Load Balance & Congestion Avoidance in EIGRP Networks by using the Characteristics of the OpenFlow Protocol

H.Hasan and J.Cosmas

Electronic & Computer Engineering Department
School of Engineering and Design, Brunel University, London, United Kingdom
e-mail: {eepghah|john.cosmas}@brunel.ac.uk

This research is sponsored by the Higher Committee for Education Development in Iraq, The Office of the Prime Minister, The Green Zone, Baghdad/ Iraq

EIGRP 屬於第三層
Congestion 屬於第四層

Abstract EIGRP and Congestion 有何關係?

In this paper, Enhanced Interior Gateway Routing Protocol (EIGRP) algorithm inside routers will be applied along with simple smart load distribution algorithm inside a controller which results a network that has the characteristics of the OpenFlow protocol.
OpenFlow protocol is a powerful method to deal with changing of traffic volumes across the links and it involves using a Controller that manages traffic across the network.
The aims of this project is to design an algorithm that combines the characteristics of both the traditional routing protocol (EIGRP) and the OpenFlow protocol, the flows from the sources to the destinations will be distributed across the network according to the mentioned algorithm which gives better load balance and avoiding channel congestion as much as possible. Both of the Routers and the Controller will share the responsibility of managing the traffic across the network.

1. Introduction and related work

Load balance has been considered as one of the most important issues in the development of network applications. Load balance can only be achieved in Dynamic Routing by comparing the links' cost of the different routes to the destination. However, this method is limited because the available bandwidth (link cost) over the links may change from time to time as many sources share parts of those links to communicate different destinations, which leads to changing of traffic loads during its active working time.

In dynamic routing networks, there are three types of routing protocols:

- Distance-vector, which finds the best path to the destination by using hop counts, for example: Routing Information Protocol (RIP).
- Link state, also called shortest-path-first protocols, which finds the best path to the destination by using bandwidth, for example: Open Shortest Path First (OSPF).
- Hybrid, which uses aspects of both the distance vector and link state, for example: Enhanced Interior Gateway Routing Protocol (EIGRP).

EIGRP routing protocol is CISCO proprietary routing protocol (Lammle, 2007, p 418).

EIGRP routing algorithm will be used in our project networks and according to this algorithm finding the best route to a specific destination, the following equation is used:

公式錯誤:

$$Metric = \left(\frac{10^7}{Bandwidth} + Delay\right) * 256 \quad (1)$$

正確:(Bandwidth + Delay) * 256

Bandwidth: represents is the least bandwidth of all outgoing links on the path to the destination (bits/seconds).

Delay: represents the sum of the delays configured on the interfaces, links and hops on the path to the destination (seconds).

The route that has the least *Metric* value is called the Successor, which is considered the best route to the destination (the Successor has the highest bandwidth and the lowest delay) (CISCO Web site, 2013).

The other discovered routes to the destination, which have the higher metric values, are called Feasible Successors.

Theoretically, for every destination there is only one successor and there are up to six Feasible Successors (Lammle, 2007, p 420).

Smart distribution of load among the Successor and Feasible Successors leads to better, even usage of all available links. In such load balance, the traffic will be distributed as much as possible among all the available paths, which results in less congestion, more efficiency, better usage of bandwidth and finally better performance (higher throughput) (Long et al., 2013).

Destination route map can be defined as a graphical representation of all the hops (routers) of the network path to a specific destination.

In Software-Defined Networking (SDN), the network's control plane is separated from the data forwarding plane.

This is represented in OpenFlow protocol where the flow tables are used by devices (routers or switches) rather than routing tables or MAC address tables (Klein and Jarschel, 2013) & (OpenFlow org Web Site 2013).

The OpenFlow system should contain at least one Controller, and a secure channel connects between devices (Routers/Switches) and the Controller. This secure channel requires very small bandwidth compared with the other channels of the network that connect routers (McKeown et al., 2008).

The Controller adds or deletes flow entries to the flow tables (Klein and Jarschel, 2013) & (OpenFlow org Web Site 2013).

Our project differs from the Openflow in one point that the discovery of the paths (routes) to the destination is the responsibility of the Router not the Controller, but the controller helps to manage the traffic across the network to provide better possible efficiency.

The reason is that if the controller fails (or the channel between the controller and any of the routers breaks down), the network will still work on with less efficiency, so this creates more reliable network. The load balance along with Openflow is discussed by several research papers. In (Long et al., 2013), Hui Long, suggests LABERIO (LoAd-BalancEd Routing wIth OpenFlow). LABERIO is designed to find alternative links for the busy ones. Part of our work will depend on the LABERIO idea.

2. Simulation software, Models and Methods used

To make a simpler simulation, the Internet Protocol address (IP address) has not been used in our simulation, however it can be added later on in further work and this does not contradict with the concept of our work.

2.1. Simulation software

Our project consists of a very general network, and can be implemented with Ethernet, fibre or wireless connections.

OMNeT++ simulation is the network modelling software that is used to build our project. OMNeT++ is an object oriented simulation that depends on both C++ and Network Description (NED) languages.

2.2. Models of the simulation network

The project consists of two networks, and each network has 16 Routers, 6 Hosts and multiple data rate channels connecting Routers and Hosts with each other. The first network is as shown in figure1.

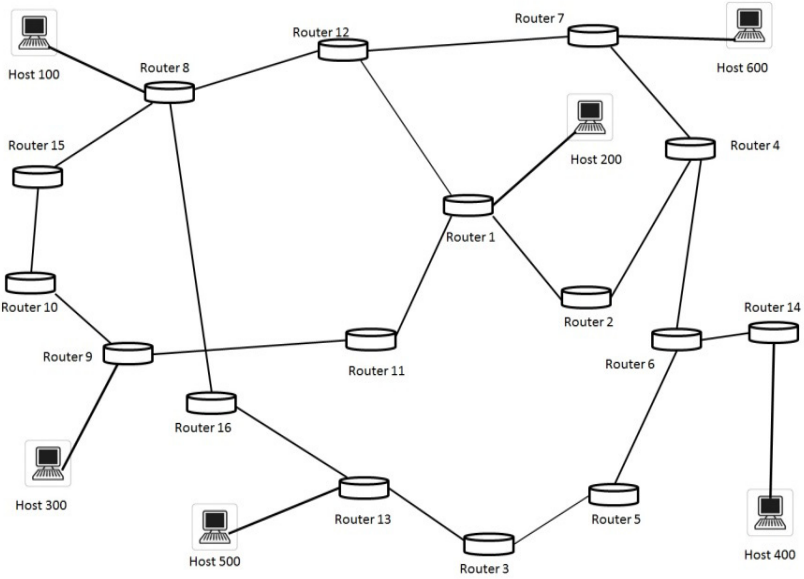

Figure1: Network without controller (the first simulation)

The second network, as shown in figure 2, is similar to the first network. The only difference is that there is a Controller device connected to each Router through very small bandwidth channels.

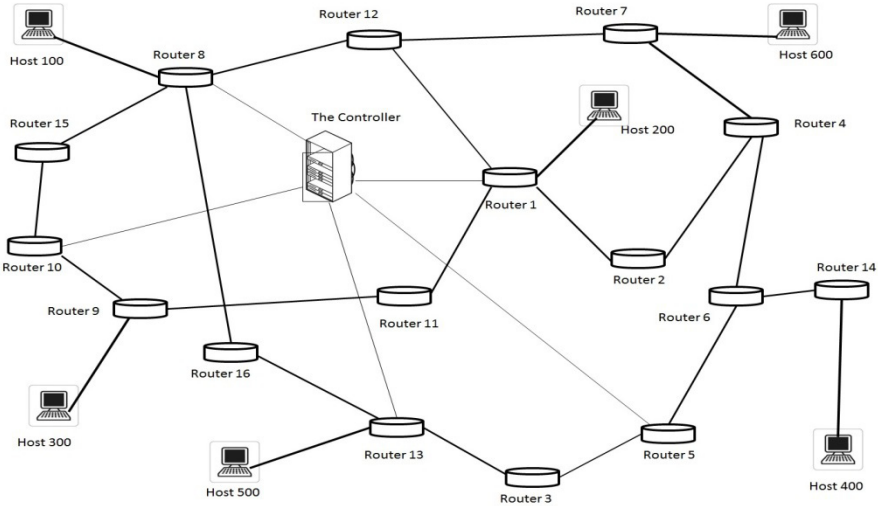

Figure2: Network with controller (the second simulation)

2.3. Methods used

EIGRP routing protocol algorithm has been built inside each router.

On the start of the simulation, all the devices identify themselves to their neighbours and the amounts of bandwidth of each of the connected channels (links) are recorded inside each router.

In the second simulation (figure 2), in addition to what has previously been mentioned, all Routers identify themselves to the Controller and the amounts of bandwidth of all the channels that connect routers are recorded in the Controller as well.

After neighbours' identification operation, the routers start exchanging information with each other and sending update packets to calculate the least *Metric* values to each destination (Host) through equation (1).

Routing loops problem may occur, and to overcome this problem *split horizon* is the best solution. In split horizon, the routing updates are not sent to the destination from which they came (Lammle, 2007, p 382).

The information that is exchanged by routers comprises the bandwidth and the delay of the connecting channels. By comparing the *Metric* different values, both of Successor and Feasible Successors are allocated, and in this operation the maps of each discovered path (Successor and Feasible Successor) to all destinations are drawn.

Note: In our simulation, One Successor and one or two Feasible Successors have been allocated.

In the first part of simulation figure 1, after exchanging of information and comparison operations are completed, the routing tables inside each router are completely built and routers are ready for routing. Hosts start sending Data flow packets to each other across the network and the routers connected to the hosts start receiving those data flow packets as the first step to forward them later to their destinations.

In our simulation, the routing table inside each router is built by using C++ map container method.

In the first simulation (figure 1) and after receiving data flow packets, each router starts the look up operation inside its routing table, then forwards each data flow packet through a specific out port. This operation is repeated in every router where the data flow packet has arrived, until those data flow packets reach their last destination.

Sometimes it is necessary for the router to send much more packets than the allowed rate through one port, because this amount of information is larger than the availability of the connected channel to bear. The Queue of the router's port will drop the extra packets which results in loss of information. This problem can be solved through smart load balancing.

In the second simulation (figure 2), after the exchange of update information and comparison operation is completed, routing tables inside each router are completely built. Then all routers send all the information they have (routing tables of Successor and Feasible Successor paths and their maps) to the Controller to be recorded there and then be dealt with later.

As a result of this mentioned operation, the controller obtains full knowledge about the topology and the components of the network as well as the successor and feasible successor routes from all routers to all destinations and their detailed paths, hops they pass through and bandwidths.

Through this knowledge, the controller will be able to manage the traffic across the network more efficiently by applying smart load balancing among routers and through channels.

3. Practical Work

3.1. Flows applied

According to our simulation, the traffic flows applied on the network will be as shown below subsequently:

- Host 100 start sending data flow of size equal to 30Mbps to Host 400.
- Host 300 start sending data flow of size equal to 30Mbps to Host 500.
- Host 200 start sending data flow of size equal to 30Mbps to Host 600.
- Then after a small period of time, two other flows will be added:
- Host 100 start sending data flow of size equal to 30Mbps to Host 300.
- Host 500 start sending data flow of size equal to 30Mbps to Host 100.
- Each of Host 100, Host 200, Host 300 and Host 500 sends 30 packets.
- Host 400 and Host 600 do not send any packets.

The map of path (route map) from each router (for successor path and first and second feasible successor paths) to all destination hosts has been drawn and stored as two dimensional arrays in every router as well as inside the controller as shown in table 1.

For example, the map of paths (successor paths) that is calculated in router 1 to all the destination hosts is as shown in Table 1 below.

Destination Host	Hop1	Hop2	Hop3	Hop4
200	0			
500	3	13		
100	11	8		
600	11	8	12	7
300	11	9		
400	2	4	6	14

Table 1: Successor map paths of router 1

table 計算 hop 有錯誤

The map of paths (first feasible successor paths) that calculated in router 1 to all the destination hosts is as shown in Table 2 below.

Destination Host	Hop1	Hop2	Hop3	Hop4
500	13			
100	13	16	8	
600	2	4	7	
300	12	8	11	9
400	3	5	6	14

Table 2: First feasible successor map paths of router 1

For Table1, *0* means that the destination host is directly connected to the router and such destination has only a Successor (no feasible successors). The numbers below the Hop column represent the numbers of the routers along the discovered path. In some routers, it can be noticed that the first (or the second) feasible successor paths have not been discovered in our simulation, which means that there is only the successor path that leads to those destinations.

In the first simulation, where there is no controller (figure 1), each router uses the EIGRP traditional routing table to forward packets on Successor routes through the appropriate port to the next hop. In some of the hops, congestion on links occurs and several packets are dropped due to this congestion. This drop of packets has been managed by a queue put on each output port of each router. The queue is programmed to drop extra packets if it contains more than 5 packets at any moment and the data rate of the flow is more than the band width of the connected channel.

Forwarding operation is repeated on each hop (router) until packet reaches to its destination or dropped on the way because of congestion.

In the second simulation, where the controller exists (figure 2), when the routers connected to the hosts receive flow packets from hosts, they obtain the flow definition from the header of the first packet of the flow that they receive and then they create router to controller packet (flow request packet) and send it to the controller. This packet contains the details of the flow (source, destination, data rate and priority), the controller examines the situation of the links (the available and the used bandwidths as well as the data rate of the links of all paths that may be used by the flow) then it takes one of the following decisions:

- For the successor path, if there is no congestion across the network then the decision of the controller is ordering the router (where the flow request packet came from) to send the flow through the successor path.
- If there is a congestion that may occur (at any link) because of the flow across the network, then the first step is to process the new flow (or the existing changed date rate flow) with the other existing flows through the routing algorithm.

3.2. The routing algorithm

Inside the controller, the following tests are applied on the new started flow to find which test results in no congestion.

- Send the flow through the first feasible successor route (**Controller decision = 0**).
- Send the flow through the second feasible successor route (**Controller decision = 1**).
- Drop the Flow (**Controller decision = 2**).
- Send the flow through the third feasible successor route (**Controller decision = 3**).
- Distribute the flow equally (1:1 rate) or not equally (1:2 rate) among the successor and the first feasible successor routes (**Controller decision = 4**).
- Distribute the flow not equally (1:2 rate) among the successor and the first feasible successor routes (**Controller decision = 5**).
- Distribute the flow not equally (2:1 rate) among the successor and the first feasible successor routes (**Controller decision = 6**).

After the above tests, the test that gives no congestion will be the decision of the controller on how to deal with the flow. The flow **Bit Rate** represents the maximum constant number of bits sent by source host and inserted inside flow data packets.

The controller's decision is represented by a controller to router response packet (decision packet) and this packet will be sent to the router where the flow request packet came from (first router that flow packets start routing). The router receives the decision packet from the controller, matches the decision with its flow table and forwards (redirect) the next flow data packets according to the controller decision through the appropriate gates to their destinations.

The decisions of the controller in our second simulation (figure 2) will result in flows that will be redistributed across the network as shown in Table 3:

Flow				Controller Decision
Flow No.	Source Host	Destination Host	Bit Rate	
1	100	400	30Mbps	0
2	300	500	30Mbps	0
3	200	600	26Mbps	4
4	100	300	30Mbps	1
5	500	100	30Mbps	1

Table 3: Decisions of the controller for each started Flow

Then the bit rate of some of the flows will change gradually after some time(values either increase or decrease), and the decision of the controller for each change of flow bit rate may change or not change according to the situation that gives the best result as shown below:

The first change of bit rate will occur in flow No. 1 & flow No. 2

Flow			Controller		
Flow No.	**Old Bit Rate**	**New Bit Rate**	**Old Decision**	**New Decision**	**changed**
1	30Mbps	25Mbps	0	0	No
2	30Mbps	32Mbps	0	0	No

Table 4: Decisions of the controller for change of bit rate of flow No. 1 & flow No. 2

Then another flow bit rate changes

Flow			Controller		
Flow No.	**Old Bit Rate**	**New Bit Rate**	**Old Decision**	**New Decision**	**changed**
5	30Mbps	32Mbps	1	6	Yes
4	30Mbps	10 Mbps	1	1	No

Table 5: Decisions of the controller for change of bit rate of flow No. 5

Then bit rate of other flows change

Flow			Controller		
Flow No.	**Old Bit Rate**	**New Bit Rate**	**Old Decision**	**New Decision**	**changed**
3	26Mbps	28Mbps	4	4	No
5	32Mbps	34Mbps	6	6	No

Table 6: Decisions of the controller for change of bit rate of flow No. 3 & flow No. 5

Then bit rate of other flows change

Flow			Controller		
Flow No.	**Old Bit Rate**	**New Bit Rate**	**Old Decision**	**New Decision**	**changed**
2	32 Mbps	34 Mbps	0	0	No
1	25 Mbps	40 Mbps	0	5	Yes

Table 7: Decisions of the controller for change of bit rate of flow No. 2 & flow No. 1

Then bit rate of other flows change

Flow			Controller		
Flow No.	**Old Bit Rate**	**New Bit Rate**	**Old Decision**	**New Decision**	**changed**
3	28Mbps	30Mbps	4	4	No
2	34Mbps	35Mbps	5	5	No

Table 8: Decisions of the controller for change of bit rate of flow No. 3 & flow No. 2

Then bit rate of other flows change

Flow			Controller		
Flow No.	**Old Bit rate**	**New Bit Rate**	**Old Decision**	**New Decision**	**changed**
3	30Mbps	34Mbps	4	6	Yes

Table 9: Decisions of the controller for change of bit rate of flow No. 3

It can be noticed that in some cases the controller sends new decisions while in other cases the decision of the controller is the same as the previous decision, this depends on the availability of band width across the links and the conditions of the network at the time of decision.

Note each router sends regular flows updates to the controller every 30 seconds and the controller will modify the distribution of the load according to these updates.

3.3. Results

After running both of our simulations, the results are shown below:

- The number of packets that are dropped in the second network (Figure 2 with controller) during the simulation are 0 packets (there is neither congestion and nor packets' drop), while the number of packets that are dropped in the first simulation network (Figure 1 without controller) during simulation are 3 packets.
- The results of both first and second simulations are represented in figures 3 & 4 respectively. For both of Figure 3 & Figure 4, the X axis represents the distribution of the routers over the network while the Y axis represents the number of data flow packets that passed through each router during simulation time. However, in the first network, Figure 3, the load distribution is less efficient, since there are only 11 routers that forward packets and 5 disabled routers (router2, router5, router10, router15 and router16 are disabled), while in the second network, Figure 4, all routers in the network are used to forward packets which means that there is better flow of packets across the network (better use of bandwidth and network resources).

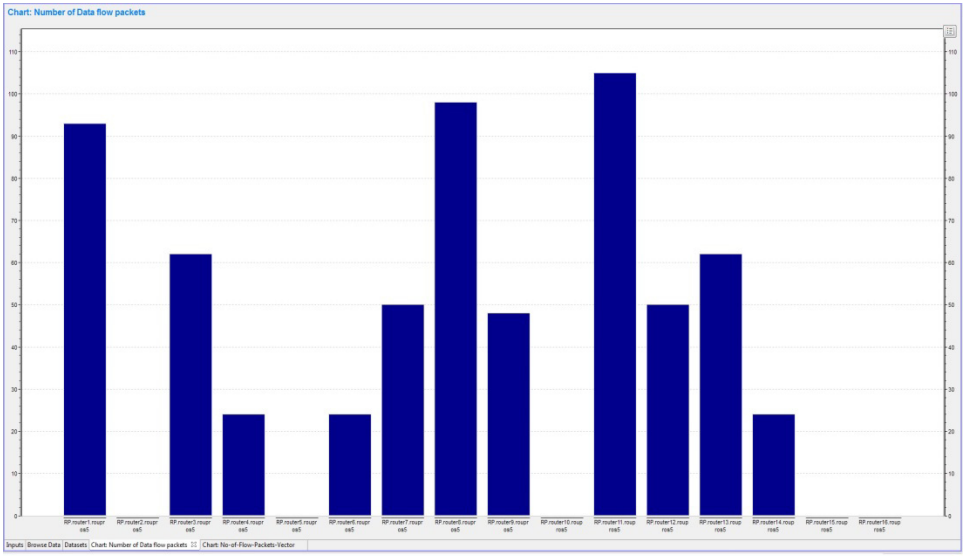

Figure 3: Load distribution (Data flow packets distribution) among routers in the first simulation without using controller

It also can be noticed in Figure 3 that the maximum load in some routers is much higher than that of their counterparts in Figure 4 which means that the load has not evenly distributed in the first simulation (without controller) as represented in Figure 3.

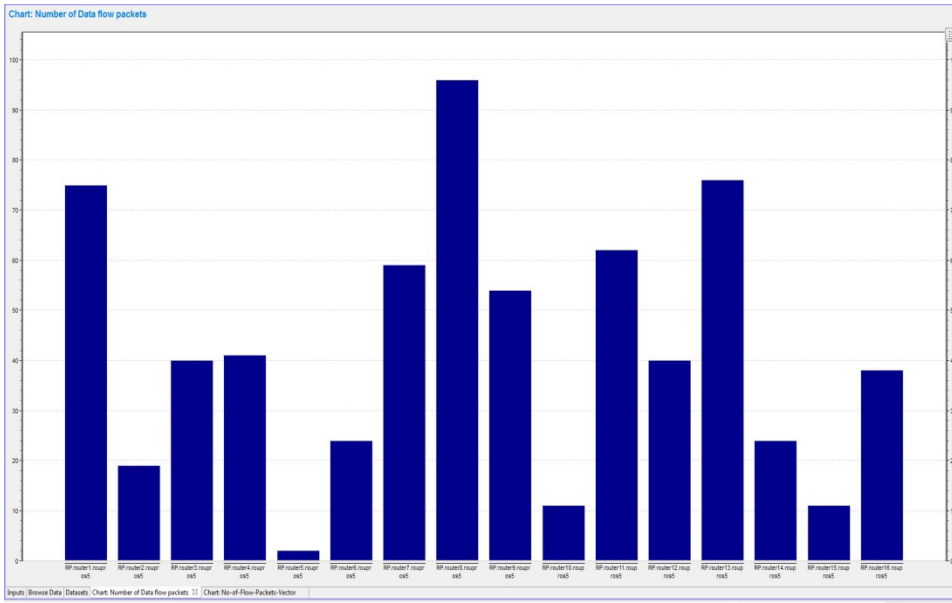

Figure 4: Load distribution (Data flow packets distribution) among routers in the second simulation by using controller (smart load balance)

4. Conclusions & Future work

The performance of the network has been improved by using more of the available network resources and better load balance.

The duty of the controller now is to manage the traffic according to the routing tables that were discovered by the routers and the available bandwidth of the links.

If the available routing tables are not sufficient to fulfil the requirements of the flows, then the controller will take the responsibility to find alternative temporary path to the destinations, creating temporary flow tables and the new discovered paths may be called the **Temporary Successors.**

5. References

CISCO web site (2013), 'Enhanced Interior Gateway Routing Protocol'. Available at: http://www.cisco.com/en/US/tech/tk365/technologies_white_paper09186a0080094cb7.shtml, (Accessed: December 2013).

Klein, D., and Jarschel, M., (2013) '*An OpenFlow Extension for the OMNeT++ INET Framework*' Available at: http://www3.informatik.uni-wuerzburg.de/research/ngn/ofomnet/paper-acm_with_font.pdf (Accessed: January 2014).
Lammle, T. (2007) *CCNA Cisco Certified Network Associate Study Guide*. 6th edn. ISBN: 978-0-470-11008-9/ Wiley Publishing.

Long, H., Shen, Y., Guo, M., and Tang, F., (2013) '*LABERIO: Dynamic load-balanced routing in OpenFlow-enabled networks',* IEEE 27th International Conference on Advanced Information Networking and Applications, pp. 290 - 297.

McKeown, N., Anderson, T., Balakrishnan , H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J.,(2008) '*OpenFlow: Enabling Innovation in Campus Networks*', OpenFlow based Publications.

OpenFlow org Web Site (2013) Available at: http://archive.openflow.org/wp/learnmore/, (Accessed: December 2013).