Configuration and analysis of the performance of RIP routing protocol

Introduction

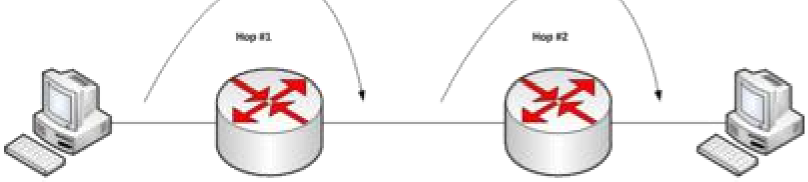
Routing involves the network topology, or the setup of hardware, that can effectively relay data. Standard protocols help to identify the best routes for data and to ensure quality transmission. Individual pieces of hardware such as routers are referred to as "nodes" in the network. Different algorithms and protocols can be used to figure out how to best route data packets, and which nodes should be used. For example, some data packets travel according to a distance vector model that primarily uses distance as a factor, whereas others use Link-State Protocol, which involves other aspects of a "best path" for data.
Data packets are also made to give networks information. Headers on packets provide details about origin and destination. Standards for data packets allow for conventional design, which can help with future routing methodologies.

Routing Information Protocol (RIP) is a dynamic routing protocol which uses hop count as a routing metric to find the best path between the source and the destination network. It is a distance vector routing protocol which has AD value 120 and works on the application layer of OSI model. RIP uses port number 520.

A **distance-vector routing protocol** in data networks determines the best route for data packets based on distance. Distance-vector routing protocols measure the distance by the number of routers a packet has to pass, one router counts as one hop. Some distance-vector protocols also take into account network latency and other factors that influence traffic on a given route. To determine the best route across a network routers, on which a distance-vector protocol is implemented, exchange information with one another, usually routing tables plus hop counts for destination networks and possibly other traffic information. Distance-vector routing protocols also require that a router informs its neighbours of network topology changes periodically.

Hop Count

Hop count is the number of routers occurring in between the source and destination network. The path with the lowest hop count is considered as the best route to reach a network and therefore placed in the routing table. RIP prevents routing loops by limiting the number of hopes allowed in a path from source and destination. The maximum hop count allowed for RIP is 15 and hop count of 16 is considered as network unreachable.



Features of RIP :

1. Updates of the network are exchanged periodically.
2. Updates (routing information) are always broadcast.
3. Full routing tables are sent in updates.
4. Routers always trust on routing information received from neighbor routers.

RIP timers

- **Update timer :** The default timing for routing information being exchanged by the routers operating RIP is 30 seconds. Using Update timer, the routers exchange their routing table periodically.
- **Invalid timer:** If no update comes until 180 seconds, then the destination router consider it as invalid. In this scenario, the destination router mark hop count as 16 for that router.
- **Hold down timer :** This is the time for which the router waits for neighbour router to respond. If the router isn't able to respond within a given time then it is declared dead. It is 180 seconds by default.
- **Flush time :** It is the time after which the entry of the route will be flushed if it doesn't respond within the flush time. It is 60 seconds by default. This timer starts after the route has been declared invalid and after 60 seconds i.e time will be 180 + 60 = 240 seconds.

Forwarding Table

The routers in autonomous systems keep forwarding tables to forward packets to their destination network. A forwarding table in RIP is a three-column table in which the first column is the address of the destination network, the second column is the address of the next router to which the packet should be forwarded. The third column is the cost (number of hops) to reach the destination network.

RIP Implementation

RIP is implemented as a process that uses the service of UDP on the well-known port number 520. RIP is a daemon process. RIP is a routing to help IP route its datagrams through the AS, the RIP messages are encapsulated inside UDP user datagrams, which in turn are encapsulated inside IP datagrams. In other words, RIP runs at the application layer but creates forwarding tables for IP at the network layer

RIP has gone through two versions: RIP-1 and RIP-2. The second version is backward compatible with the first version.

RIP Messages

A client and a server, like any other processes, need to exchange message. RIP-2 defines the format of the message. Part f the message, which is called, can be repeated as needed in a message. Each entry carries the information related to one line in the forwarding table of the router that sends the message.

RIP has two types of messages: **request** and **response**. A request message is sent by a router that has just come up or by router that has some time-out entries. A request message can ask about specific entries or all entries. A response (or update) message can be either solicited or unsolicited. A solicited response message is sent only in answer to a request message. It contains information about the destination specified in the corresponding request message. An unsolicited response message, on the other hand, is send periodically, every 30 seconds or when there is a change in the forwarding table.

Components used and their arrangement

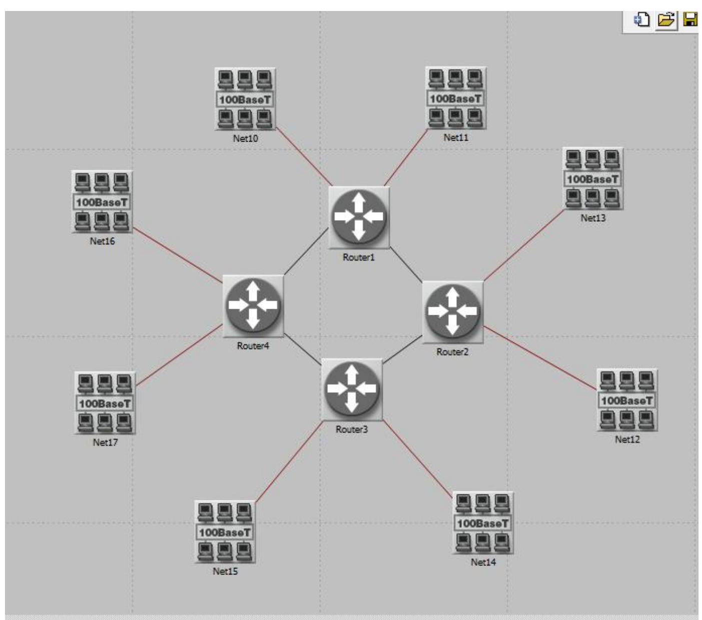
1. ethernet4_slip8_gtwy : The ethernet4_slip8_gtwy node model represents an IP-based gateway supporting four Ethernet hub interfaces and eight serial line interfaces. IP packets arriving on any interface are routed to the appropriate output interface based on their destination IP address.

2. 100BaseT_LAN : It is used to represent a Fast Ethernet LAN in a switched topology. The object contains any number of clients as well as one server. Client traffic can be directed to the internal server as well as external servers.

3. 100 BASE-T links : The 100BaseT duplex link represents an Ethernet connection operating at 100 Mbps.

4. PPP_DS3 : A DS3 line (sometimes also referred to as T3) is a dedicated, high bandwidth, fiber circuit . The PPP_DS3 link has a data rate of 44.736 Mbps. Connects two nodes running IP.

5. Link Failure/Recovery object: This controller node can be used to model failure-recovery scenarios in a given model. It provides attributes for controlling the time and status of objects in the model.



Methodology

1. Each router is connected to two 100BaseT_LANs via 100BaseT links. The routers are connected to one another via PPP_DS3 links.

2. Choose the Statistics To test the performance of the RIP, we will collect the following statistics

   Right-click anywhere in the project workspace and select Choose Individual Statistics from the pop-up menu.

   In the Choose Results dialog box, check the following statistics:

   a. **Global Statistics** · **RIP** · **Traffic Sent (bits/sec).**

   b. **Global Statistics** · **RIP** · **Traffi c Received (bits/sec).**

   c. **Nodes Statistics** · **Route Table** · **Total Number of Updates.**

<u>For scenario 1 (NO_FAILURE):</u>

The routers in this network will build their routing tables with no need for further updating because we don't simulate any node or link failures.

3. Configure the Simulation

(i) Click on and the Configure Simulation window should appear.

(ii) Set the duration to be 10.0 minutes.

(iii) Click on the **Global Attributes** tab and change the following attributes:

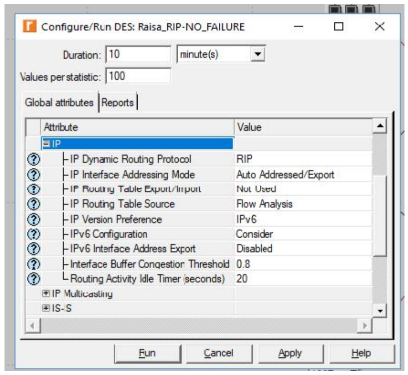
a. **IP Dynamic Routing Protocol** = **RIP** . This sets the RIP protocol to the routing

protocol of all routers in the network.

b. **IP Interface Addressing Mode** = **Auto Addressed/Export** .

c. **RIP Sim Efficiency** = **Disabled** . This makes RIP keep updating the routing table in

case there are any changes in the network



For scenario 2 (FAILURE):

In this scenario, we will simulate failures so that we can compare the behavior of the routers in both cases.

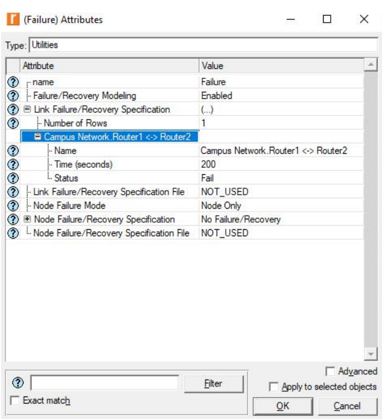
1. Duplicate the previous scenario.

2. Add a **Failure Recovery** object to your workspace and name it **Failure**

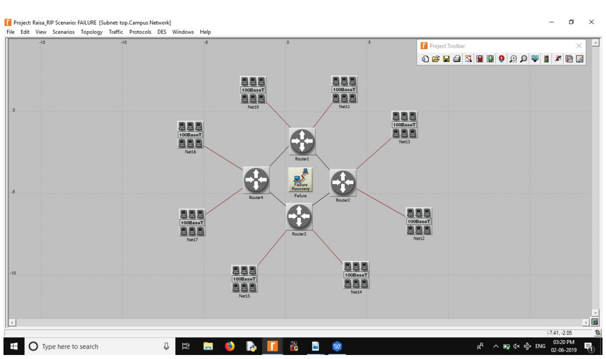
3. Right-click on the **Failure** object · **Edit Attributes**

· Expand the **Link Failure/Recovery Specification** hierarchy · Set **rows** to **1**

· Set the attributes of the added row, **row 0**



Configure the Failure object such that it will “fail” the link between **router_1** and **router_2** 200 s into the simulation.



For scenario 3 (RECOVERY):

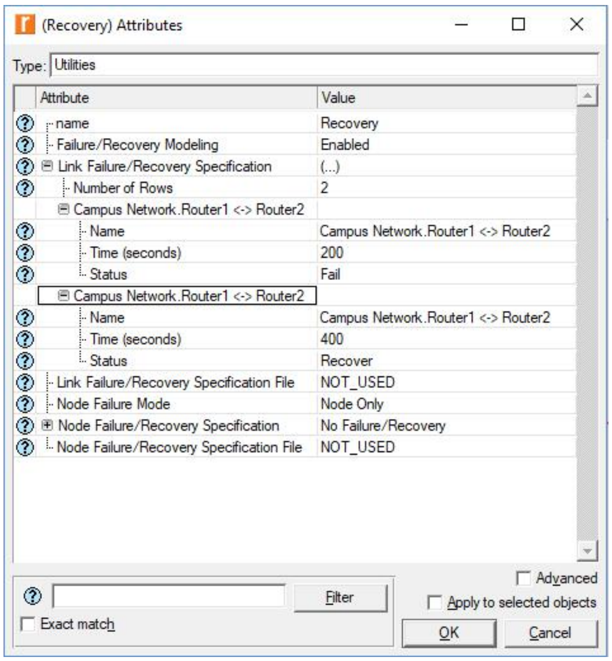
In this scenario, we will simulate recovery from failure and see the how the forwarding tables get updated

1. Duplicate the previous scenario.

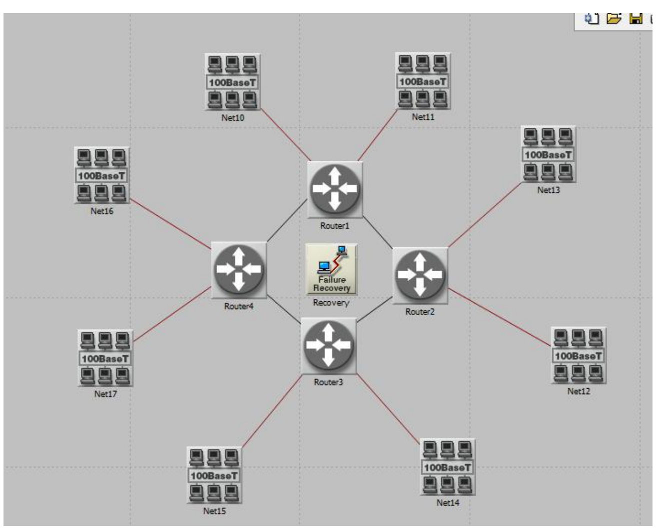
2. Add a **Failure Recovery** object to your workspace and name it **Recovery**

3. Right-click on the **Failure** object · **Edit Attributes**

· Expand the **Link Failure/Recovery Specification** hierarchy · Set **rows** to 2

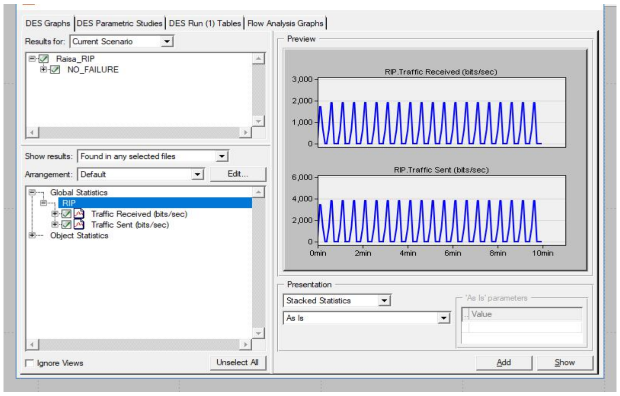


Configure the Failure object such that it will “fail” the link between **Router1** and **Router2** **200 s** into the simulation and it will “recover” between **Router1** and **Router2** at **400s**

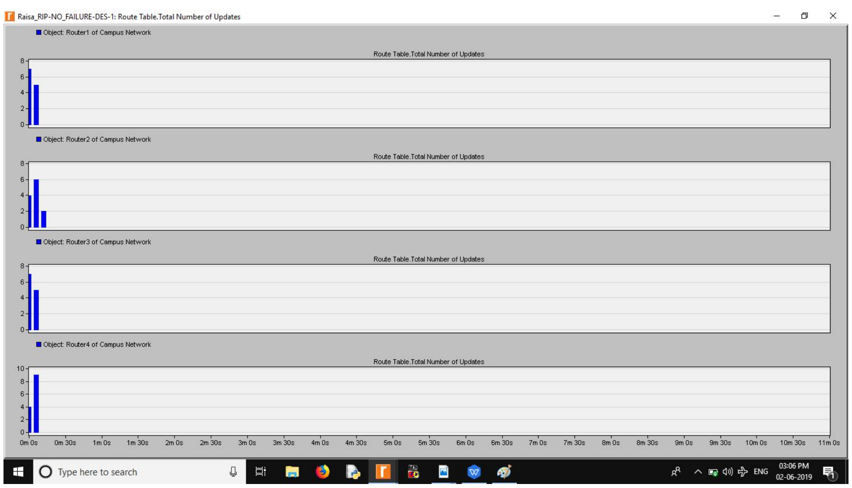


Results and Analysis

Scenario 1 (NO_FAILURE):

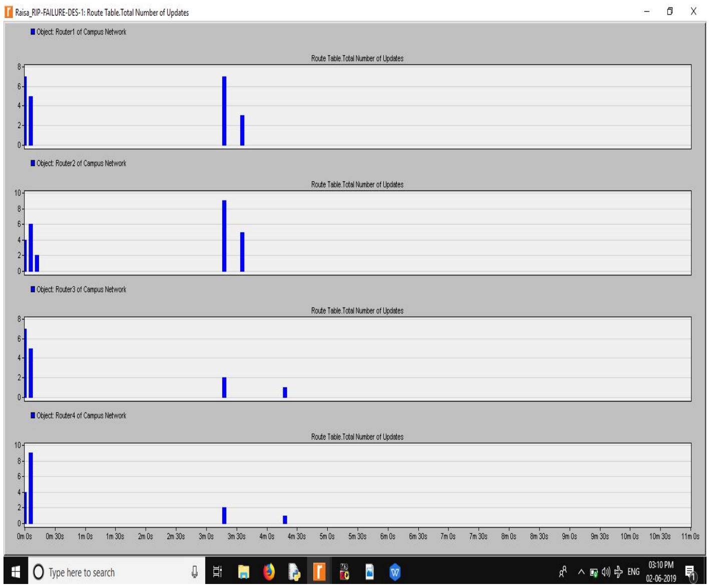


When we observe the traffic sent and traffic received in Scenario1, it is the same since there is no failure in the links and hence the traffic sent and received is the same.



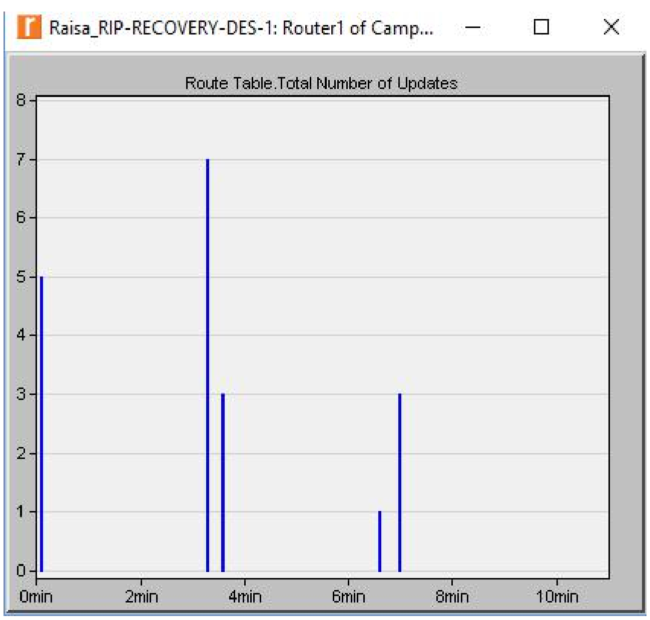
This graph indicates the number of updates made in the routing table when we run the simulation. We can observe that the forwarding table is updated once at the beginning when the details of each simulation when the routers share the information with one another. Since there is no failure in any of the links, the routing table is not updated again.

SCENARIO 2 (FAILURE):

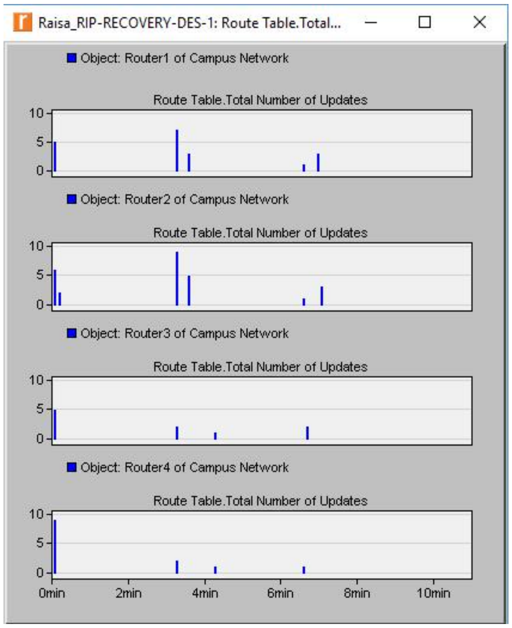


In this scenario, the Router 1 and 2 experience failure at 200 seconds (3m20s). In the above graph, we see that the routing table is updated twice. The router gets updated every time there is a change in any of the links. In this scenario, the forwarding table gets updated when the link failure occurs which is evident from the graph.

SCENARIO 3 (RECOVERY):



In the above graph we observe that the routing table is updated twice. Once at 200s when the link between Router1 and Router2 experiences failure. And then again at 400s when the link between Router1 and Router2 recovers.



In the above graph, we observe that all the nodes get updated on failure and recovery. This is due to the property of RIP routing protocol in which a node sends all the information it knows to its neighbouring nodes. Hence the routing information is passed on by nodes to all the neighbouring nodes and the routing table is updated at those times. Since RIP is dynamic routing, the routing table gets updated at every change.

References

[1] RIP: IETF RFC number 2453 (www.ietf.org/rfc.html).

[2] Aaros Balchunas, “Routing Information Protocol”, Issue 1, April 2012

[3] Jon Larizgoitia Burgaña , “Design and evaluation of a link-state routing protocol for Internet-Wide Geocasting ”, May2017