frames that are larger than the maximum allowed size are called giants.
- **CRC errors** - On Ethernet and serial interfaces, CRC errors usually indicate a media or cable error. Common causes include electrical interference, loose or damaged connections, or incorrect cabling. If you see many CRC errors, there is too much noise on the link and you should inspect the cable. You should also search for and eliminate noise sources.

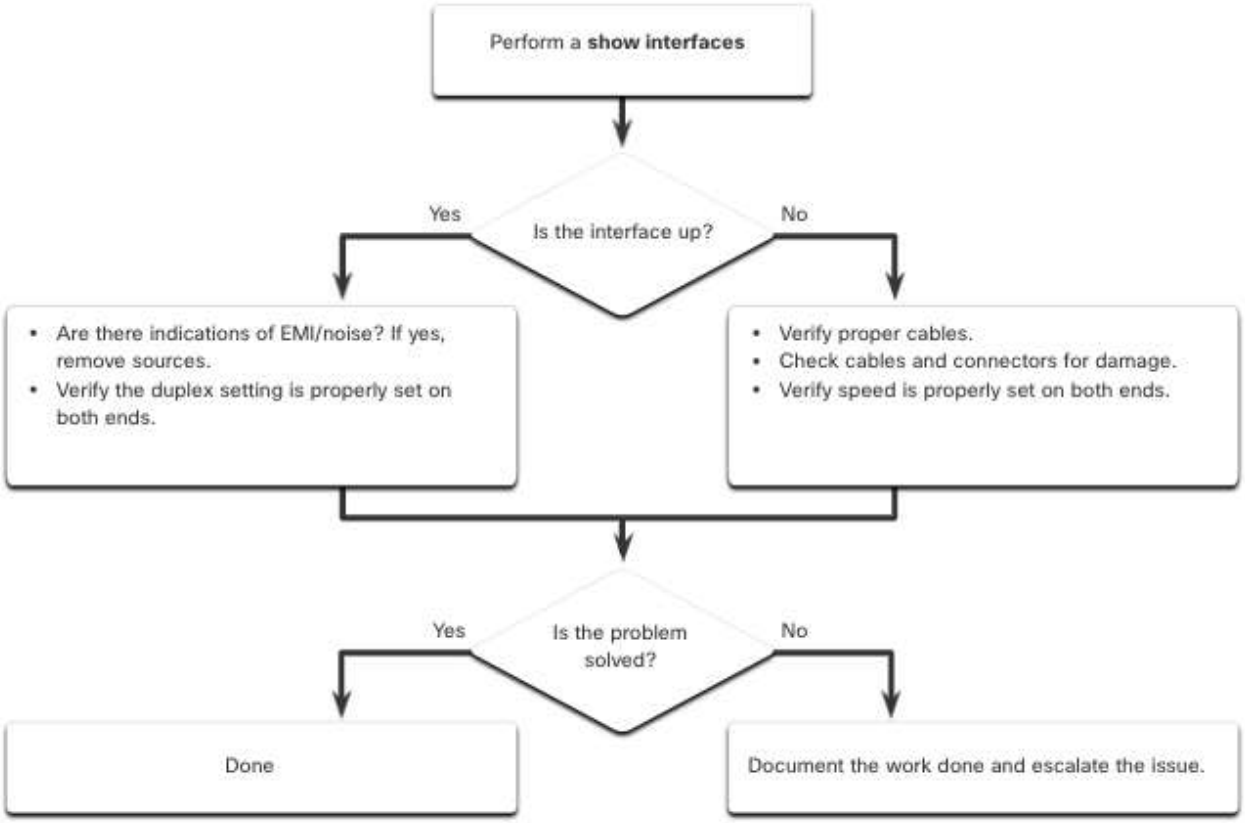
Interface Input and Output Errors (Cont.)

"Output errors" is the sum of all errors that prevented the final transmission of datagrams out the interface that is being examined. The reported output errors from the **show interfaces** command include the following:

- **Collisions** - Collisions in half-duplex operations are normal. However, you should never see collisions on an interface configured for full-duplex communication.
- **Late collisions** - A late collision refers to a collision that occurs after 512 bits of the frame have been transmitted. Excessive cable lengths are the most common cause of late collisions. Another common cause is duplex misconfiguration.

Troubleshooting Network Access Layer Issues

To troubleshoot scenarios involving no connection, or a bad connection, between a switch and another device, follow the general process shown in the figure.



1.3 Secure Remote Access

Secure Remote Access

Telnet Operation

Telnet uses TCP port 23. It is an older protocol that uses unsecure plaintext transmission of both the login authentication (username and password) and the data transmitted between the communicating devices.

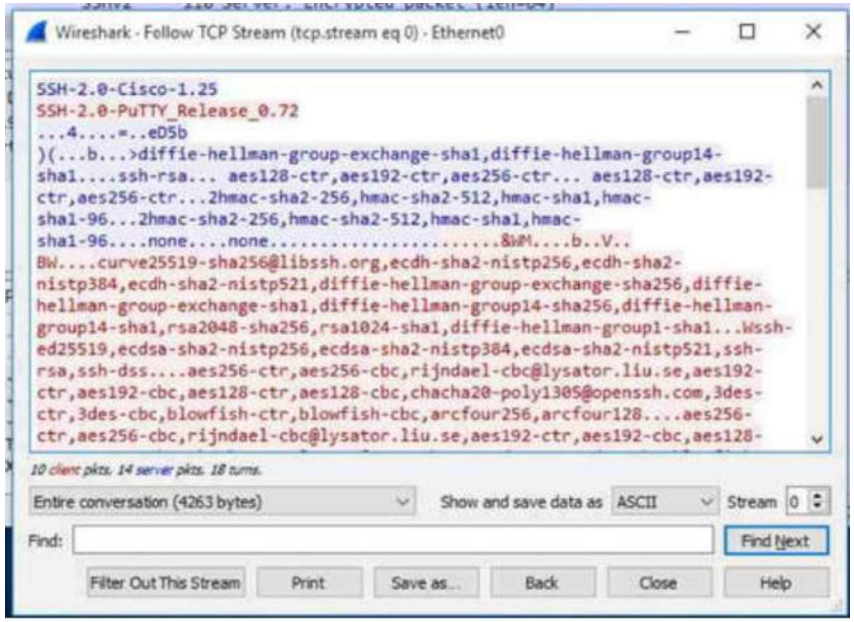
A threat actor can monitor packets using Wireshark. For example, in the figure the threat actor captured the username **admin** and password **ccna** from a Telnet session.

Secure Remote Access

SSH Operation

Secure Shell (SSH) is a secure protocol that uses TCP port 22. It provides a secure (encrypted) management connection to a remote device. SSH should replace Telnet for management connections. SSH provides security for remote connections by providing strong encryption when a device is authenticated (username and password) and also for the transmitted data between the communicating devices.

The figure shows a Wireshark capture of an SSH session. The threat actor can track the session using the IP address of the administrator device. However, unlike Telnet, with SSH the username and password are encrypted.



Secure Remote Access

Verify the Switch Supports SSH

To enable SSH on a Catalyst 2960 switch, the switch must be using a version of the IOS software including cryptographic (encrypted) features and capabilities. Use the **show version** command on the switch to see which IOS the switch is currently running. An IOS filename that includes the combination “k9” supports cryptographic (encrypted) features and capabilities.

The example shows the output of the **show version** command.

```
S1# show version
Cisco IOS Software, C2960 Software (C2960-LANBASEK9-M), Version 15.0(2)SE7, RELEASE SOFTWARE
(fc1)
```

Secure Remote Access

Configure SSH

Before configuring SSH, the switch must be minimally configured with a unique hostname and the correct network connectivity settings.

Step 1: Verify SSH support - Use the **show ip ssh** command to verify that the switch supports SSH. If the switch is not running an IOS that supports cryptographic features, this command is unrecognized.

Step 2: Configure the IP domain - Configure the IP domain name of the network using the **ip domain-name** domain-name global configuration mode command.

Step 3: Generate RSA key pairs - Generating an RSA key pair automatically enables SSH. Use the **crypto key generate rsa** global configuration mode command to enable the SSH server on the switch and generate an RSA key pair.

Note: To delete the RSA key pair, use the **crypto key zeroize rsa** global configuration mode command. After the RSA key pair is deleted, the SSH server is automatically disabled.

Step 4: Configure user authentication - The SSH server can authenticate users locally or using an authentication server. To use the local authentication method, create a username and password pair using the **username** username **secret** password global configuration mode command.

Step 5: Configure the vty lines - Enable the SSH protocol on the vty lines by using the transport input ssh line configuration mode command. Use the **line vty** global configuration mode command and then the login local line configuration mode command to require local authentication for SSH connections from the local username database.

Step 6: Enable SSH version 2 - By default, SSH supports both versions 1 and 2. When supporting both versions, this is shown in the show ip ssh output as supporting version 2. Enable SSH version using the ip ssh version 2 global configuration command.

Secure Remote Access

Verify SSH is Operational

On a PC, an SSH client such as PuTTY, is used to connect to an SSH server. For example, assume the following is configured:

- SSH is enabled on switch S1
- Interface VLAN 99 (SVI) with IPv4 address 172.17.99.11 on switch S1
- PC1 with IPv4 address 172.17.99.21

Using a terminal emulator, initiate an SSH connection to the SVI VLAN IPv4 address of S1 from PC1.

When connected, the user is prompted for a username and password as shown in the example. Using the configuration from the previous example, the username **admin** and password **ccna** are entered. After entering the correct combination, the user is connected via SSH to the command line interface (CLI) on the Catalyst 2960 switch.

```
Login as: admin
Using keyboard-interactive
Authentication.
Password:
S1> enable
Password:
S1#
```

Secure Remote Access

Verify SSH is Operational (Cont.)

To display the version and configuration data for SSH on the device that you configured as an SSH server, use the **show ip ssh** command. In the example, SSH version 2 is enabled.

```
S1# show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
To check the SSH connections to the device, use the show ssh command as shown.
S1# show ssh
%No SSHv1 server connections running.
Connection Version Mode Encryption  Hmac         State                 Username
0          2.0     IN   aes256-cbc  hmac-sha1    Session started       admin
0          2.0     OUT  aes256-cbc  hmac-sha1    Session started       admin
S1#
```

Secure Remote Access

Packet Tracer – Configure SSH

In this Packet Tracer, you will do the following:

- Secure passwords
- Encrypt communications
- Verify SSH implementation

1.4 Basic Router Configuration

Configure Basic Router Settings

Cisco routers and Cisco switches have many similarities. They support a similar modal operating system, similar command structures, and many of the same commands. In addition, both devices have similar initial configuration steps. For example, the following configuration tasks should always be performed. Name the device to distinguish it from other routers and configure passwords, as shown in the example.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# hostname R1
R1(config)# enable secret class
R1(config)# line console 0
R1(config-line)# password cisco
R1(config-line)# login
R1(config-line)# exit
R1(config)# line vty 0 4
R1(config-line)# password cisco
R1(config-line)# login
R1(config-line)# exit
R1(config)# service password-encryption
R1(config)#
```

Configure Basic Router Settings (Cont.)

Configure a banner to provide legal notification of unauthorized access, as shown in the example.

```
R1(config)# banner motd $ Authorized Access Only! $
R1(config)#
```

Save the changes on a router, as shown in the example.

```
R1# copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```

