

Improvement of Performance of EIGRP Network by Using a Supervisory Controller with Smart Congestion Avoidance Algorithm

Hasanein Hasan

Electronic and Computer Engineering
Brunel University London
London, United Kingdom
eepghah@brunel.ac.uk

John Cosmas

Electronic and Computer Engineering
Brunel University London
London, United Kingdom
John.cosmas@brunel.ac.uk

Zaharias Zaharis

Electrical and Computer Engineering
Aristotle University of Thessaloniki
Thessaloniki, Greece
zaharis@auth.gr

Pavlos Lazaridis

Engineering and Technology
University of Huddersfield
Huddersfield, United Kingdom
p.lazaridis@hud.ac.uk

Sinan Khwandah

Maritime, Science and Engineering
Southampton Solent University
Southampton, United Kingdom
sinan.khwandah@solent.ac.uk

Abstract—The aim of this research is to enhance the performance of a traditional Enhanced Interior Gateway Routing Protocol (EIGRP) network by adding some features of the Software Defined Networking (SDN) in a modified approach to improve some network's performance metrics like link congestion and packet loss. The improvement of those metrics leads to an improvement in the total throughput of the network. This approach results in a network that contains an intelligent dynamic supervisory corrector controller, which is capable of detecting the location of a congestion that may take place before or at the brink of its occurrence and dealing with selected flows on selected routers across the network in a way that prevents the congestion by using a smart heuristic congestion avoidance and routing algorithm thereby reducing the generation of control messages as much as possible.

Keywords — Computer networks; Routing protocols; EIGRP; SDN; Openflow; Flow Distribution; Traffic control; Congestion, Reliability

I. INTRODUCTION

IN computer networks, the router is considered to be an active element that has the responsibility of forwarding data packets between computer networks. Each router forwards packets according to its routing table. The router use route updating packets to build and maintain routing tables. Protocols that send route update packets are called routing protocols [1]. However, the routing performance could be degraded due to congestion which increases delay and packet loss [2]. Network congestion is considered as one of the major reasons of quality degradation; it can be caused by uneven distribution of traffic or lack of network resources. Pre- or very early detection of the *occurrence* and the *location* of congestion provide a solution for better load balance by preventing congestion events and thus reducing congestion effects as much as possible.

A Software Defined Networking (SDN) OpenFlow example provides a centralized decision of packets forwarding. The controller is connected to all switches. The controller's

responsibility is to inform the switches how to forward the traffic [3]. This central controller can see the network and all its circulating flows thoroughly, giving global and optimal management of network. The OpenFlow switches are simple and dumb since the forward decisions are defined by the controller [4], [5]. For every new flow (i.e., flow with header that does not match any entry value in the existing flow table), the switch is required to ask the controller on how to distribute the flow and so the controller has to make a decision (an improved flow table) [3]. This means that the switches will fully depend on controller. This provides central flows management and load balance where the congestion can be avoided. Therefore the openflow has been proposed to be used in load balancing of data centres [6].

Using the openflow protocol in practical routing networks suffers from several difficulties. Connecting the controller directly to all Routers inside network is not feasible. Positioning the controller at a specific location inside the network and connecting it to a limited number of routers detaches it far away from others, especially if the size of the network is huge. The updating messages directed from the remote routers to the controller will subsequently suffer from *processing delay*, *queueing delay*, *transmission delay* and *propagation delay* [7] during the hops they traverse until reaching the controller and correspondingly for the control messages issued by the controller. Providing secure channel between the controller and all routers is also not feasible. There is a probability of packet loss and re-transmission. In addition, if the network is huge and many flows are traversing it, the controller as a unique decision maker will be loaded with a lot of mathematical operations to perform and many control messages to send which also contribute to an extra delay.

Our project combines the practical side of the traditional routing protocols with the intelligence of the SDN. The network of our project uses EIGRP routing protocol to build routing tables and forward packets. It also uses the SDN notion to enhance the EIGRP performance by providing load balance

(if necessary). The routers are independent on the controller in building their routing tables. However they may receive commands from the controller (temporary flow table) to change their forwarding decision related to specific flows to avoid congestion.

A real time dynamic monitoring controller with supervision and intervention mechanisms has been developed, which reduces the number of control messages across the network and the dependency of routers upon the controller. The controller examines the flows distribution over the network and only intrudes to deal with congestion. It is provided with congestion avoidance heuristic algorithm. This algorithm consists of three sub-routing algorithms working sequentially to deal with any congestion problem by providing better load balance.

II. RELATED WORK

Load balancing and congestion avoidance have been studied in SDN and conventional routing networks, especially the Open Shortest Path First (OSPF) routing protocol. As the publications deal with the development efforts of EIGRP are very few and limited, the comparing of our scheme is mainly with OSPF development efforts.

As far as the traditional routing protocol network is concerned, two kinds of congestions occur namely: the local congestion and the remote congestion [2]. The EIGRP supports unequal cost path load balancing among the feasible successor paths depending on the metric value of the path [8]. This load balance operation has been applied without referring to the remote congestion. Furthermore, it may not be necessary to split the flow if it has very low data rate and the successor path fulfills the forwarding requirements. If a flow is split somewhere and recombined at another place, the sequence order of packets will change. Thus, our algorithm avoids flow split as much as possible and takes in consideration the remote congestion problem. In EIGRP, load balance cannot be applied over a path if the path is not a feasible successor [8]. Our algorithm may use any path to avoid congestion whether the path is feasible successor or not.

In [2], Al-Shabibi et al. suggests a congestion aware multi-path routing protocol in a multi-route network, where there is no central controller to manage the traffic and the amount of traffic assigned to each path is determined by each router in response to the congestion signals received from routers along the path. Every router sends its updated information to the neighbouring routers when there is any significant change detected. Although the routers are fully independent, the disadvantage of this approach is that there may be a lot of information updating messages issued by routers, which may not only consume a significant portion of the available bandwidth but also consume a lot of processing and network capacity through re-transmission with possibility of packet loss, which increases the network delay.

In [9], Oki et al. proposes the Smart Open Shortest Path First (S-OSPF) load balance algorithm based on a model where the traffic is split only at source edge nodes and distributed to

the neighbor nodes with optimum ratios. From the neighbor nodes, the traffic is routed according to the OSPF protocol. The optimal traffic distribution is obtained through Linear Programming (LP) by using decision variables like network congestion ratio and traffic portion from parameters of traffic demand and link capacity. The traffic is not split over multiple paths. From our point of view, splitting the traffic only from the source edge node may prevent using of some intermediate links which are not located on the OSPF main paths of the flows from being used. Our proposed algorithm has the ability to split or divert traffic from any node (router) depending on the instantaneous flow condition of the network; whether the node is located at the edge of the network or inside it.

In SDN, several methods were proposed to develop performance [4] and load balance [10], [11], [12]. In [4], Fernandez et al. proposes the use of a proactive controller that pre-populates flow tables to the forwarding devices, which solves the processing time problem.

In the case of the LLoad-Balancing over UnStructured networks (LOBUS) algorithm [10], Handigol et al. considers a load-balancer, called Plug-n-Serve. Plug-n-Serve has the ability to balance the load over arbitrary unstructured networks, and tries to reduce the average response time. In [11], Long et al. suggests the application of the LoAd-BalancEd Routing with OpenFlow (LABERIO) algorithm, which deals with congestion and selects a path according to the available bandwidth; a NOX controller is used in LABERIO. In [12], Koerner slices the network resources and uses a different controller for every slice by creating a Flow Visor application to deal with different services. In [13], Yu et al. makes use of the controller to deal with the link failure incidences in OSPF networks.

The SDN-Based Equal Cost Multi Path (ECMP) distribution algorithm in data centers is proposed by Zang et al [14]. The OpenFlow protocol is used to optimize the ECMP algorithm to provide dynamic adjustment to the flow forwarding operation depending on the bandwidth utilization of core links. If the bandwidth utilization of a core switch exceeds specific threshold, the controller will forward some flows to other links which have lower bandwidth utilization [14]. The SDN-based ECMP distribution algorithm is restricted to deal with network of FatTree topology where the costs of the paths are equal and the change in flow path is applied on the core switches only. The latency in communication between the routers and the controller has been discussed in [15] where Phemius et al experiences its effect on TCP and UDP packets by using a network floodlight controller. Two solutions are suggested here to be applied together: double the contact with the controller and increase the buffer size of the switch where packets are held. Even with those solutions, the latency in taking decisions still exists. Our project provides a solution that combines different ideas especially the idea of LABERIO algorithm [11].

III. NETWORK MODEL FEATURES

In this section, the designed network is presented. The network has been designed by using OMNeT++ simulator.

A. Network Model and Network Messages

Our network architecture utilizes traditional network EIGRP routing protocol which developed by adopting the concepts of SDN network controller. Fig. 1 represents this adoption.

Our project examines two networks. In the first one there are 16 routers and 6 hosts, while in the second there are 10 routers and 5 hosts, as shown in Figs. 3 and 6 respectively.

For each network, two simulations are applied and compared to test the effectiveness of the congestion avoidance algorithm; in the first simulation, the network is without controller (simple EIGRP) while in the second one, the network operates in the presence of a controller (developed EIGRP by SDN) where the congestion avoidance algorithm is applied. The same traffic has been assumed on both simulations per network to compare their performance.

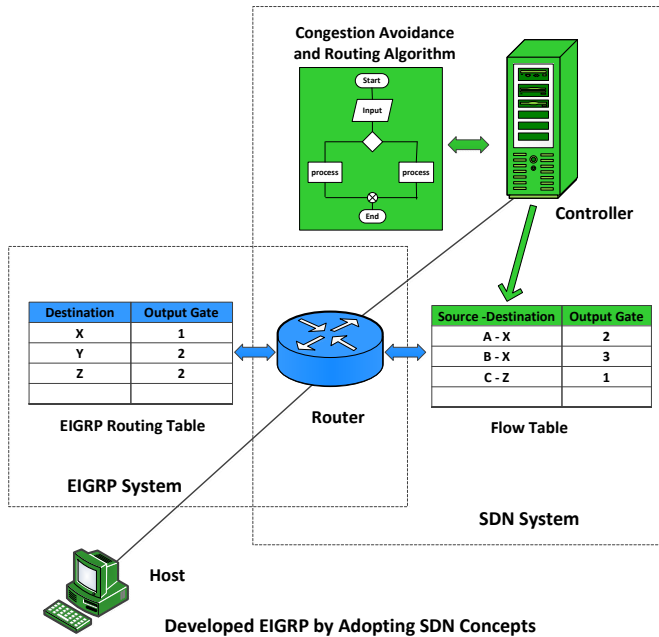


Fig. 1. Developed EIGRP by adopting SDN concepts

Before delving into the network operation details, there are different propose messages and packets travel across the network:

- EIGRP route update messages used to build the EIGRP routing table inside the router.
- Information messages issued by routers and directed to the controller. The Information messages carry the network information like router address, routing tables of each router and its links bandwidth.
- Flow status updating messages issued by head-end routers and directed to the controller. They carry flow current information like flow source, flow destination and flow data rate.
- Control messages (including approval messages) issued

by the controller and directed to specific routers.

- Regular updating messages issued by all routers every several minutes and forwarded to the controller.
- Acknowledge messages issued by controller and routers and to confirm the arrival of other messages.
- The data packets are issued only by hosts and directed to other hosts. Data packets represent the flows.

B. The Router

The router in our architecture has two tables, a *traditional EIGRP routing table*, which is built by the router itself through information acquired from other routers and a *flow table* that is received through control messages sent by the controller. The flow table is temporary and has higher priority than the EIGRP routing table. Some flows are forwarded according to the EIGRP routing table whereas other flows are forwarded according to the flow table or both tables. The EIGRP routing table may change according to the change in topology (link failure), while the flow table changes according to the decision of the controller. The routing table can deal with all flows to all destinations while the flow table deals with some flows directed to specific destination. The router forwarding architecture is shown in Fig. 1.

1) The EIGRP routing tables inside Routers

The multi-path discovery process depends on the EIGRP-based algorithm that designed and embedded inside each router. Each router must have at least one path to every destination. The EIGRP-based algorithm depends on both the available bandwidth and the delay according to the following expression of the default behavior:

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

In (1), *Bandwidth* is the least bandwidth (measured in kilobits/second) of all the outgoing links to the destination, while *Delay* represents the sum of the delays (measured in 10s of microseconds) configured on the interfaces, links, and hops on the path to the destination [16].

The EIGRP-based algorithm has been used because it can find several paths to the destinations. The route that has the least *Metric* value is called the *successor*, and is considered to be the best route to the destination (i.e., the successor has the highest bandwidth and the lowest delay). The neighbour routers that have an advertised metric less than the metric of the current routing table are called *feasible successors* [17]. Theoretically, for every destination there are only one successor and up to six feasible successors [1].

At the beginning of the simulation, all routers start exchanging hello messages with their neighbors. While the exchange of information proceeds, a comparing operation is performed by using (1), which results in the construction of the *EIGRP* routing tables inside the routers. In our simulation, to preserve the packet order and obtain the best performance, all the data packets are forwarded through the successor path. The controller takes the responsibility of using (if necessary) feasible successor paths or other paths by applying its congestion avoidance and load balancing algorithm.

2) The queue of the router

The queue is connected to the output gates of the router; the capacity of the queue is designed to have the same value for all the routers in the network. To make the connection between the routers and the controller faster and more reliable, scheduling mechanism of the queue is adjusted so that the sending priority of flow status updating messages and control messages is higher than the sending priority of data packets.

3) The flow counter

If the decision of the controller for a specific router is to forward a flow over more than one path according to a specific ratio, then the router will establish a flow counter to apply this distribution ratio among the gates.

C. The Controller

The controller is the device that has the responsibility to manage the traffic across the network and deal with any congestion problem. It can be connected only to some routers. For the routers that are not directly connected to the controller, the information, updating and control messages use the EIGRP routing table to find their way from routers to the controller and vice versa.

1) The controller behavior

The controller receives all the information related to the network, such as topology, channels bandwidth and EIGRP routing tables of all routers through the information messages sent by the routers. According to this information, the controller constructs the network topology and it can predict the behaviour of the routers (the forwarding decision related to data packets) when they deal with the applied flows. It can be said that the mathematical and comparison operations inside the controller represent a replica of the network and the decision is the most complicated operation of the controller.

The controller receives the flow status updating messages from the head-end routers and analyses them according to the behavior of the network against the flow to see if there is any possible congestion. When the controller predicts congestion on one link or more, it uses its congestion avoidance algorithm to solve the problem then intrudes to change the behaviour of selected routers by changing their forwarding decisions related to selected flows to prevent the congestion. These flows will be distributed or re-distributed temporarily according to the controller's decision which is represented in the router by the received flow table. The amount of the data rate sent on each specific router's gate is an accurate value defined by the controller. The intervention of the controller in our architecture is very limited with very few control messages required to be sent.

2) The congestion avoidance and routing algorithm

When the controller intrudes to solve a congestion problem, it applies the proposed algorithms, which consists of three sub-algorithms (processes); firstly, *the feasible successors flow distribution sub-algorithm*, then *the temporary successor sub-algorithm* and finally *the clearing way sub-algorithm*. These sub-algorithms are applied consecutively to deal with a specific flow and only on a high traffic situation, where the

probability of congestion is high as well. If the current sub-algorithm solves the congestion problem, then there is no necessity to apply the next sub-algorithm. If the first two processes of the algorithm applied on a certain flow cannot solve the congestion problem, the third process will repeat the first two processes on another flow and so on.

The congestion occurs when the summation of flows' rates applied on a link is bigger than the capacity of the link. The success of congestion avoidance algorithm or sub-algorithm depends on the topology, the available resources and the currently applied load. The structure of the controller congestion avoidance algorithm described above is shown in Fig. 2.

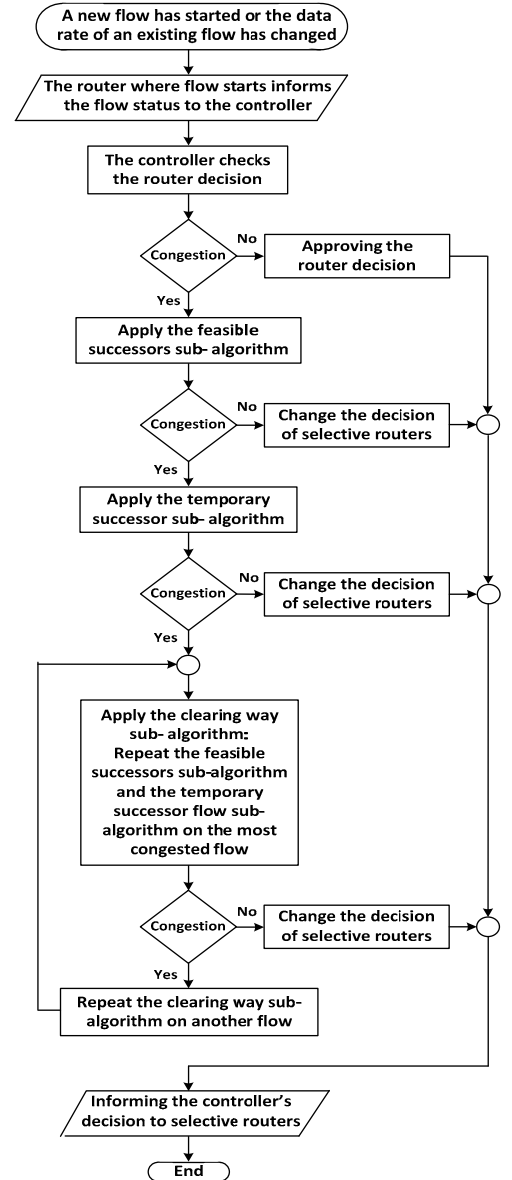


Fig. 2. The congestion avoidance algorithm inside the controller.

IV. SYSTEM PERFORMANCE

For every new flow (or obvious change in the data rate of an existing flow), the following events take place:

The head-end router (the router where a flow starts), will normally deal with the flow, i.e., it will forward the flow by default according to the EIGRP routing table through the appropriate gate and, at the same time, a flow updating message that contains the flow information will be sent to the controller.

When the controller receives the flow updating message from the router, it will start internal mathematical operations to model the network behaviour on dealing with the flow (the flow forwarding operation across the routers on the successor path) and notices if there is a congestion that may occur at any link inside the network due to this flow. If there is no possible congestion to occur at any link over the path where the flow is supposed to pass, the controller sends the head-end router an approval control message. The head-end router and the other routers over the path continue forwarding flow packets according to their EIGRP routing table.

For example, in network 1 of Fig. 3, host B starts sending a low data rate flow to host C, so that the flow B-C will pass through the successor path across routers 1, 11 and 9. The head-end router (router 1) informs the controller about the flow status by sending flow status updating messages. The controller internally tests the network behaviour against the flow; the result will be no congestion at any link if the flow is forwarded over the successor path. According to this result, the controller simply sends an approval message to router 1. Routers 1, 11 and 9 continue forwarding the flow B-C according to their EIGRP routing tables and over the successor path. The flow B-C is transmitted successfully from its sender (host B) to its final destination (host C) as shown in Fig. 3.

If congestion occurs at any link/links due to this flow, the controller applies its *congestion avoidance algorithm* to deal with the network flows. The first step of the congestion avoidance algorithm is to utilize the feasible successor flow distribution algorithm.

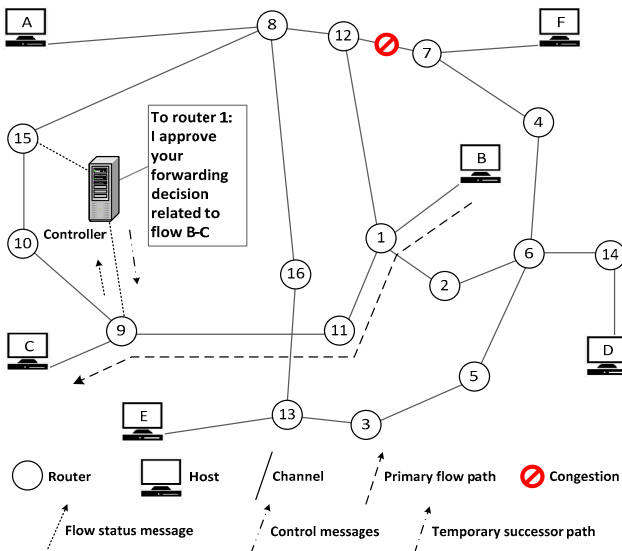


Fig. 3. Normal traffic distribution with no congestion in network 1.

1) The feasible successors flow distribution sub-algorithm

1. Allocate the links, where the congestion occurs or is

supposed to occur, mark them with a no entry symbol represented by and estimate the amount of the excessive data rate that causes the congestion.

2. Test the links through all the feasible successor paths and record the available bandwidth on each feasible successor path.
3. If the available bandwidth of the first feasible successor path is greater than or equal to the extra amount of flow data rate (that causes the congestion), then forward the extra flow data rate through this path.
4. If the available bandwidth of the first feasible successor path is not enough, then repeat step 3 on the second feasible successor path and so on until finding the feasible successor path that can handle the extra data rate.

For example, suppose that the data rate of the flow B-C of Fig. 3 has been increased. Router 1 will inform the controller about the situation of flow B-C by sending flow status updating message. The controller starts an internal testing operation to the flow current forwarding path, which results in a congestion on link 1-11 (similar to the network's real situation).

To solve the congestion problem, the controller activates its congestion avoidance algorithm. The first step of the congestion avoidance algorithm is searching among the available feasible successor paths to find another path to forward the excessive flow data rate that causes the congestion. The controller decides to divert the extra flow data rate to the first feasible successor through routers 1, 12, 8, 15, 10 and 9.

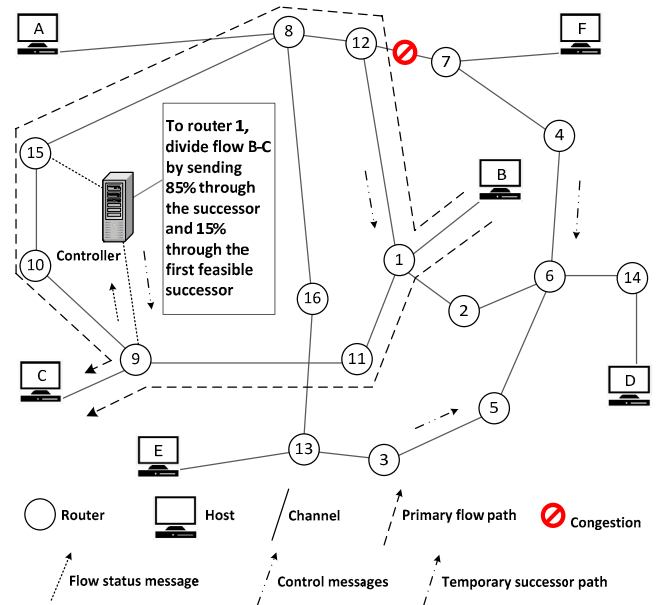


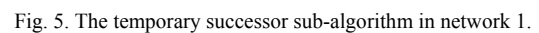
Fig. 4. The feasible successors flow distribution sub-algorithm in network 1.

The controller inform router 1 to divide flow B-C between the successor and the first feasible successor paths through its control message. The flow B-C will be implemented through both the successor and the first feasible successor paths in specific ratio imposed by the controller so that no congestion is going to take place at any link. The flow B-C is distributed as shown in Fig. 4.

2) The temporary successor sub-algorithm

- For example, suppose that in the network 1 depicted in Fig. 4, where both of the successor (1, 11, and 9) and the first feasible successor (1, 12, 8, 15, 10, and 9) paths cannot handle the high data rate of flow B-C that caused congestion on links 1-11 and 12-8. The controller will apply the temporary successor sub-algorithm and the discovered temporary successor path will be across routers 1, 2, 6, 5, 3, 13, 16, 8, 15, 10, and 9 as shown in Fig. 5. The extra flow data rate will be forwarded through this path. The flow B-C will be divided

It is obvious that to the head-end router (router 1), the congestion at link 1-11 represents a local congestion while the congestion at link 12-8 represents a remote congestion.



3) The clearing way sub-algorithm

1. Allocate the links where the congestion is about to occur, mark them with a no entry symbol , and then allocate all the flows that pass through the congested links.
2. Find a flow that is mostly repeated on those congested links.
3. Extract the amount of the flow data rate that is equal to the excessive data rate that causes the congestion.
4. Repeat using the successor flow distribution sub-algorithm and the temporary successor sub-algorithm to find an alternative path to this selected flow.
5. If there is no alternative path to the flow that is mostly repeated on the congested links, then repeat the same steps on the secondly most repeated flow on the congested links and so on until terminating the congestion or reducing its effects as much as possible.

- Some flows will be forwarded according to the routing tables, some other flows will be forwarded according to the flow tables and finally some others will be forwarded according to both tables.

For example, suppose that in the network 2 depicted in Fig. 6, host A sends a flow to host E through the main path 1, 2, 3, 4 and 5, host B sends another flow to host C through path 2, 3 and host C sends a third flow to host D through path 3, 4.

Both flows B-C and C-D increase their data rate until they occupy almost all the available bandwidth of links 2-3 and 3-4 respectively. The most congested flow now is flow A-E because it is the flow mostly repeated on the congested links.

The controller applies the feasible successors flow distribution and the temporary successor sub-algorithms to find an alternative path to flow A-E. The new discovered path passes through hops 1, 6, 7, 8, 9, 10 and 5. Flow A-E is fully forwarded over the new bypass path as shown in Fig. 7.

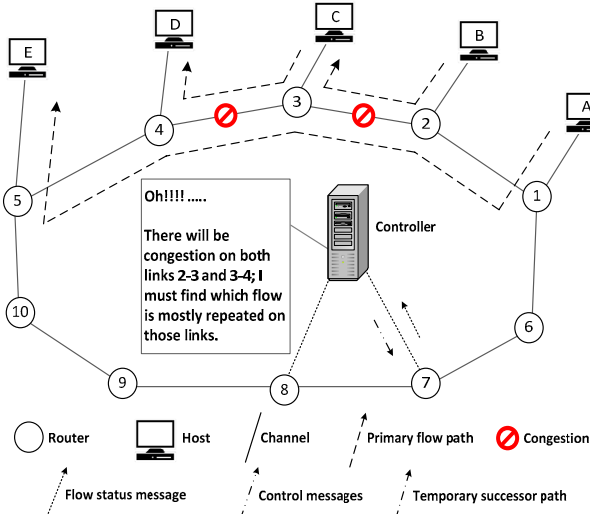


Fig. 6. Traffic distribution with congestion at two successive links in network 2.

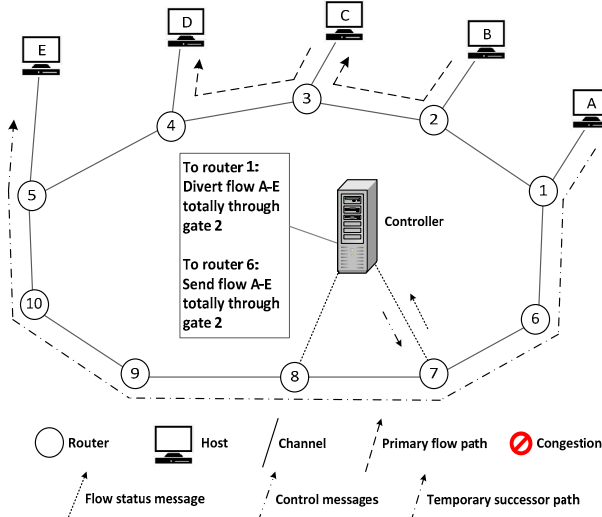


Fig. 7. The application of the clearing way sub-algorithm in network 2, where flow A-E has been completely diverted to an alternative path.

The new path represents a temporary successor path. It can

be noticed that even when a data rate increase occurred in flows B-C and C-D, the path that changed is the path of flow A-E. During the simulation, the routers send regular updating messages to the controller; those updating messages inform the controller of the current status of each router, which helps to detect if there is any control message dropped by mistake and therefore to correct the situation.

V. RESULTS

Both networks 1 and 2, shown respectively in Figs. 3 and 6, have been tested on 2 simulations. In the first simulation there is no controller (EIGRP), while in the second one there is a supervisory controller (Developed EIGRP by SDN/ openflow). The resulted throughput of networks 1 and 2 for both simulations (with and without controller) are shown respectively in Figs. 8 and 9.

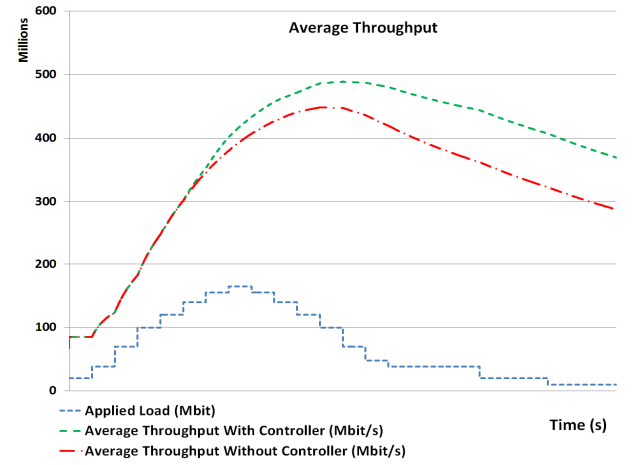


Fig. 8. Throughput comparison between two simulations of network 1 (with and without controller) during the same applied load.

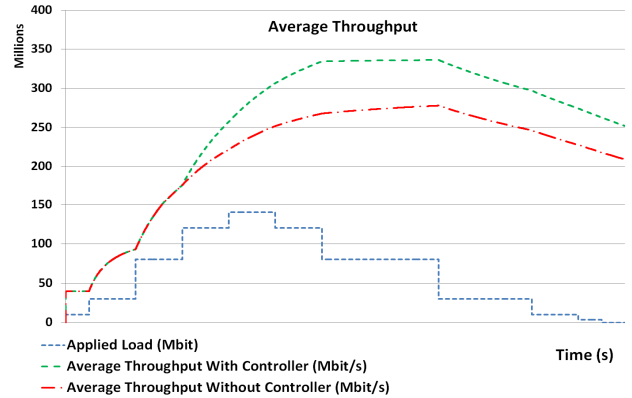


Fig. 9. Throughput comparison between two simulations of network 2 (with and without controller) during the same applied load.

For Figs. 8 and 9., it seems that for low traffic, the throughput values are almost the same either by using a controller or not. By increasing the applied traffic in both simulations in both networks, the controller congestion avoidance algorithm starts working inside the controller and according to its behavior; some flows will be diverted or divided over paths to avoid congestion. Therefore, the throughput curve will obviously increase, while in the case

where there is no controller, the throughput curve will slightly increase. The higher the throughput the lesser the packet drop. As the applied load increases, the controller effect becomes more obvious. After traffic returns to its normal rate, both curves start to become parallel. This means that the dealing with traffic is now similar for both simulations per network.

The number of control messages issued by the controller is limited. To re-distribute a flow according to the feasible successors flow distribution sub-algorithm, the controller contacts the head-end router only. The controller only contacts router 1 to divide flow B-C between the successor and the feasible successor paths as shown before in Fig. 4.

To re-distribute a flow according to the temporary successor sub-algorithm, the controller contacts the routers which their EIGRP routing table differs from the decision of the controller to build a flow table inside them. Also it may contact the head-end router or any router on the EIGRP path to split the flow. The controller contacts routers 1, 2, 6, 5 and 3 to re-distribute flow B-C among the successor, the feasible successor and the temporary successor paths as shown in Fig. 5.

VI. CONCLUSION

The development of the EIGRP routing protocol by using the concepts of the SDN provides a network with better load balance, less packet drop and higher throughput. Using a controller supplies full monitoring and semi central management to the network. The performance improvement induced by the controller depends on the topology and the available resources, such as the number of nodes and the channel bandwidth. The limited number of the control messages issued by the controller saves bandwidth and gives the routers more autonomy. Under low traffic conditions, the network behaves as a traditional EIGRP network. The controller works on higher applied load to avoid congestion. Under very high traffic conditions, the applied algorithm inside the controller fails to protect all packets from being dropped. However, a network that utilizes the proposed controller still performs better than a network that has no controller support.

REFERENCES

- [1] T. Lammle, "Cisco Certified Network Associate Study Guide," 6th ed., ed. Indiana: Wiley Publishing, 2007, pp. 22, 420–421.
- [2] A. Al-Shabibi and B. Martin, "MultiRoute - a congestion-aware multipath routing protocol," High Performance Switching and Routing (HPSR), 2010 International Conference on , vol., no., pp.88,93, 13-16 June 2010
- [3] D. Klein and M. Jarschel (2013, March). An OpenFlow Extension for the OMNeT++ INET Framework. [Online]. Available: http://www3.informatik.uniuerzburg.de/research/ngn/ofomnet/paper-acm_with_font.pdf
- [4] M.P. Fernandez, "Comparing OpenFlow Controller Paradigms Scalability: Reactive and Proactive," *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on* , vol., no., pp.1009,1016, 25-28 March 2013
- [5] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks". [Online]. Available: <http://archive.openflow.org/documents/openflow-wp-latest.pdf>
- [6] P. Wang, J. Lan and Chen, "OpenFlow based flow slice load balancing," in *Communications, China* , vol.11, no.12, pp.72-82, Dec. 2014
- [7] D. Bertsekas and R. Gallager, "Data Networks," 2nd ed., A Pearson, Ed. New Jersey: Prentice-Hall, 1992, pp. 149-150.
- [8] Cisco Systems. How Does Unequal Cost Path Load Balancing (Variance) Work in IGRP and EIGRP?. Document ID: 13677. [Online]. Available: <http://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/13677-19.html>
- [9] E. Oki, and A. Iwaki, "Load-Balanced IP Routing Scheme Based on Shortest Paths in Hose Model," *Communications, IEEE Transactions on* , vol.58, no.7, pp.2088,2096, July 2010
- [10] N. Handigol, S. Seetharaman, M. Flajslik, N. McKeown, and R. Johari, "Plug-n-Serve: Load-balancing web traffic using OpenFlow," Demo at ACM SIGCOMM, Aug. 2009.
- [11] H. Long, Y. Shen, M. Guo and F. Tang, "LABERIO: Dynamic load-balanced Routing in OpenFlow-enabled Networks," *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on* , vol., no., pp.290,297, 25-28 March 2013
- [12] M. Koerner and O. Kao, "Multiple Service Load-Balancing with OpenFlow," In Proc. HPSR, 2012, pp. 210-214.
- [13] Y. Yu, C. Shanzhi, L. Xin and W. Yan, "A framework of using OpenFlow to handle transient link failure," *Transportation, Mechanical, and Electrical Engineering (TMEE), 2011 International Conference on* , vol., no., pp.2050,2053, 16-18 Dec. 2011
- [14] H. Zhang, X. Guo, J. Yan, B. Liu and Q. Shuai, "SDN-based ECMP algorithm for data center networks," *Computing, Communications and IT Applications Conference (ComComAp), 2014 IEEE* , vol., no., pp.13,18, 20-22 Oct. 2014
- [15] K. Phemius and M. Bouet, "OpenFlow: Why latency does matter," *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on* , vol., no., pp.680,683, 27-31 May 2013.
- [16] Cisco Systems. Enhanced Interior Gateway Routing Protocol. Document ID: 16406. [Online]. Available: <http://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/16406-eigrp-toc.html>
- [17] Cisco Systems. Introduction to EIGRP. Document ID: 13669. [Online]. Available: <http://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/13669-1.html>