

Fall 2024 CS120

Routing Protocol Simulation

Yihui Yan
yanyh@shanghaitech.edu.cn

Outline

1. Simulator: IMUNES
2. RIP (Routing Information Protocol)
3. OSPF (Open Shortest Path First)

1. Simulator: IMUNES*

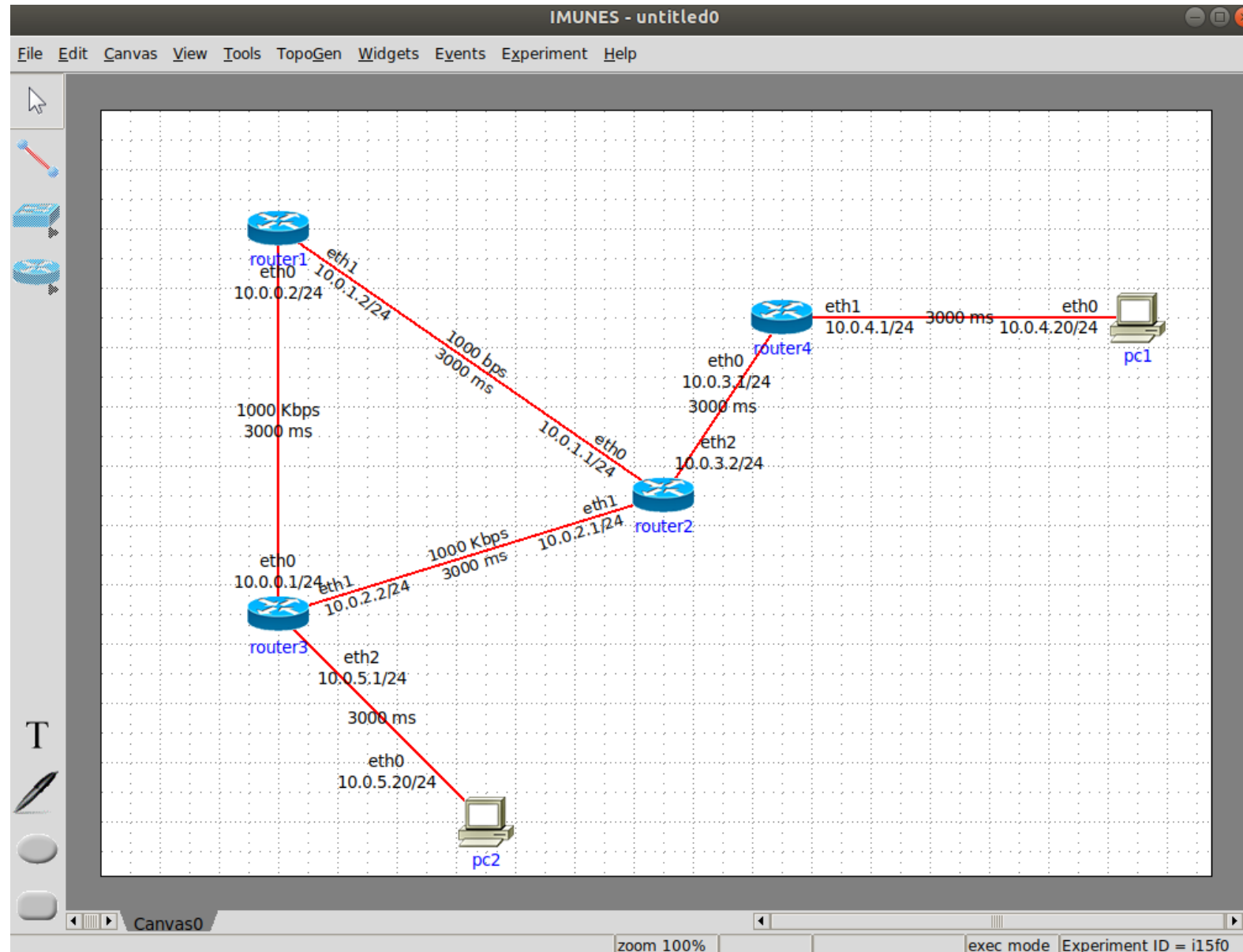
1.1. Introduction

1.2. Installation

1.3. Basic Operation

*IMUNES: <https://github.com/imunes/imunes>, <http://imunes.net/dl/guide/>

1. Simulator: IMUNES



1.1. Introduction

Advantages:

1. Open-source network simulator
2. User guide manual and imunes-examples repositories
3. GUI
4. Inspect network traffic

1.2. Installation

1.2.1. install software dependencies:

Ubuntu 18.04 LTS (Mint 19, 19.1)

```
# apt install openvswitch-switch docker.io xterm wireshark \
make imagemagick tk tcllib util-linux
```

1.2.2. install IMUNES

```
$ sudo git clone https://github.com/imunes/imunes.git
```

```
$ cd imunes
```

```
$ sudo make install
```

```
$ sudo imunes -p //create a template filesystem for the topologies to work
```

1.2.3. Run IMUNES GUI

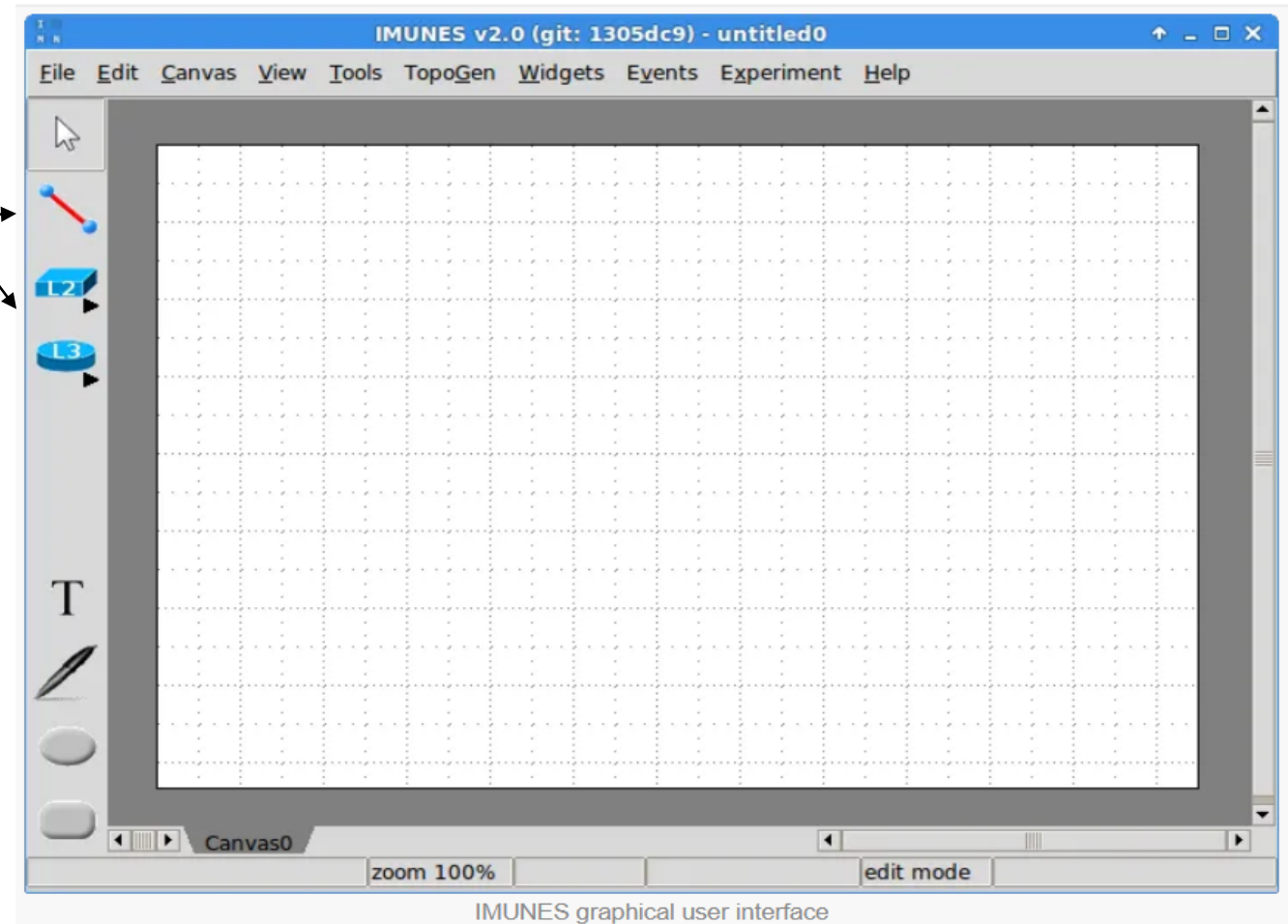
```
$ sudo imunes
```

1.3. Basic Operation

1.3.1. Building a simple network

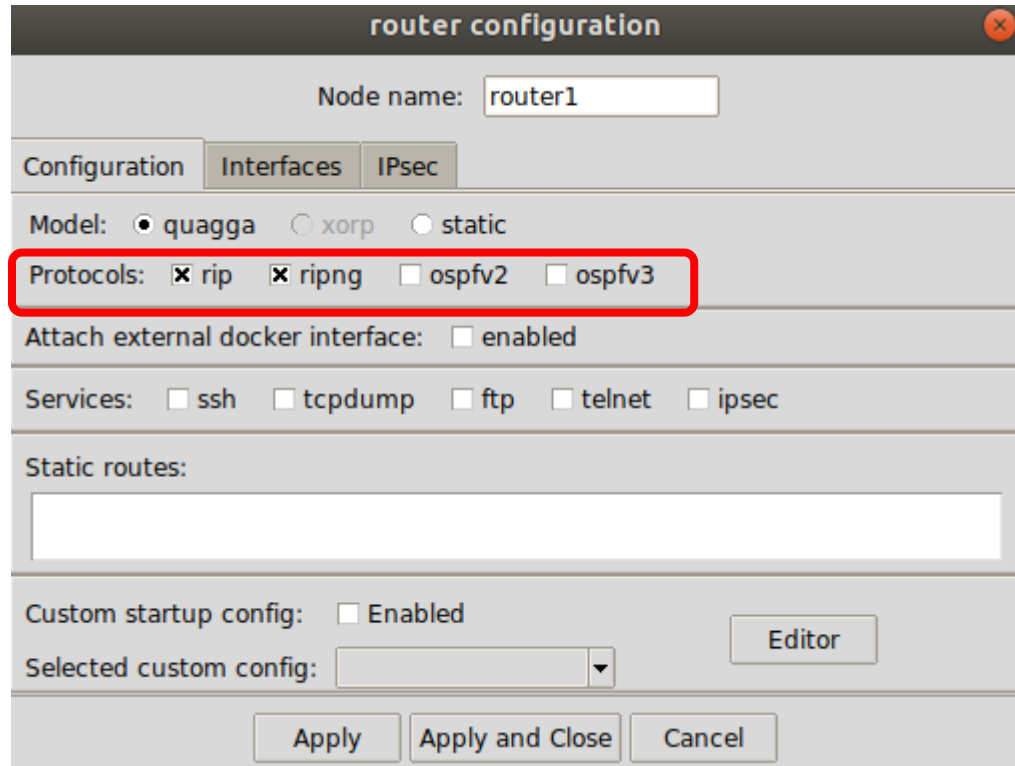
Add Nodes (router, PC, switch...)

Connect Nodes



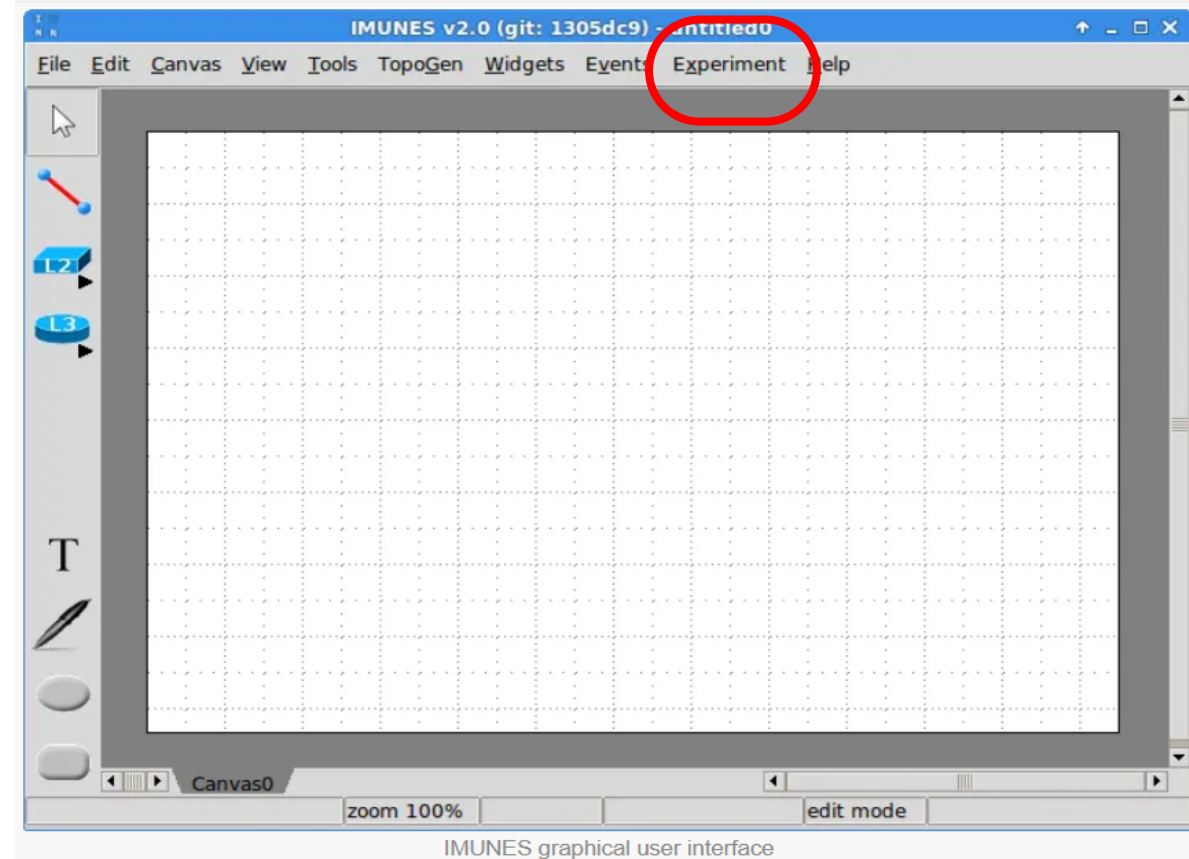
1.3. Basic Operation

1.3.2. Configuring a simple network:
Double click on the network element

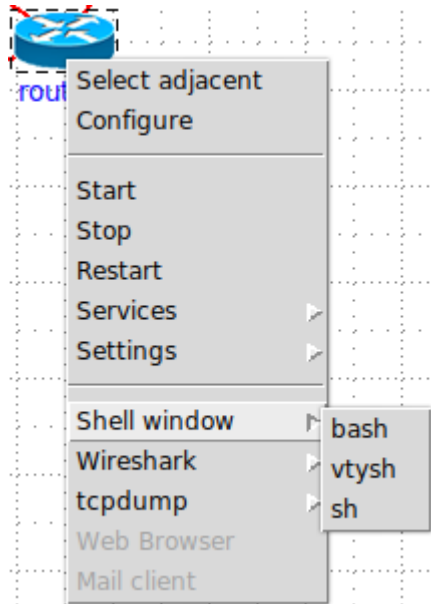


The image shows a 'router configuration' dialog box. At the top, there's a 'Node name' field with 'router1' entered. Below this are three tabs: 'Configuration', 'Interfaces', and 'IPsec'. The 'Configuration' tab is active. It contains several sections: 'Model' with radio buttons for 'quagga' (selected), 'xorp', and 'static'; 'Protocols' with checkboxes for 'rip' (checked), 'ripng' (checked), 'ospfv2', and 'ospfv3' (this entire section is highlighted with a red rectangle); 'Attach external docker interface' with an 'enabled' checkbox; 'Services' with checkboxes for 'ssh', 'tcpdump', 'ftp', 'telnet', and 'ipsec'; 'Static routes' with an empty text area; 'Custom startup config' with an 'Enabled' checkbox; and 'Selected custom config' with a dropdown menu. At the bottom are 'Apply', 'Apply and Close', and 'Cancel' buttons, along with an 'Editor' button.

1.3.3. Start simulation
Experiment ->Execute



1.4. Shell Windows



bash: Unix shell. An implementation of sh.
Configure and examine the system.

vtysh: Shell for Quagga routing engine.
Configure and examine the router.

sh: Usually, /bin/sh is used to point to /bin/dash on most GNU/Linux systems. The standard command interpreter for the system.

```
root@router1:/# ls -al /bin/sh
lrwxrwxrwx 1 root root 4 Jan 24 2017 /bin/sh -> dash
root@router1:/#
```

1.4. Shell Windows: bash

1. **ifconfig**: check or set network interface configurations
 2. **ping**: check connectivity and show the delay
 3. **traceroute**: trace the routing path and show the delay
 4. **netstat**: examine network information (interface/routing table)
- * --help will give you more information

```
root@router1:/# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.5.1 netmask 255.255.255.0 broadcast 10.0.5.255
    inet6 fe80::4000:aaff:fe00:e prefixlen 64 scopeid 0x20<link>
    inet6 fc00:1f::1 prefixlen 64 scopeid 0x0<global>
    ether 42:00:aa:00:00:0e txqueuelen 1000 (Ethernet)
    RX packets 81 bytes 7657 (7.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 358 bytes 30984 (30.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
root@pc9:/# ping 10.0.6.1
PING 10.0.6.1 (10.0.6.1) 56(84) bytes of data.
64 bytes from 10.0.6.1: icmp_seq=1 ttl=64 time=0.405 ms
64 bytes from 10.0.6.1: icmp_seq=2 ttl=64 time=0.063 ms
64 bytes from 10.0.6.1: icmp_seq=3 ttl=64 time=0.065 ms
64 bytes from 10.0.6.1: icmp_seq=4 ttl=64 time=0.063 ms
^C
--- 10.0.6.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3053ms
rtt min/avg/max/mdev = 0.063/0.149/0.405/0.147 ms
```

```
root@pc2:/# traceroute 10.0.4.20
traceroute to 10.0.4.20 (10.0.4.20), 30 hops max, 60 byte packets
 1  10.0.5.1 (10.0.5.1)  0.066 ms  0.013 ms  0.006 ms
 2  10.0.2.1 (10.0.2.1)  12.865 ms  13.605 ms  14.045 ms
 3  10.0.3.1 (10.0.3.1)  14.861 ms  16.387 ms  16.747 ms
 4  10.0.4.20 (10.0.4.20)  17.285 ms  18.177 ms  19.110 ms
```

```
root@pc2:/# netstat --interface
Kernel Interface table
Iface      MTU      RX-OK RX-ERR RX-DRP RX-OVR      TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0       1500      612      0      0 0          47      0      0      0 BMRU
lo         16384      36      0      0 0          36      0      0      0 LRU

root@pc2:/# netstat --route
Kernel IP routing table
Destination Gateway      Genmask      Flags   MSS Window  irtt Iface
default    10.0.5.1    0.0.0.0      UG        0 0        0 eth0
10.0.5.0    0.0.0.0    255.255.255.0 U         0 0        0 eth0
```

1.4. Shell Windows: vtysh*

- show ip route [rip | ospf | ...]

```
router2# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

O>* 10.0.0.0/24 [110/20] via 10.0.1.2, eth0, 00:02:38
   *                via 10.0.2.2, eth1, 00:02:38
O   10.0.1.0/24 [110/10] is directly connected, eth0, 6d00h39m
C>* 10.0.1.0/24 is directly connected, eth0
O   10.0.2.0/24 [110/10] is directly connected, eth1, 6d00h07m
C>* 10.0.2.0/24 is directly connected, eth1
O   10.0.3.0/24 [110/10] is directly connected, eth2, 6d00h39m
C>* 10.0.3.0/24 is directly connected, eth2
O>* 10.0.4.0/24 [110/20] via 10.0.3.1, eth2, 6d00h27m
O>* 10.0.5.0/24 [110/20] via 10.0.2.2, eth1, 00:02:38
C>* 127.0.0.0/8 is directly connected, lo
O>* 127.0.0.1/32 [110/0] is directly connected, lo, 01w0d01h
```

```
router2# show ip route ospf
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

O>* 10.0.0.0/24 [110/20] via 10.0.1.2, eth0, 00:03:05
   *                via 10.0.2.2, eth1, 00:03:05
O   10.0.1.0/24 [110/10] is directly connected, eth0, 6d00h39m
O   10.0.2.0/24 [110/10] is directly connected, eth1, 6d00h07m
O   10.0.3.0/24 [110/10] is directly connected, eth2, 6d00h40m
O>* 10.0.4.0/24 [110/20] via 10.0.3.1, eth2, 6d00h27m
O>* 10.0.5.0/24 [110/20] via 10.0.2.2, eth1, 00:03:05
O>* 127.0.0.1/32 [110/0] is directly connected, lo, 01w0d01h
```

1.4. Shell Windows: vtysh

- show ip ospf neighbor (see 3.1)
- show ip ospf database (see 3.1)
- show ip ospf interface

```
router2# show ip ospf interface eth0
eth0 is up
  ifindex 80, MTU 1500 bytes, BW 0 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.0.1.1/24, Broadcast 10.0.1.255, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 10.0.3.2, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 10.0.3.2, Interface Address 10.0.1.1
  Backup Designated Router (ID) 10.0.1.2, Interface Address 10.0.1.2
  Saved Network-LSA sequence number 0x80000006
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 7.208s
  Neighbor Count is 1, Adjacent neighbor count is 1
```

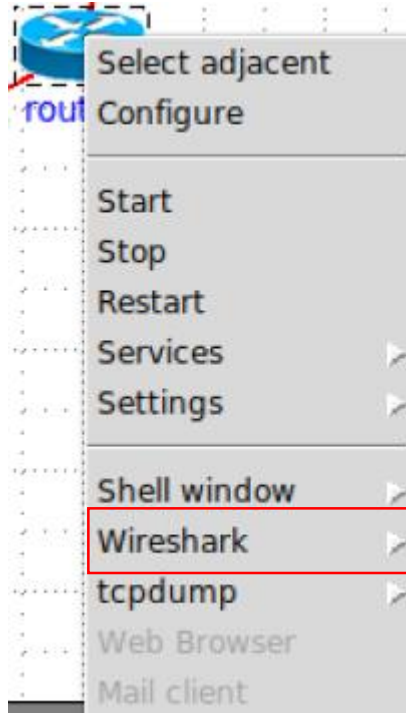
1.4. Shell Windows: vtysh

Configure the cost of OSPF (also see 3.3)

- config terminal
- interface INTERFACE_NAME
- bandwidth BW / ospf cost COST

```
router2# configure terminal
router2(config)# interface eth0
router2(config-if)# ospf cost 66
router2(config-if)# bandwidth 666
router2(config-if)# exit
router2(config)# exit
router2# show ip ospf interface
eth0 is up
  ifindex 80, MTU 1500 bytes, BW 666 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.0.1.1/24, Broadcast 10.0.1.255, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 10.0.3.2, Network Type BROADCAST, Cost: 66
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 10.0.3.2, Interface Address 10.0.1.1
  Backup Designated Router (ID) 10.0.1.2, Interface Address 10.0.1.2
  Saved Network-LSA sequence number 0x80000006
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 9.812s
  Neighbor Count is 1, Adjacent neighbor count is 1
```

1.5. Wireshark*

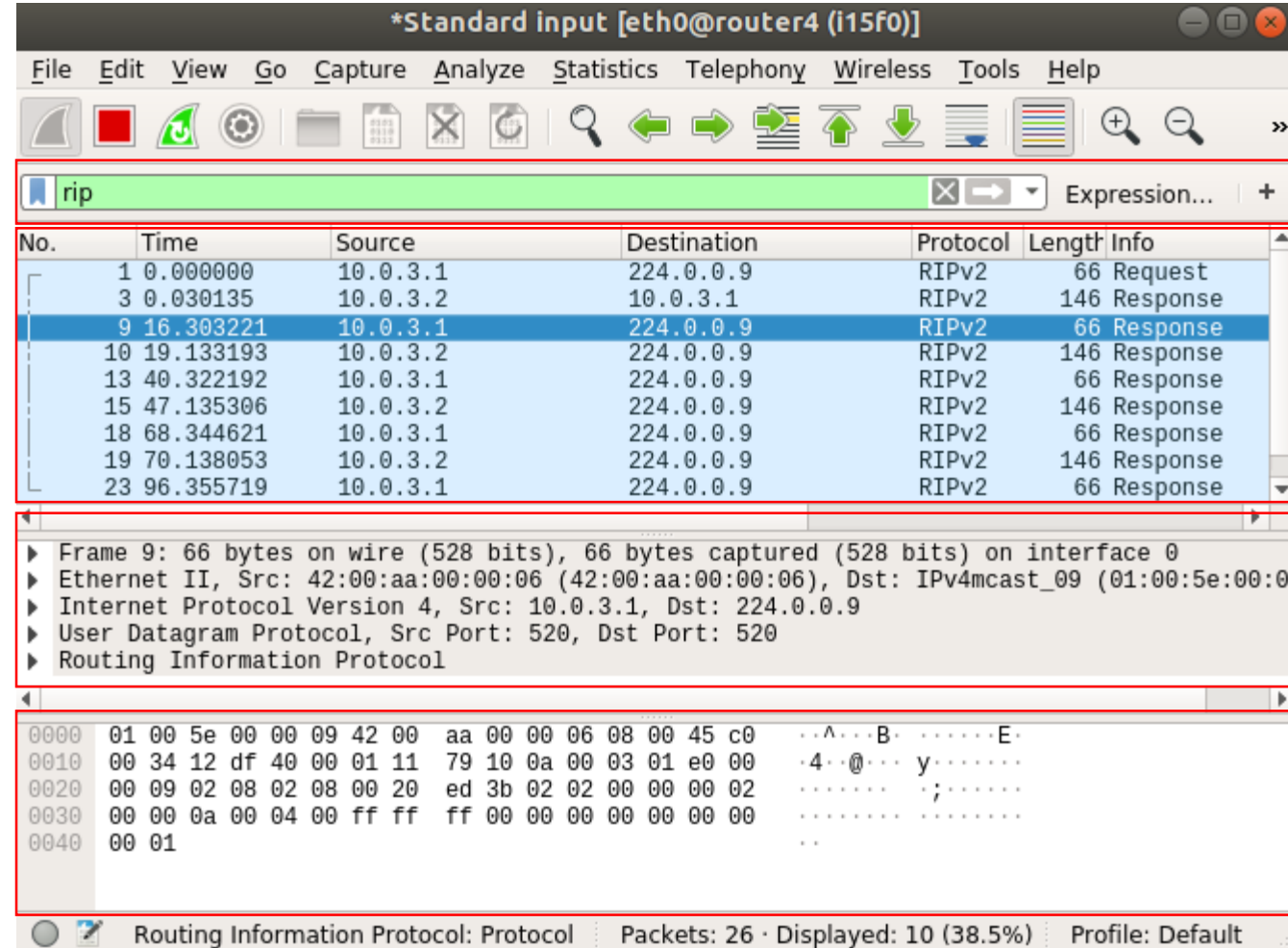


Filter: Protocol type /
(Src/Dst) IP address /
port ...

List of captured packet

Packet details

Packet in Hex



*Wireshark (user guide): https://www.wireshark.org/docs/wsug_html_chunked/

2. RIP* (Routing Information Protocol)

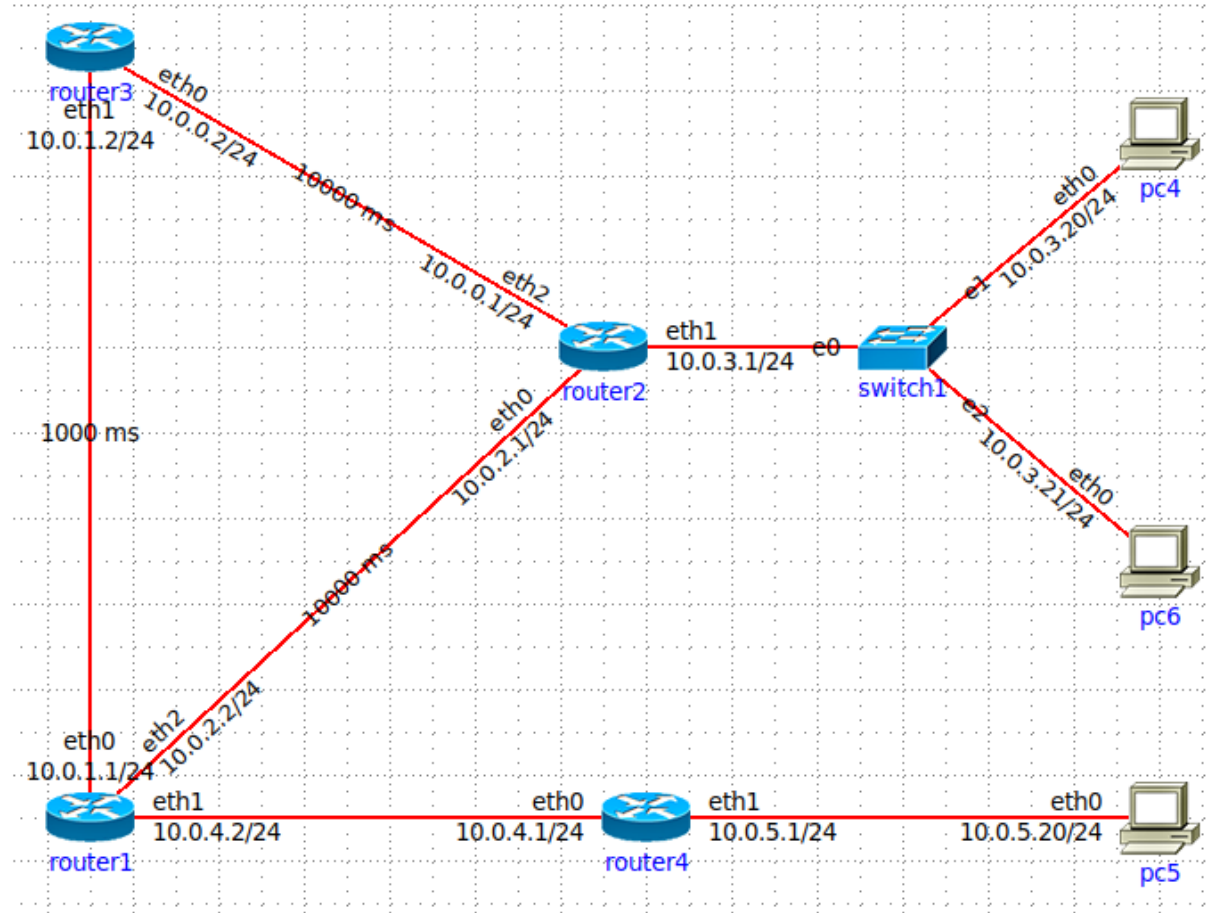
2.1. Stable Network

2.2. Encounter Network Node Failure

2.3. Count to Infinity Problem

2.4. Silent Failure

2. RIP (Routing Information Protocol)



*The RIP topology file is rip.imn, open it in the IMUNES.

2.1. Stable Network

RIP Routing Table

IMUNES: router4 (console) bash

Every 1.0s: vtysh -c "show ip rip" router4: Thu Oct 29 11:05:09 2020

Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
(n) - normal, (s) - static, (d) - default, (r) - redistribute,
(i) - interface

① Network	② Next Hop	③ Metric	From	④ Tag	Time
R(n) 10.0.0.0/24	10.0.4.2	3	10.0.4.2	0	02:53
R(n) 10.0.1.0/24	10.0.4.2	2	10.0.4.2	0	02:53
R(n) 10.0.2.0/24	10.0.4.2	2	10.0.4.2	0	02:53
R(n) 10.0.3.0/24	10.0.4.2	3	10.0.4.2	0	02:53
C(i) 10.0.4.0/24	0.0.0.0	1	self	0	
C(i) 10.0.5.0/24	0.0.0.0	1	self	0	

① Destination Network

② Next Hop

③ Convergent Costs

④ Time-to-live

RIP advertisement - periodic update

*Standard input [eth0@router4 (i15f0)]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

rip and ip.src==10.0.4.1

No.	Time	Source	Destination	Protocol	Length	Info
625	362663.274029	10.0.4.1	224.0.0.9 *	RIPv2	66	Response
628	362686.276114	10.0.4.1	224.0.0.9	RIPv2	66	Response
632	362715.277717	10.0.4.1	224.0.0.9	RIPv2	66	Response
635	362747.278324	10.0.4.1	224.0.0.9	RIPv2	66	Response
637	362775.280457	10.0.4.1	224.0.0.9	RIPv2	66	Response
641	362810.281305	10.0.4.1	224.0.0.9	RIPv2	66	Response
644	362845.283309	10.0.4.1	224.0.0.9	RIPv2	66	Response

Frame 644: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: 42:00:aa:00:00:08 (42:00:aa:00:00:08), Dst: IPv4mcast_09 (01:00:5e:00:00:09)
Internet Protocol Version 4, Src: 10.0.4.1, Dst: 224.0.0.9
User Datagram Protocol, Src Port: 520, Dst Port: 520
Routing Information Protocol
Command: Response (2)
Version: RIPv2 (2)
IP Address: 10.0.5.0, Metric: 1

IP Address (rip.ip), 4 bytes Packets: 645 · Displayed: 195 (30.2%) Profile: Default

2.1. Stable Network

RIP Routing Table

```
IMUNES: router4 (console) bash
Every 1.0s: vtysh -c "show ip rip"          router4: Thu Oct 29 11:05:09 2020
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
  (n) - normal, (s) - static, (d) - default, (r) - redistribute,
  (i) - interface
```

	Network	Next Hop	Metric	From	Tag	Time
R(n)	10.0.0.0/24	10.0.4.2	3	10.0.4.2	0	02:53
R(n)	10.0.1.0/24	10.0.4.2	2	10.0.4.2	0	02:53
R(n)	10.0.2.0/24	10.0.4.2	2	10.0.4.2	0	02:53
R(n)	10.0.3.0/24	10.0.4.2	3	10.0.4.2	0	02:53
C(i)	10.0.4.0/24	0.0.0.0	1	self	0	
C(i)	10.0.5.0/24	0.0.0.0	1	self	0	

Why does not Router4 advertise its whole RIP routing table?

RIP advertisement - periodic update

```
*Standard input [eth0@router4 (i15f0)]
```

No.	Time	Source	Destination	Protocol	Length	Info
625	362663.274029	10.0.4.1	224.0.0.9	RIPv2	66	Response
628	362686.276114	10.0.4.1	224.0.0.9	RIPv2	66	Response
632	362715.277717	10.0.4.1	224.0.0.9	RIPv2	66	Response
635	362747.278324	10.0.4.1	224.0.0.9	RIPv2	66	Response
637	362775.280457	10.0.4.1	224.0.0.9	RIPv2	66	Response
641	362810.281305	10.0.4.1	224.0.0.9	RIPv2	66	Response
644	362845.283309	10.0.4.1	224.0.0.9	RIPv2	66	Response

```
Frame 644: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Ethernet II, Src: 42:00:aa:00:00:08 (42:00:aa:00:00:08), Dst: IPv4mcast_09 (01:00:5e:0
Internet Protocol Version 4, Src: 10.0.4.1, Dst: 224.0.0.9
User Datagram Protocol, Src Port: 520, Dst Port: 520
Routing Information Protocol
  Command: Response (2)
  Version: RIPv2 (2)
  IP Address: 10.0.5.0, Metric: 1
```

IP Address (rip.ip), 4 bytes Packets: 645 · Displayed: 195 (30.2%) Profile: Default

2.1. Stable Network

Split Horizon*

A node **CANNOT** send those routes it learned from each neighbor back to that neighbor.

E.g., A with (C, 2, B) will not send route (C, 2) in routing update to B

To prevent **routing loops** that involve **only two gateways**.

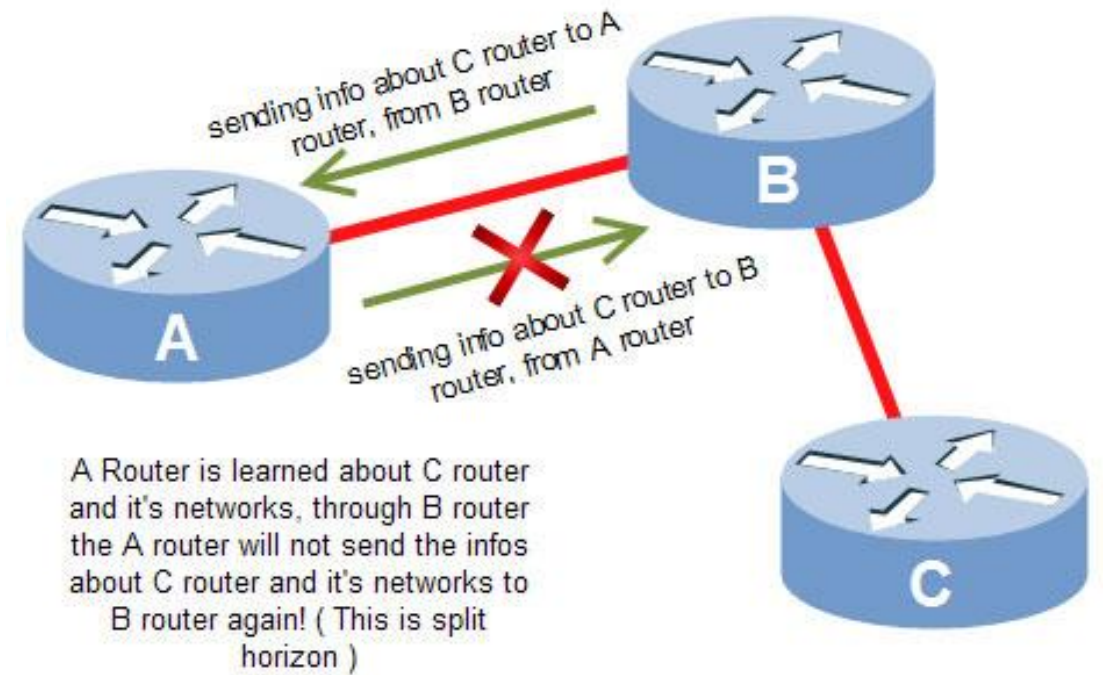
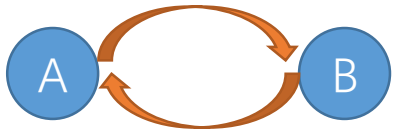
Without Split Horizon:

Count to infinity after node C is down

→ Node B sets (C,inf)

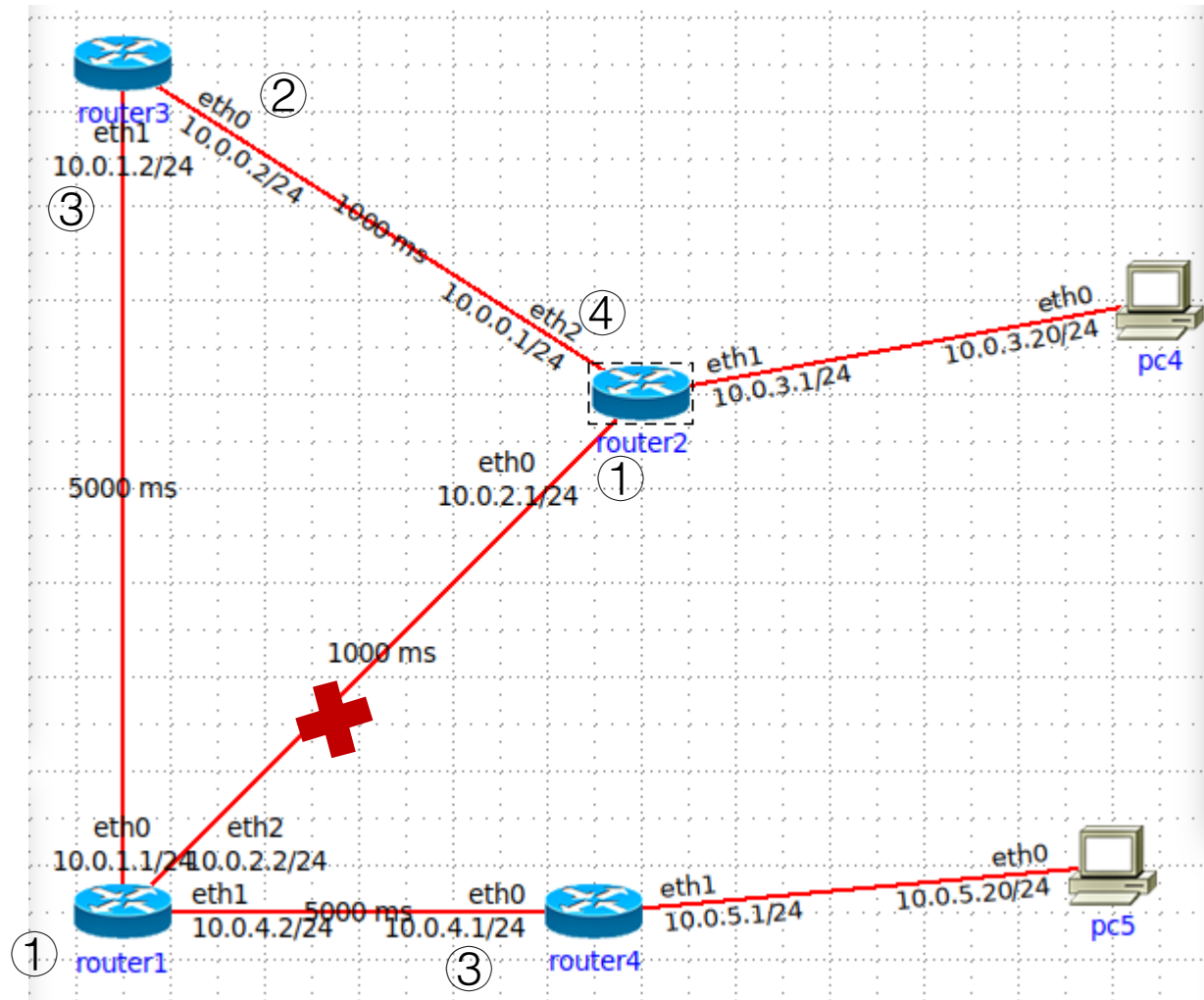
→ Node B receives (C,2) from A

→ Node B sets (C, 3, A) ...



*Split Horizon: <https://tools.ietf.org/html/rfc1058#section-2.2.1>

2.2. Normal Case

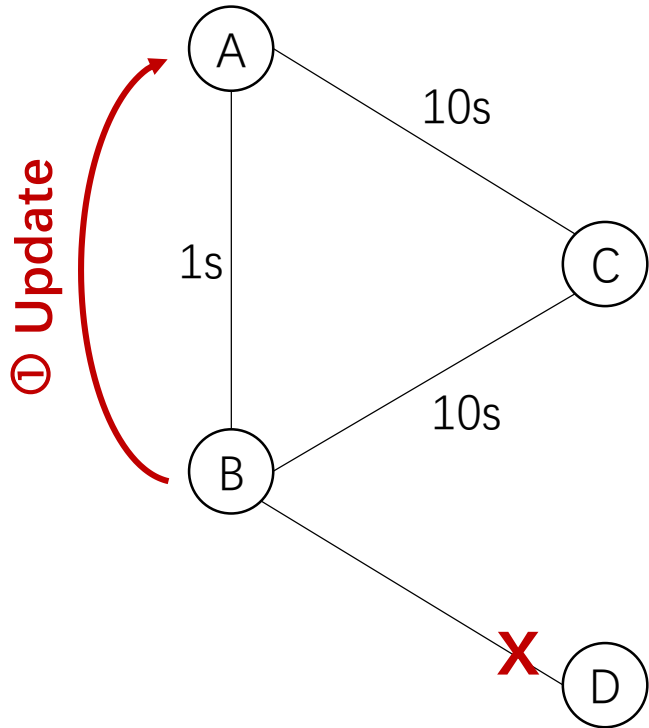


Triggered Updates

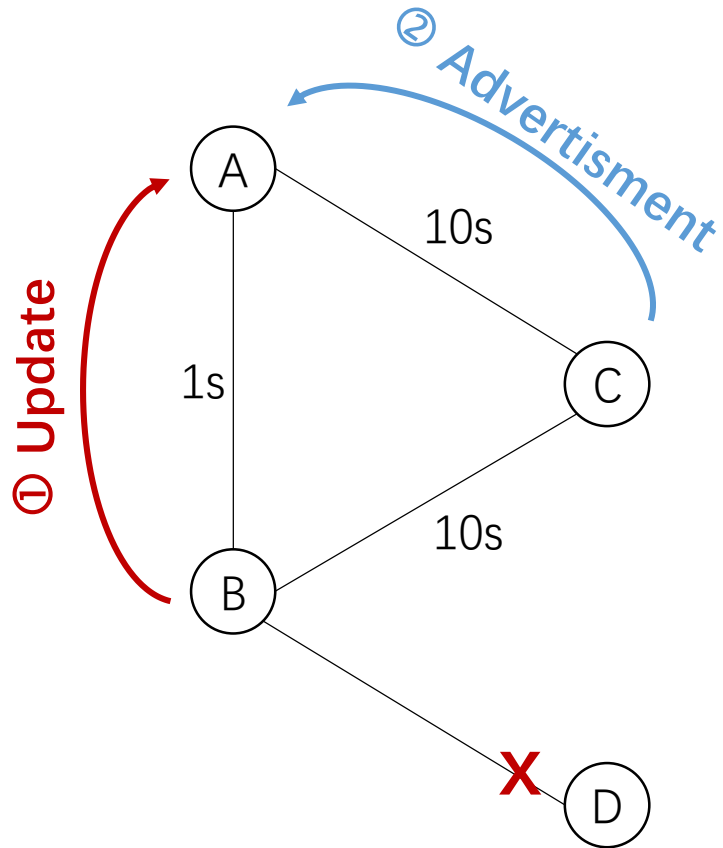
- ① **T=0: Router1&2** detect link failure
→ update their routing tables
→ send RIP advertisement to their neighbors
- ② **T=1s: Router3** eth0 receives RIP response from router2 eth2
→ update routing table
- ③ **T=5s: Router3** eth1 & **Router4** eth0 receive RIP response from router1 eth0 & router1 eth1
→ update routing table
- ④ **T=6s: Router2** eth2 receive RIP response from router3 eth0
→ update routing table

.....

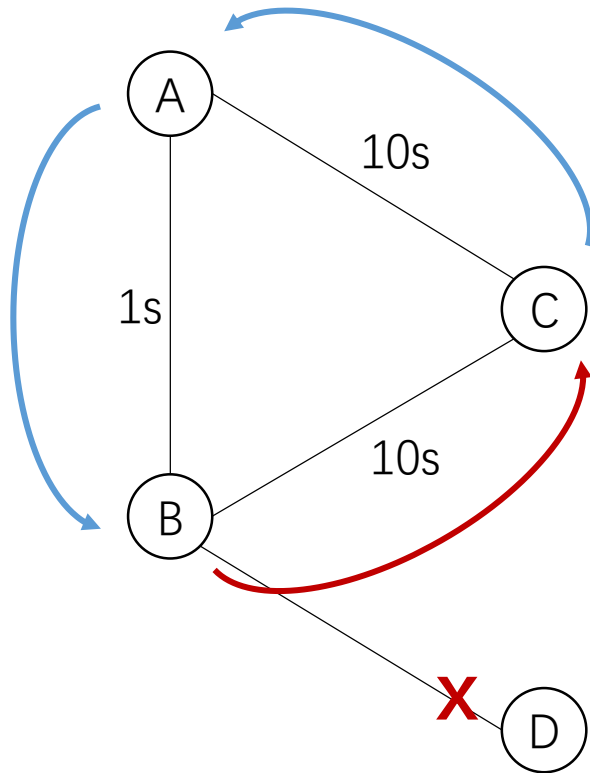
2.3 Count to Infinity Case

[illegible]

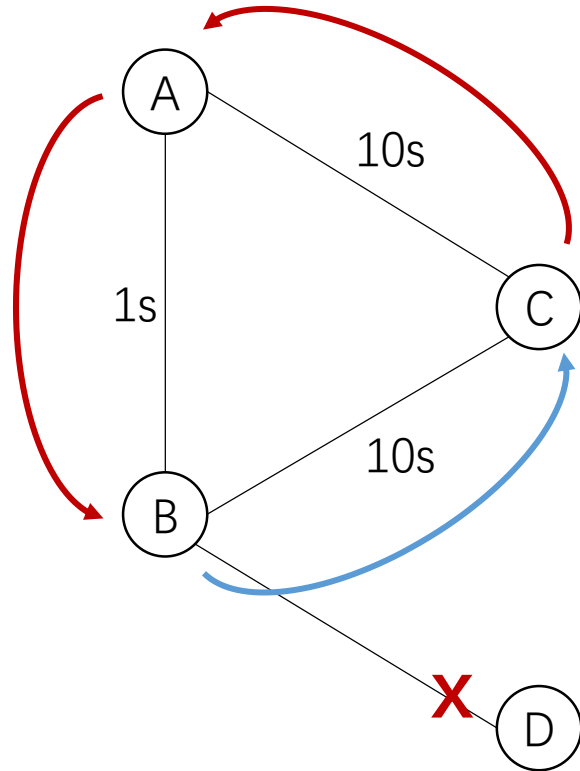
2.3 Count to Infinity Case

[illegible]

2.3 Count to Infinity Case

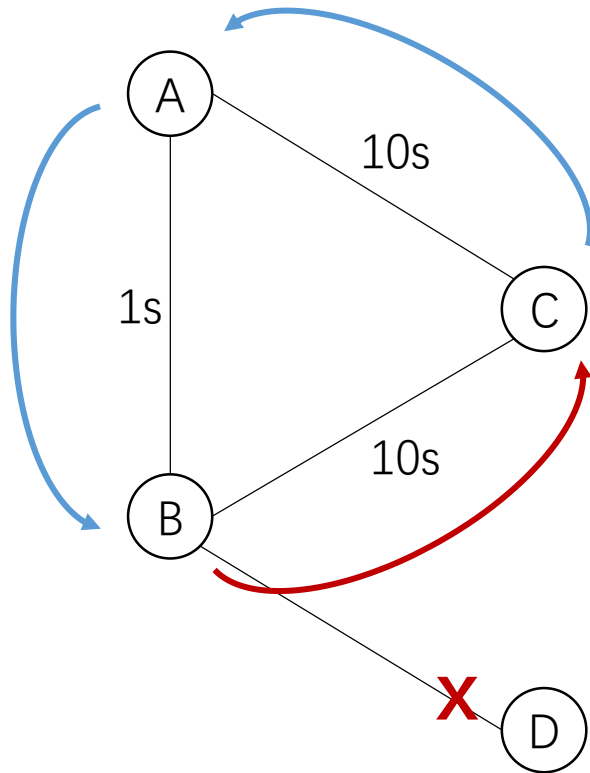
[illegible]

2.3 Count to Infinity Case



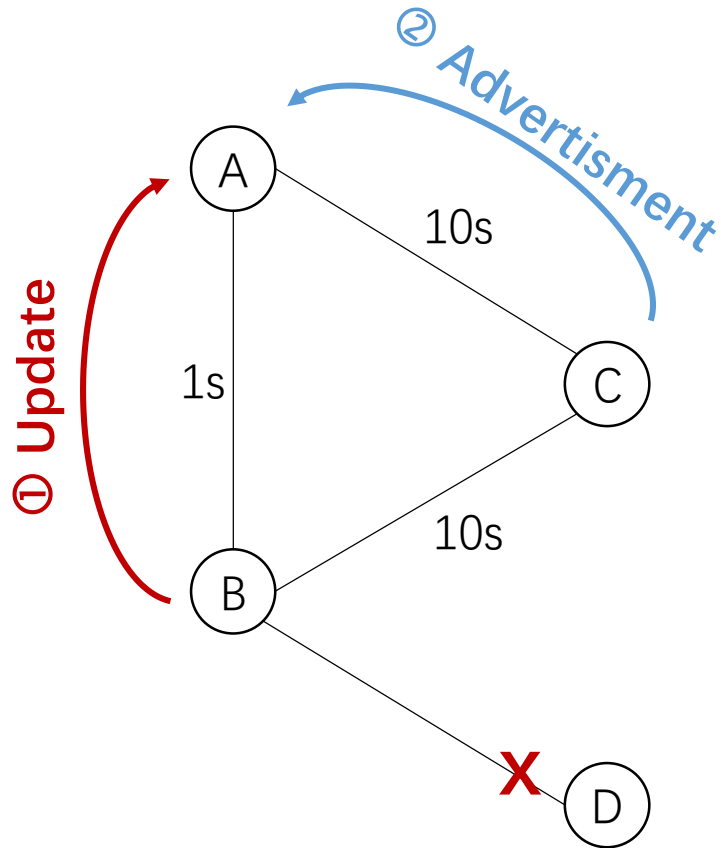
T	A	B	C
<0	2	1	2
0	2	16	2
1	16	16	2
n=3	16	16	2
10	16	16	16
n+10	3	16	16
n+11	3	4	16
20	16	4	16
21	16	16	16
n+11+10	16	16	5

2.3 Count to Infinity Case



T	A	B	C
<0	2	1	2
0	2	16	2
1	16	16	2
n=3	16	16	2
10	16	16	16
n+10	3	16	16
n+11	3	4	16
20	16	4	16
21	16	16	16
n+11+10	16	16	5
31	16	16	16
n+31	6	16	16
n+32	6	7	16

2.3 Count to Infinity Case



	T	A	B	C
①	<0	2	1	2
	0	2	16	2
	1	16	16	2
	n=3	16	16	2
	10	16	16	16
	n+10	3	16	16
	n+11	3	4	16
	20	16	4	16
	21	16	16	16
	n+11+10	16	16	5
	31	16	16	16
	n+31	6	16	16
	n+32	6	7	16

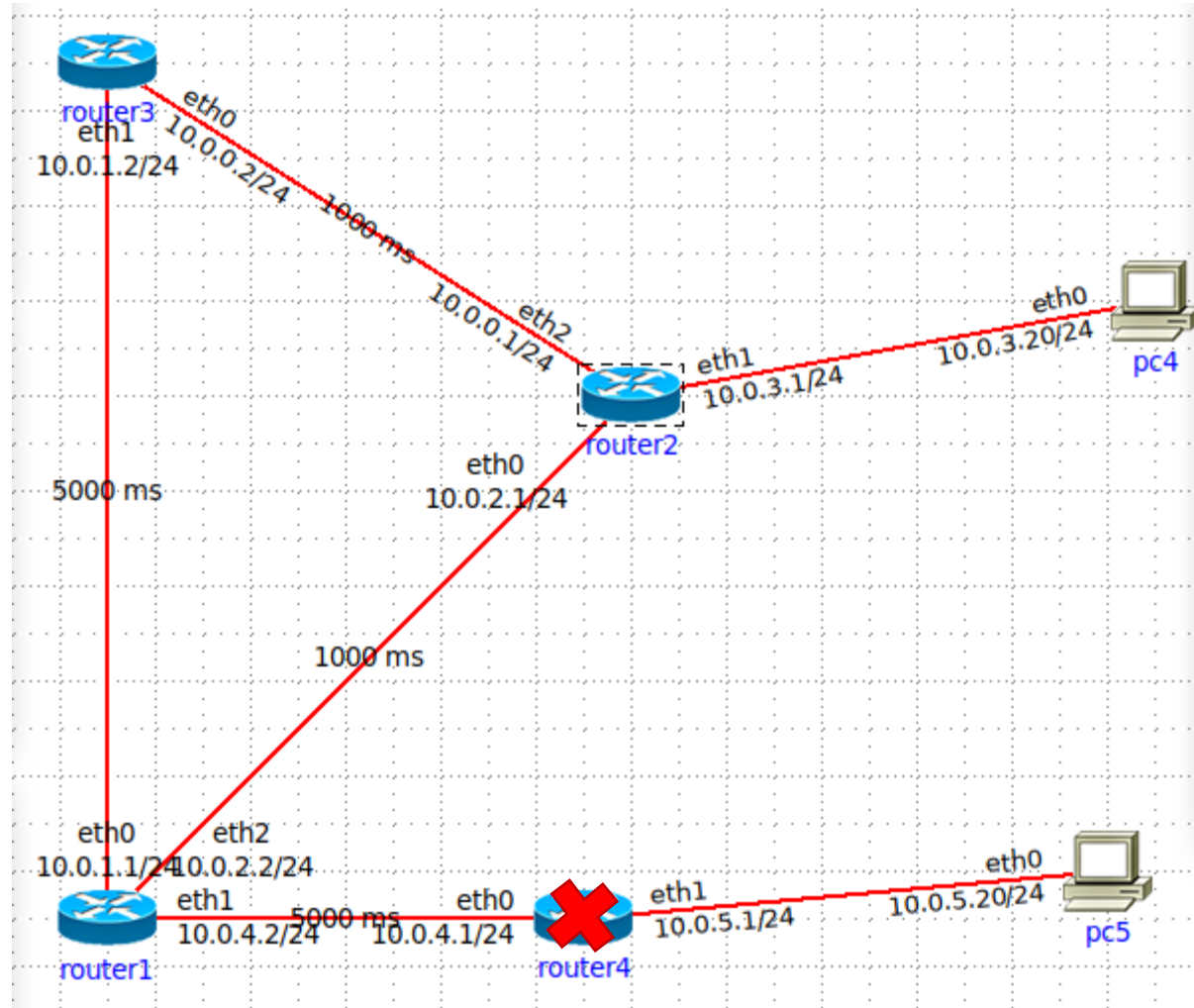
,where $-9 < n < 10$

2.3 Count to Infinity Case

Simulation

2.4. Silent Failure Case

Kill RIP process of Router4



2.4. Silent Failure Case

Kill RIP process of Router4

1

```
Every 1.0s: vtysh -c "show ip rip" router1: Thu Oct 29 11:12:54 2020
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
  (n) - normal, (s) - static, (d) - default, (r) - redistribute,
  (i) - interface

Network        Next Hop        Metric From      Tag Time
R(n) 10.0.0.0/24 10.0.2.1         2 10.0.2.1       0 02:45
C(i) 10.0.1.0/24 0.0.0.0          1 self           0
C(i) 10.0.2.0/24 0.0.0.0          1 self           0
R(n) 10.0.3.0/24 10.0.2.1         2 10.0.2.1       0 02:45
C(i) 10.0.4.0/24 0.0.0.0          1 self           0
R(n) 10.0.5.0/24 10.0.4.1         2 10.0.4.1       0 02:52
```

3

```
Every 1.0s: vtysh -c "show ip rip" router1: Thu Oct 29 11:15:56 2020
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
  (n) - normal, (s) - static, (d) - default, (r) - redistribute,
  (i) - interface

Network        Next Hop        Metric From      Tag Time
R(n) 10.0.0.0/24 10.0.2.1         2 10.0.2.1       0 02:47
C(i) 10.0.1.0/24 0.0.0.0          1 self           0
C(i) 10.0.2.0/24 0.0.0.0          1 self           0
R(n) 10.0.3.0/24 10.0.2.1         2 10.0.2.1       0 02:47
C(i) 10.0.4.0/24 0.0.0.0          1 self           0
R(n) 10.0.5.0/24 10.0.4.1        16 10.0.4.1       0 01:54
```

2

```
Every 1.0s: vtysh -c "show ip rip" router1: Thu Oct 29 11:15:44 2020
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
  (n) - normal, (s) - static, (d) - default, (r) - redistribute,
  (i) - interface

Network        Next Hop        Metric From      Tag Time
R(n) 10.0.0.0/24 10.0.2.1         2 10.0.2.1       0 02:53
C(i) 10.0.1.0/24 0.0.0.0          1 self           0
C(i) 10.0.2.0/24 0.0.0.0          1 self           0
R(n) 10.0.3.0/24 10.0.2.1         2 10.0.2.1       0 02:53
C(i) 10.0.4.0/24 0.0.0.0          1 self           0
R(n) 10.0.5.0/24 10.0.4.1         2 10.0.4.1       0 00:03
```

Detect failure: 180s

+

Delete infinity entry: 120s

3. OSPF* (Open Shortest Path First)

3.1. Stable Network

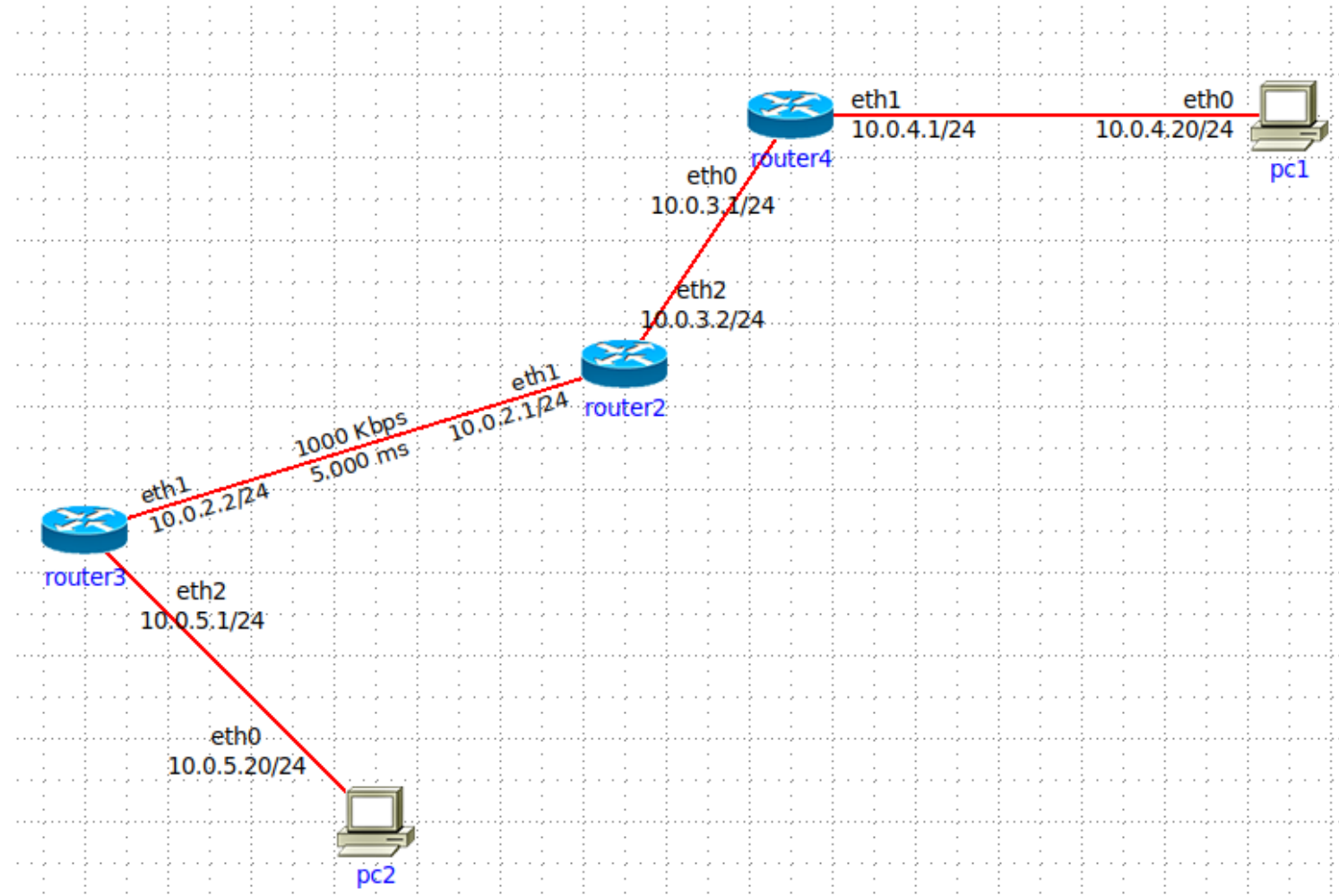
3.2. Add New Node

3.3. Cost in OSPF

3.4. Silent Failure

*OSPFv2: <https://tools.ietf.org/html/rfc2328>

3.1. Stable Network



*The OSPF topology file is ospf.imn, open it in the IMUNES.

3.1. Stable Network

OSPF Routing Table

```
router1# show ip route ospf
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

O    10.0.0.0/24 [110/10] is directly connected, eth0, 00:02:45
O    10.0.1.0/24 [110/10] is directly connected, eth1, 00:02:45
O>*  10.0.3.0/24 [110/20] via 10.0.1.1, eth1, 00:01:57
O>*  10.0.4.0/24 [110/30] via 10.0.1.1, eth1, 00:01:56
O>*  127.0.0.1/32 [110/0] is directly connected, lo, 00:02:45
```

- ① Destination Network
- ② Administrative Distance* / Cost
- ③ Next Hop
- ④ Existence Time

```
router1#
router1# show ip ospf neighbor
```

OSPF Neighbour Table

Neighbor ID	Pri	State	Dead Time	Address	Interface
10.0.3.2	1	Full/DR	30.135s	10.0.1.1	eth1:10.0.1.2

```
router1#
router1# show ip ospf database
```

OSPF Link State Database

OSPF Router with ID (10.0.1.2)

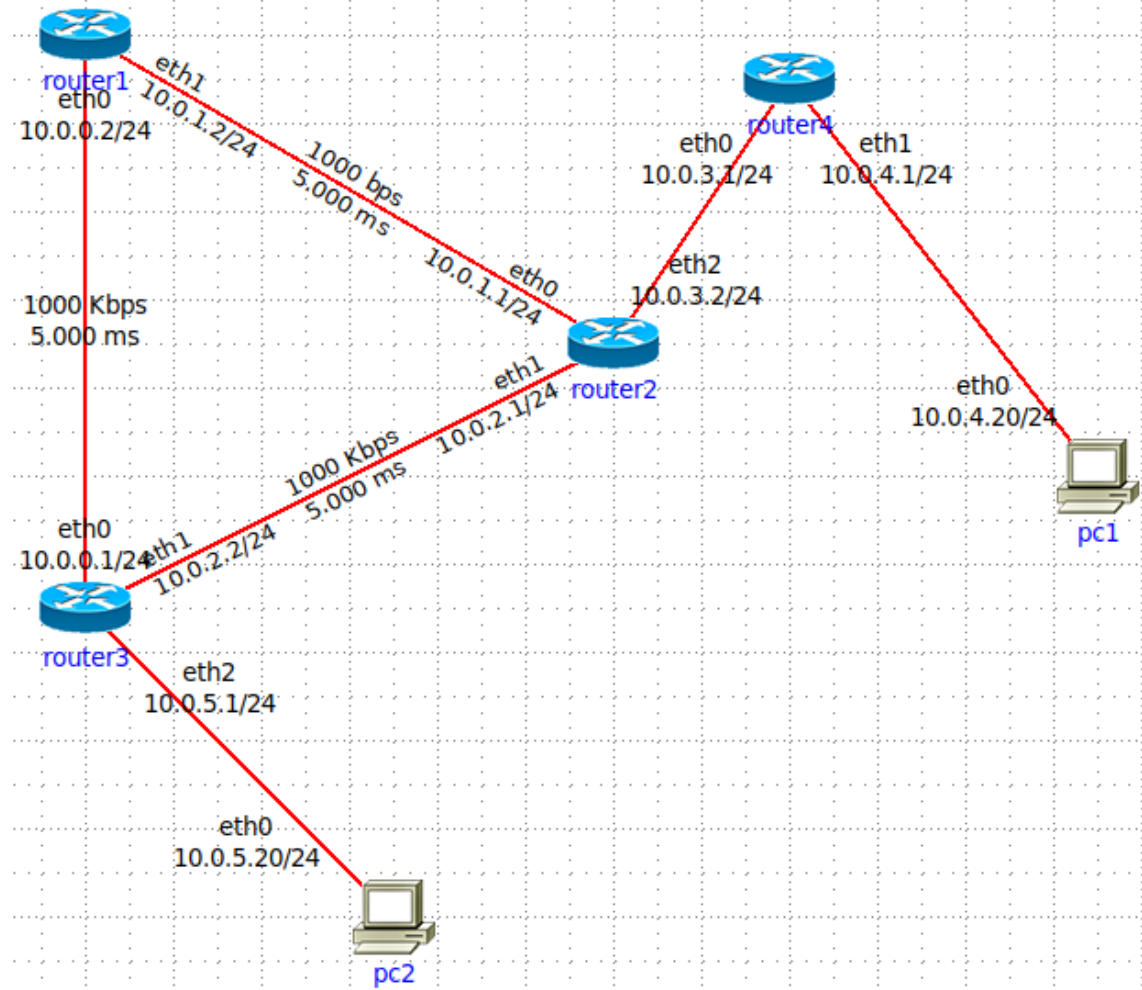
Router Link States (Area 0.0.0.0)

Link ID	ADV Router	Age	Seq#	CkSum	Link count
10.0.1.2	10.0.1.2	133	0x80000007	0x76e3	3
10.0.3.2	10.0.3.2	136	0x80000008	0x3a0a	3
10.0.4.1	10.0.4.1	139	0x80000006	0x1c34	3

Net Link States (Area 0.0.0.0)

Link ID	ADV Router	Age	Seq#	CkSum
10.0.1.1	10.0.3.2	136	0x80000001	0x7ba7
10.0.3.1	10.0.4.1	138	0x80000001	0x7da1

3.2. Add New Node



*The OSPF topology file is ospf.imn, open it in the IMUNES.

3.2. Add New Node

```
router1# show ip route ospf
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

O    10.0.0.0/24 [110/10] is directly connected, eth0, 00:03:18
O    10.0.1.0/24 [110/10] is directly connected, eth1, 00:02:31
O>*  10.0.2.0/24 [110/20] via 10.0.0.1, eth0, 00:02:27
      *
      via 10.0.1.1, eth1, 00:02:27
O>*  10.0.3.0/24 [110/20] via 10.0.1.1, eth1, 00:02:31
O>*  10.0.4.0/24 [110/30] via 10.0.1.1, eth1, 00:02:26
O>*  10.0.5.0/24 [110/20] via 10.0.0.1, eth0, 00:02:27
O>*  127.0.0.1/32 [110/0] is directly connected, lo, 00:03:20
router1#
router1#
router1# show ip ospf neighbor

Neighbor ID    Pri State           Dead Time Address        Interface
10.0.5.1       1 Full/DR           36.902s 10.0.0.1        eth0:10.0.0.2
10.0.3.2       1 Full/DR           36.092s 10.0.1.1        eth1:10.0.1.2
router1#
router1#
router1# show ip ospf database

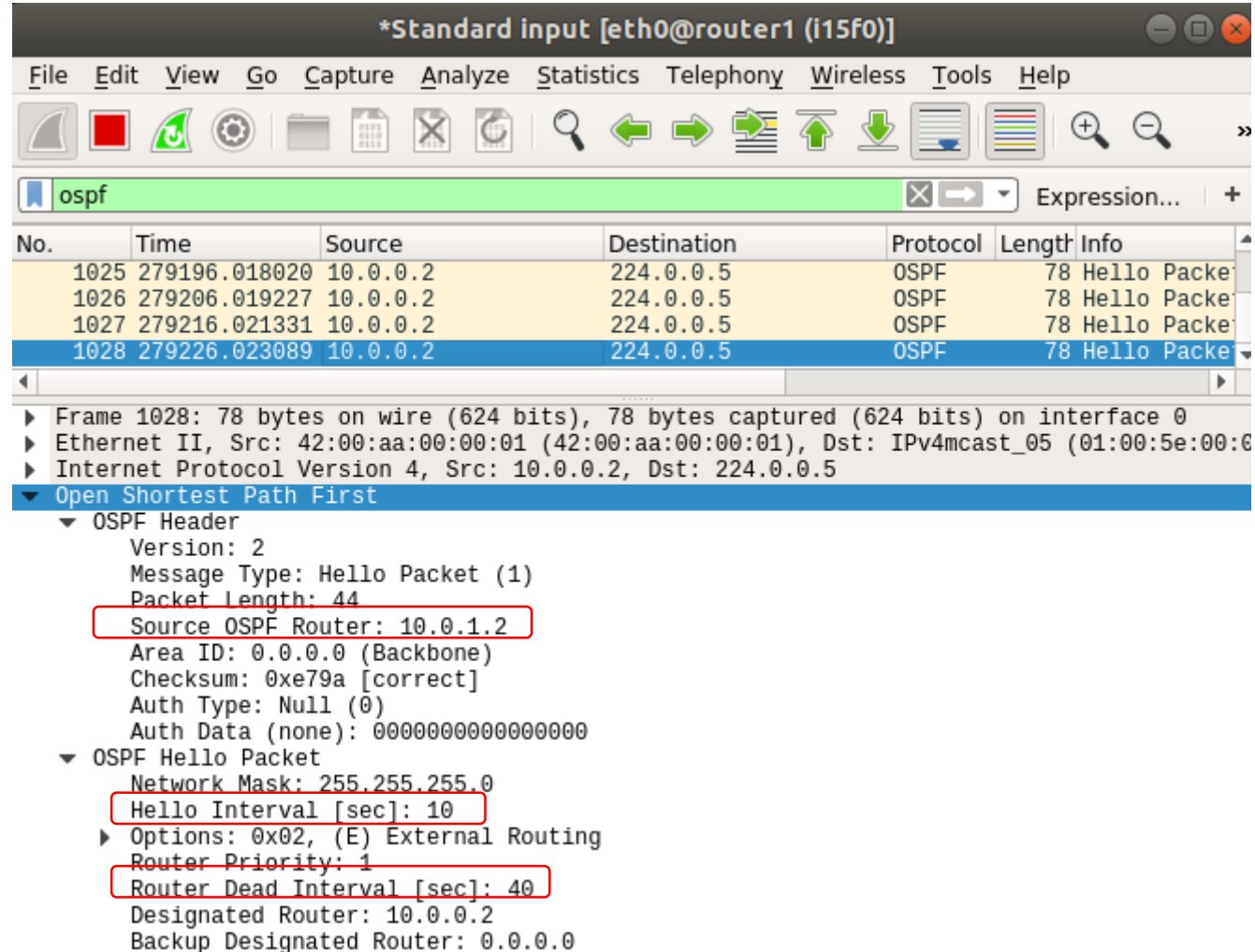
        OSPF Router with ID (10.0.1.2)

                Router Link States (Area 0.0.0.0)

Link ID        ADV Router    Age Seq#           CkSum Link count
10.0.1.2       10.0.1.2      184 0x80000009 0xf754 3
10.0.3.2       10.0.3.2      180 0x8000000c 0x14f7 4
10.0.4.1       10.0.4.1      186 0x80000006 0x1c34 3
10.0.5.1       10.0.5.1      185 0x80000009 0x36e4 4
```

3.2. Add New Node

OSPF Hello - periodic update



The image shows a Wireshark packet capture window titled '*Standard input [eth0@router1 (i15f0)]'. The filter bar shows 'ospf'. The packet list displays four OSPF Hello packets from 10.0.0.2 to 224.0.0.5. Packet 1028 is selected, and its details are expanded. The details pane shows the OSPF Header and OSPF Hello Packet fields, with several values highlighted by red boxes.

No.	Time	Source	Destination	Protocol	Length	Info
1025	279196.018020	10.0.0.2	224.0.0.5	OSPF	78	Hello Packe
1026	279206.019227	10.0.0.2	224.0.0.5	OSPF	78	Hello Packe
1027	279216.021331	10.0.0.2	224.0.0.5	OSPF	78	Hello Packe
1028	279226.023089	10.0.0.2	224.0.0.5	OSPF	78	Hello Packe

Frame 1028: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0
Ethernet II, Src: 42:00:aa:00:00:01 (42:00:aa:00:00:01), Dst: IPv4mcast_05 (01:00:5e:00:00:05)
Internet Protocol Version 4, Src: 10.0.0.2, Dst: 224.0.0.5

▼ Open Shortest Path First

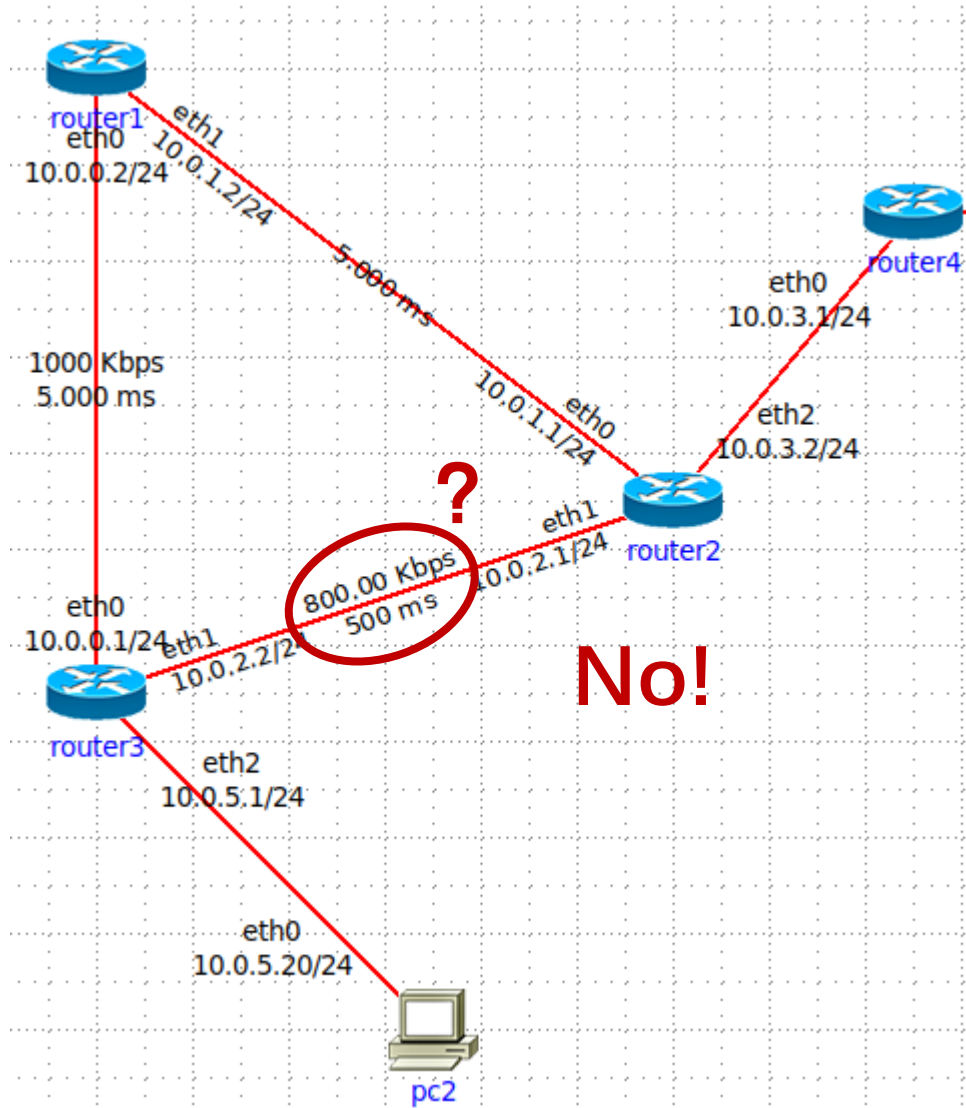
- ▼ OSPF Header
 - Version: 2
 - Message Type: Hello Packet (1)
 - Packet Length: 44
 - Source OSPF Router: 10.0.1.2
 - Area ID: 0.0.0.0 (Backbone)
 - Checksum: 0xe79a [correct]
 - Auth Type: Null (0)
 - Auth Data (none): 0000000000000000
- ▼ OSPF Hello Packet
 - Network Mask: 255.255.255.0
 - Hello Interval [sec]: 10
 - Options: 0x02, (E) External Routing
 - Router Priority: 1
 - Router Dead Interval [sec]: 40
 - Designated Router: 10.0.0.2
 - Backup Designated Router: 0.0.0.0

3.2. Add New Node

Type 1 LSA

```
▼ LSA-type 1 (Router-LSA), len 72
  .000 0000 0000 0010 = LS Age (seconds): 2
  0... .. = Do Not Age Flag: 0
  ▶ Options: 0x02, (E) External Routing
  LS Type: Router-LSA (1)
  Link State ID: 10.0.0.1
  Advertising Router: 10.0.0.1
  Sequence Number: 0x8000000b
  Checksum: 0x80a2
  Length: 72
  ▶ Flags: 0x02, (E) AS boundary router
  Number of Links: 4
  ▼ Type: Transit ID: 10.0.0.2          Data: 10.0.0.1          Metric: 10
    Link ID: 10.0.0.2 - IP address of Designated Router
    Link Data: 10.0.0.1
    Link Type: 2 - Connection to a transit network
    Number of Metrics: 0 - TOS
    0 Metric: 10
  ▶ Type: Transit ID: 10.0.2.1          Data: 10.0.2.2          Metric: 10
  ▶ Type: Stub ID: 10.0.5.0             Data: 255.255.255.0    Metric: 10
  ▶ Type: Stub ID: 127.0.0.1           Data: 255.255.255.255 Metric: 0
```

3.3. Cost in OSPF



Cost = $10^8 / (\text{BW of interface in bps})$

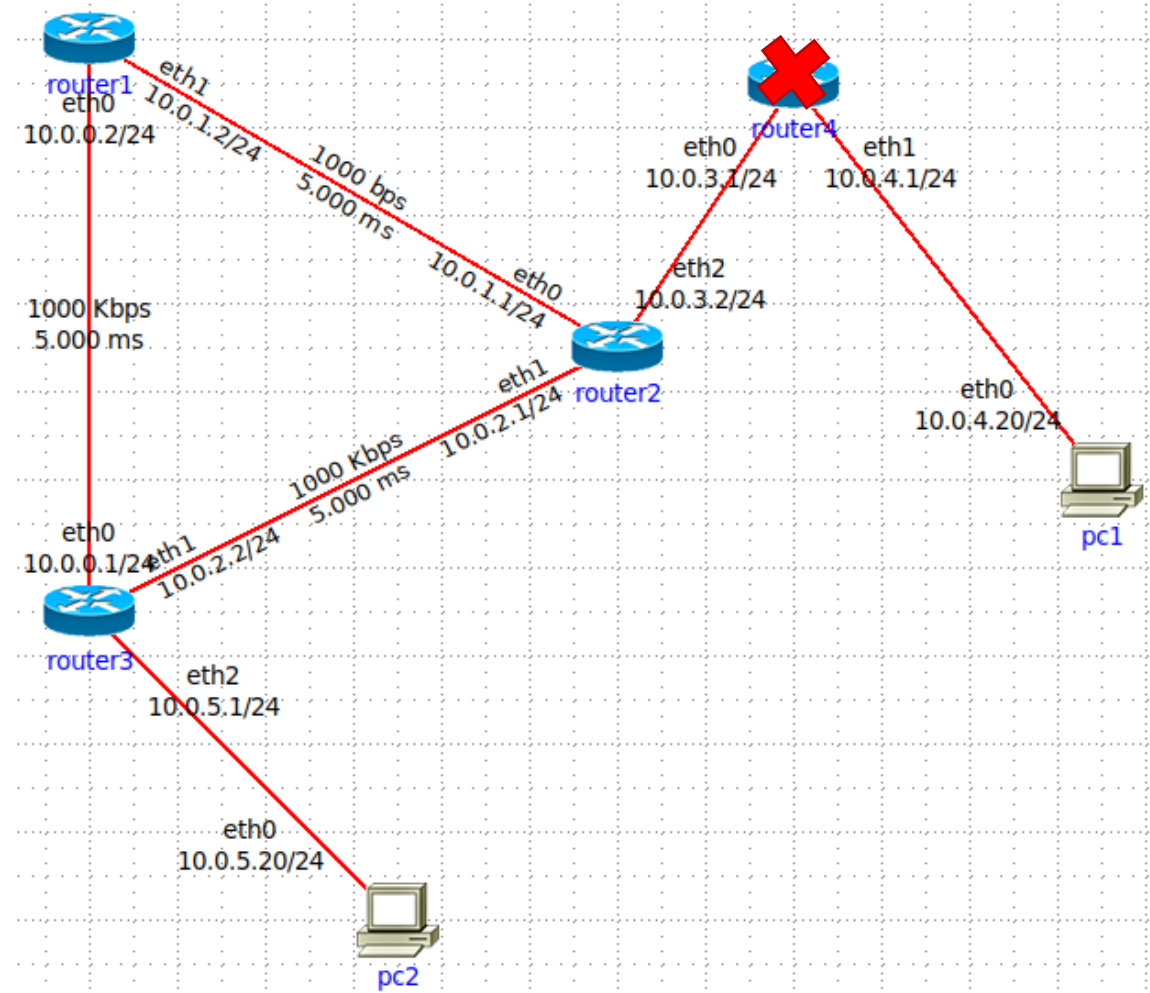
```
eth1 is up
  ifindex 79, MTU 1500 bytes, BW 0 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.0.1.2/24, Broadcast 10.0.1.255, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 10.0.1.2, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State Backup, Priority 1
  Designated Router (ID) 10.0.3.2, Interface Address 10.0.1.1
  Backup Designated Router (ID) 10.0.1.2, Interface Address 10.0.1.2
  Saved Network-LSA sequence number 0x80000005
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
  Hello due in 8.439s
  Neighbor Count is 1, Adjacent neighbor count is 1
```

```
eth1 is up
  ifindex 79, MTU 1500 bytes, BW 1000 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.0.1.2/24, Broadcast 10.0.1.255, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 10.0.1.2, Network Type BROADCAST, Cost: 100
  Transmit Delay is 1 sec, State Backup, Priority 1
```

```
eth1 is up
  ifindex 79, MTU 1500 bytes, BW 1000000 Kbit <UP,BROADCAST,RUNNING,MULTICAST>
  Internet Address 10.0.1.2/24, Broadcast 10.0.1.255, Area 0.0.0.0
  MTU mismatch detection:enabled
  Router ID 10.0.1.2, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State Backup, Priority 1
```

3.4. Silent Failure

Kill OSPF process of Router4



3.4. Silent Failure

Kill OSPF process of Router4

①

```
0>* 10.0.0.0/24 [110/20] via 10.0.1.2, eth0, 00:05:32
*
0 10.0.1.0/24 [110/10] is directly connected, eth0, 00:06:23
0 10.0.2.0/24 [110/10] is directly connected, eth1, 00:07:30
0 10.0.3.0/24 [110/10] is directly connected, eth2, 00:07:30
0>* 10.0.4.0/24 [110/20] via 10.0.3.1, eth2, 00:02:20
0>* 10.0.5.0/24 [110/20] via 10.0.2.2, eth1, 00:05:32
0>* 127.0.0.1/32 [110/0] is directly connected, lo, 00:07:30

Every 1.0s: vtysh -c "show ip ospf neighbor" router2: Thu 0
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
10.0.1.2	1	Full/DR	31.372s	10.0.1.2	eth0:10.0.1.1
10.0.5.1	1	Full/Backup	37.797s	10.0.2.2	eth1:10.0.2.1
10.0.4.1	1	Full/Backup	38.950s	10.0.3.1	eth2:10.0.3.2

③

```
0>* 10.0.0.0/24 [110/20] via 10.0.1.2, eth0, 00:06:22
*
0 10.0.1.0/24 [110/10] is directly connected, eth0, 00:07:03
0 10.0.2.0/24 [110/10] is directly connected, eth1, 00:08:13
0 10.0.3.0/24 [110/10] is directly connected, eth2, 00:08:13
0>* 10.0.5.0/24 [110/20] via 10.0.2.2, eth1, 00:06:22
0>* 127.0.0.1/32 [110/0] is directly connected, lo, 00:08:13

Every 1.0s: vtysh -c "show ip ospf neighbor" router2: Thu 0
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
10.0.1.2	1	Full/DR	39.232s	10.0.1.2	eth0:10.0.1.1
10.0.5.1	1	Full/Backup	37.573s	10.0.2.2	eth1:10.0.2.1

②

```
0>* 10.0.0.0/24 [110/20] via 10.0.1.2, eth0, 00:06:13
*
0 10.0.1.0/24 [110/10] is directly connected, eth0, 00:06:57
0 10.0.2.0/24 [110/10] is directly connected, eth1, 00:08:04
0 10.0.3.0/24 [110/10] is directly connected, eth2, 00:08:04
0>* 10.0.4.0/24 [110/20] via 10.0.3.1, eth2, 00:02:54
0>* 10.0.5.0/24 [110/20] via 10.0.2.2, eth1, 00:06:13
0>* 127.0.0.1/32 [110/0] is directly connected, lo, 00:08:04

Every 1.0s: vtysh -c "show ip ospf neighbor" router2: Thu 0
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
10.0.1.2	1	Full/DR	38.380s	10.0.1.2	eth0:10.0.1.1
10.0.5.1	1	Full/Backup	34.792s	10.0.2.2	eth1:10.0.2.1
10.0.4.1	1	Full/Backup	3.954s	10.0.3.1	eth2:10.0.3.2

4 Hello packets (40s) to detect failure.
Delete the table entry immediately.

3.4. Silent Failure

Simulation

Questions?

Source Code of Routing Algorithms

1. RIP

<https://github.com/Quagga/quagga/tree/master/ripd>

2. OSPF

<https://github.com/Quagga/quagga/tree/master/ospfd>