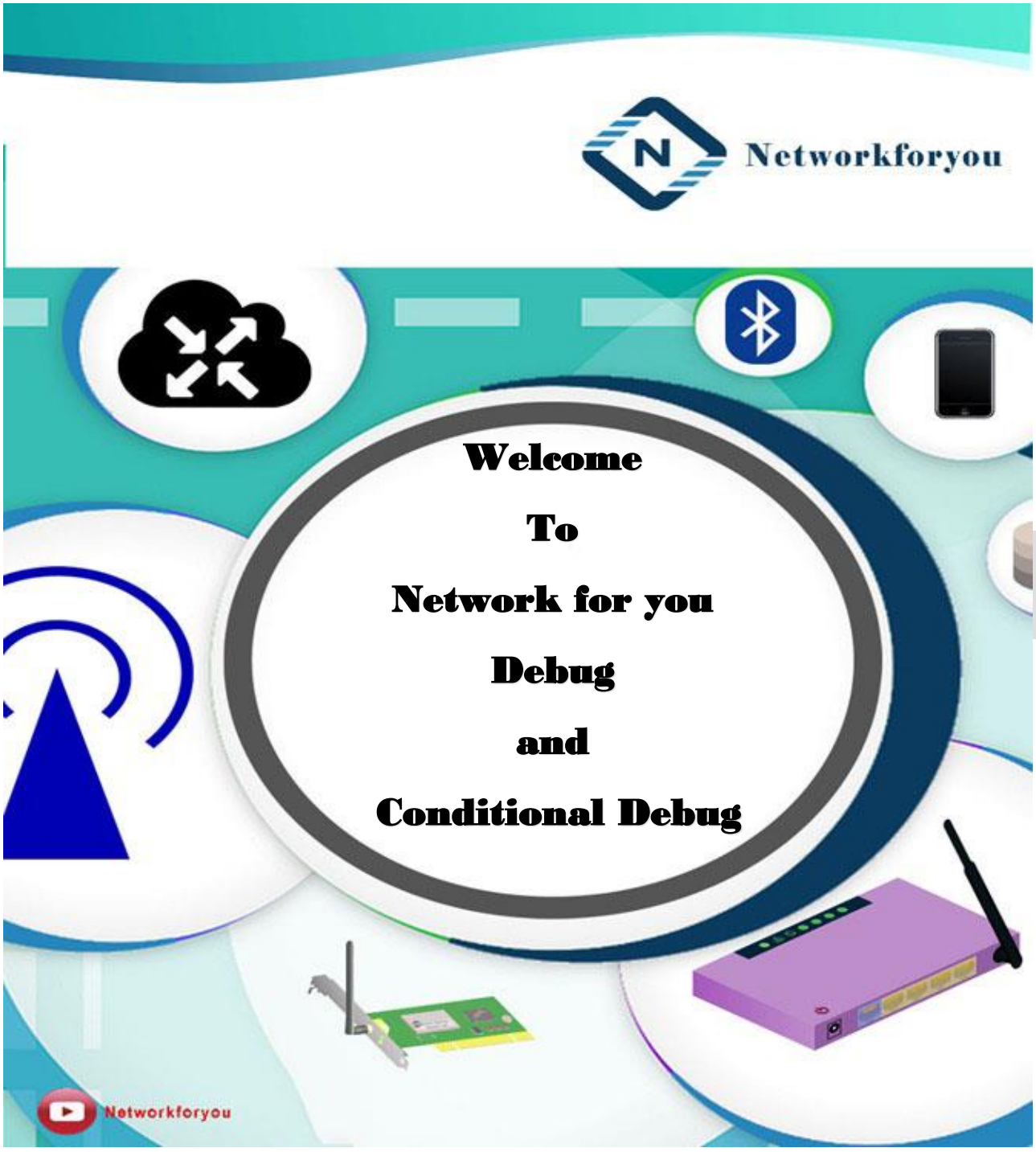

Networkforyou
Welcome
To
Network for you
Debug
and
Conditional Debug
Networkforyou

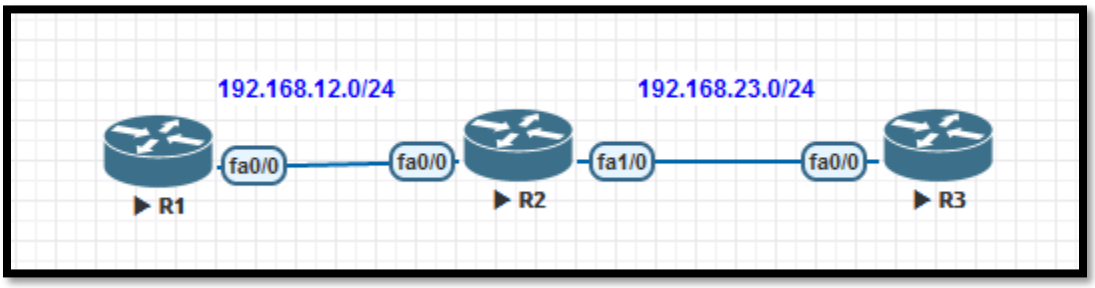
Debug and Conditional Debug:

- Debugs are useful tools when troubleshooting specific issues.
- Debug commands help to show the real time information on device.
- Debug facility used to keep track in case of events or protocol errors.
- Debug facility used to keep track the internal messages taking place.
- Conditional debug is very useful to filter out some of the debug information.
- Conditionally debugging generates debugging messages matching given condition.
- Such as conditionally debugging messages for one interface or sub-interface.
- Turn on debugging for all interfaces that meet specified conditions.
- Debugs can produce a huge amount of information, which overload the router.
- Use conditional debug to limit information based on specific interface, or protocol.
- Using no debug all or undebug all does not remove the condition debugging.

Debugging and Condition Debugging	
R1# debug ip icmp	R1# debug condition ?
R1# debug ip packet	R1#debug condition interface f0/0
R1# debug interface	R1#undebug condition interface f0/0
R1# debug eigrp packets hello	R1#show debugging
R1#show debug condition	R1#no debug all

Lab time:

let do a Basic OSPF lab and use Debug and Conditional Debug:



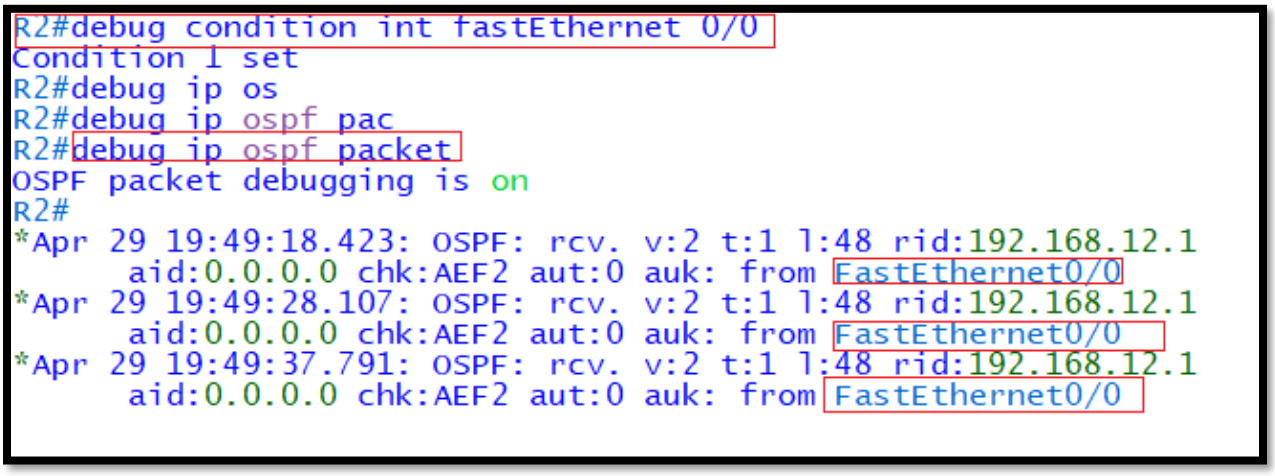
R1 Configuration	R2 Configuration	R3 Configuration
en config t hostname R1 int f0/0 ip add 192.168.12.1 255.255.255.0 no sh router ospf 1 int f0/0 ip ospf 1 area 0	en config t hostname R2 int f0/0 ip add 192.168.12.2 255.255.255.0 no sh int f1/0 ip add 192.168.23.1 255.255.255.0 no sh router ospf 1 int f0/0 ip ospf 1 area 0 int f1/0 ip ospf 1 area 0	en config t hostname R3 int f0/0 ip add 192.168.23.2 255.255.255.0 no sh router ospf 1 int f0/0 ip ospf 1 area 0

When Enable OSPF debugging on R2. Showing OSPF debug information from both interfaces

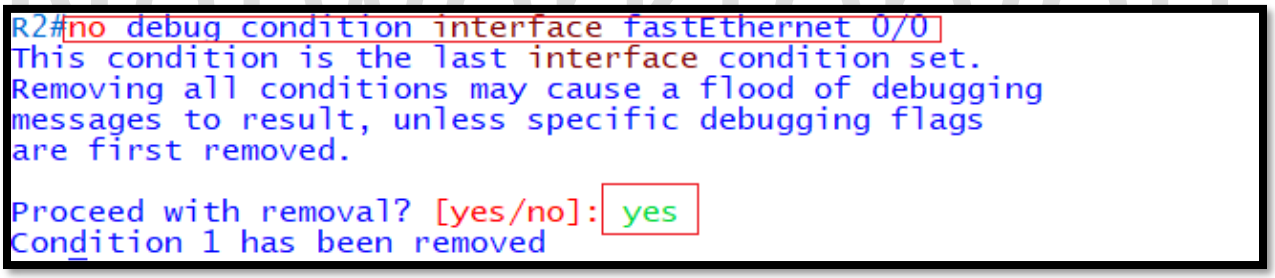
```
R2#debug ip ospf packet
OSPF packet debugging is on
R2#
*Apr 29 19:46:00.055: OSPF: rcv. v:2 t:1 l:48 rid:192.168.23.2
      aid:0.0.0.0 chk:8DF1 aut:0 auk: from FastEthernet1/0
*Apr 29 19:46:05.815: OSPF: rcv. v:2 t:1 l:48 rid:192.168.12.1
      aid:0.0.0.0 chk:AEF2 aut:0 auk: from FastEthernet0/0
```

If only, want to see the debug information from one interface then use a debug condition.

Using debug condition only can see OSPF debug information from the FastEthernet 0/0 interface.



To remove condition debug put no in the begging of the command and type yes.



Ping Command:

- Ping stand for Packet Internet groper.
- The ping command is a very common method for troubleshooting the accessibility of devices.
- It uses two Internet Control Message Protocol (ICMP) query messages, ICMP echo requests and ICMP echo replies, to determine whether a remote host is active.
- The ping command also measures the amount of time it takes to receive the echo reply.
- The ping command first sends an echo request packet to an address, and then it waits for a reply.
- The ping is successful only if the echo request gets to the destination, and the destination is able to get an echo reply back to the source of the ping.

Character	Meaning
!	Successful receipt of a reply.
.	Device timed out while waiting for a reply.
U	A destination unreachable error protocol data unit (PDU) was received.
Q	Source quench (destination too busy).
M	Could not fragment.
?	Unknown packet type.
&	Packet lifetime exceeded.

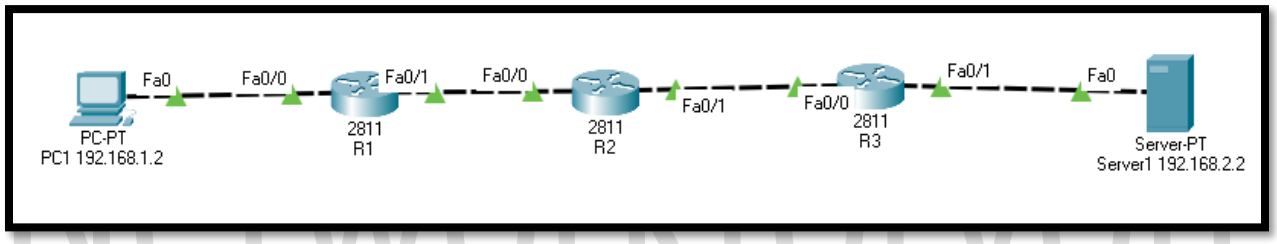
Extended Ping Command:

- When a normal ping command is sent from a router, the source address of the ping is the IP address of the interface that the packet uses to exit the router.
- If an extended ping command is used, the source IP address can be changed to any IP address on the router.

```
R2#ping
Protocol [ip]:
Target IP address: 10.0.0.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: yes
Source address or interface: 192.168.5.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
Packet sent with a source address of 192.168.5.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
```

Traceroute Command:

- Traceroute, like the ping command can be used to isolate problems in our network. The ping command is a bit limited sometimes.
- The traceroute command can be used to discover the routes packets take to a remote destination, as well as where routing breaks down.
- The device executing the traceroute command sends out a sequence of User Datagram Protocol (UDP) datagrams, each with incrementing Time-To-Live (TTL) values, to an invalid port address at the remote host.
- The purpose behind the traceroute command is to record the source of each ICMP "time exceeded" message to provide a trace of the path the packet took to reach the destination.



- When we send a ping from PC1 (192.168.1.2) to Server1 (192.168.2.2) and this ping doesn't work, what does it mean? We'll know something is not working but we don't know whether the problem is in between PC1-R1, R1-R2, R2-R3 or R3-Server1.
- If you know the IP addresses of all routers in the path then you could ping all of these routers one by one. What if you have no idea how many routers are in between? Or if you don't know their IP addresses?

The traceroute command will help us with that. Let see for above example:

```
C:\>tracert 192.168.2.2

Tracing route to 192.168.2.2 over a maximum of 30 hops:

  1    0 ms      0 ms      0 ms      192.168.1.1
  2    0 ms      0 ms      0 ms      192.168.12.2
  3    0 ms      0 ms      0 ms      192.168.23.2
  4    *         0 ms      6 ms      192.168.2.2

Trace complete.
```

Above used the tracert (traceroute) command on a Windows computer to trace the path from my computer to Server1. You can see all the IP addresses and hostnames of the routers in between my computer and the server that responds to 192.168.2.2. The response times that you see is the round trip time from my computer to the router. For each router, traceroute sends three probes.

So, how does traceroute work?

Traceroute uses the TTL (Time to Live) field in the IP packet header. Normally, TTL is used to prevent packets from being forwarded forever when there is a routing loop. Whenever an IP packet is forwarded by a router, the TTL is decreased by one. When the TTL is zero, the IP packet will be discarded.

```
R1#Traceroute 192.168.2.1
Type escape sequence to abort.
Tracing the route to 192.168.2.1

  1   192.168.12.2    8 msec    0 msec    0 msec
  2   192.168.23.2    0 msec    0 msec    0 msec
```

```
C:\>tracert 192.168.2.2

Tracing route to 192.168.2.2 over a maximum of 30 hops:

  1    0 ms      0 ms      0 ms      192.168.1.1
  2    0 ms      0 ms      0 ms      192.168.12.2
  3    0 ms      0 ms      0 ms      192.168.23.2
  4    *         *         *         Request timed out.
  5    *         *         *         Request timed out.
  6    *         *         *         Request timed out.
  7    *         *         *         Request timed out.
  8    *         *         *         Request timed out.
  9    *         *         *         Request timed out.
  10    *         *         *         Request timed out.
  11    *         *         *         Request timed out.
  12    *         *         *         Request timed out.
  13    *         *         *         Request timed out.
  14    *         *         *         Request timed out.
  15    *         *         *         Request timed out.
```


If you see above picture with this you can say till R3 is ok after that some issues so this way we can trace issues.

Extended Traceroute Command:

- The extended traceroute command is a variation of the traceroute command.
- An extended traceroute command can be used to see what path packets are taking to get to a destination, and the command can be used to check routing at the same time.
- This is helpful for troubleshooting routing loops, or for determining where packets are getting lost.