An SDN-based congestion-aware routing algorithm over wireless mesh networks

Hao Fu[a], Yuan-an Liu, Kai-ming Liu and Yuan-yuan Fan
School of Electronic Engineering, Beijing University of Posts and Telecommunications,
Beijing, 100876, China
E-mail: [a]fuhao2015@bupt.edu.cn

The issue of routing is vitally important in Wireless Mesh Networks. But currently, most routing protocols such as OLSR and AODV cannot make the most of multiple paths between the source site and destination site because of the complexity and the cost. Software Defined Networking(SDN) structure promises to obtain the network configuration effectively, and with a centralized controller, it can deploy fine-grained routing algorithms to make full use of the network resources, while ensuring that the control overhead is acceptable. In this paper, we propose a new approach of SDN-based routing algorithm, or SDNR. We introduce link saturation to SDN, with which the SDN controller can figure out the congested path and reroute the following traffic to another non-congested path, which is the real-time optimal one, to ensure the network throughput. We compare SDNR to classic routing protocols and demonstrate its superiority.

Keywords: SDN; Routing; Link saturation; Congestion-aware.

1. Introduction

With the rapid growth of Wireless Mesh Networks, more and more communications are performing over WMNs. Therefore the demand on resources is also growing significantly. However, the resources utilization rate is far from efficient because of the constraints of routing protocols currently in WMNs. The main reason is that most routing protocols always find only the optimal path between source site and destination site, and route all the traffic onto the path, which means the traffic will still be pushed onto the path even if it becomes congested and this process will not stop until the valid time of the route table entry runs out. At the same time, the bandwidth resources on the other paths between the source site and destination site have not been fully used. On the other hand, with some additional improvements, OLSR [1] and AODV can figure out the multiple paths and balance the amount of traffic to these paths proportional to their costs [2] [3]. Still, to achieve this, the control overhead will increase severely and some extra transport layer protocol will be needed. Moreover, this solution is inefficient because suboptimal paths will always be used even if the optimal path can carry out all the traffic.

Software Defined Networking is a paradigm that separates control functions from the WMN routers. It focuses the control functions onto a centralized unit, called controller, while data forwarding functions remain within the WMN routers, which is called forwarders in SDN. OpenFlow is the protocol that controller uses to configure the forwarders, which contains the rules that controller uses to communicate with forwarders.

It is also the foundation of the SDN's flow-based routing architecture where the controller makes decisions of every flow in the WMN based on the network state information delivered by forwarders. Controller will update the rules if network state information changes and find the optimal routing strategy for each flow. The flow-based routing architecture is the reason why SDN is suitable for deploying of fine-grained routing algorithms to improve the network performance.

SDN architecture and OpenFlow have been proposed as better solutions for many scenarios. The works in [4] and [5] develop a scheme to enable the dynamic quality of service over OpenFlow networks to transmit video streaming. The author encoded the video into one base layer, which is indispensable in decoding, and several enhancement layers. With the flow-based routing architecture of OpenFlow, the controller finds the optimal route for the base layer, which ensures that the base layer suffers no packet loss and has minimum delay variation, and several suboptimal routes for the enhancement layers to save the cost. They have got better video quality and acceptable cost in contrast with the traditional network structure. The work in [6] defines a hybrid structure that combines OpenFlow with the traditional routing protocol. With this structure, they can take advantage of OpenFlow to increase the network throughput when the OpenFlow and SDN controller are running correctly, and they can still operate the network when it is suffering a controller failure. [7] points out that the challenge for deploying centralized control of OpenFlow over a decentralized WMN, and it gives a solution to this problem.

In this paper, we propose a SDN-based routing strategy to route the packets in WMNs. We introduce link saturation to SDN, the forwarders calculate its saturation and deliver it to the controller periodically, with which the controller can figure out the link congested situations and find the optimal paths for the following flows. Similar work has been done in [8], FAMTAR uses a similar forwarding table like OpenFlow, but it still uses the traditional ways to find the paths. We compare our scheme to both FAMTAR and a classic routing protocol OLSR and demonstrate its superiority.

The rest of this paper is organized as follows: The proposed architecture for deploying OpenFlow and SDN over WMNs is introduced in Section 2. Section 3 proposes our SDN-based routing algorithm. Simulation experiments and tests results are presented in Section 4. Finally, we conclude this paper in Section 5.

2. OpenFlow Over Wireless Mesh Networks

As mentioned in [6] and [7], the first problem to solve before deploying OpenFlow over WMNs is the control network scheme. Controller needs a secure channel to communicate with forwarders in SDN architecture. Two kinds of solutions have been proposed for this problem: in-band control network and out-of-band control network. The former one means that the control signal shares the same WMN with data traffic, while the latter one means that using two different WMNs to transmit control signal and data traffic respectively.

Figure 1 shows the architecture of OpenFlow in-band control deployment. Both the control signal and the data traffic use the same wireless links among forwarders. On one hand, in-band control network is better than out-of-band control network because it saves one WMN, lowering the costs and complexity. On the other hand, the links between forwarders have to carry out two kinds of traffic, which means the control signal occupies the limited wireless bandwidth resources and the network may suffer performance degradation.

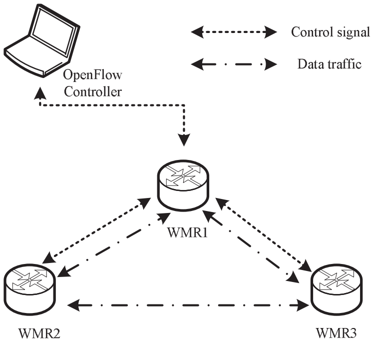

Fig. 1. In-band control scheme.

Figure 2 shows the architecture of OpenFlow out-of-band control deployment. The control signal and data traffic pass through two different networks. It is reasonable to use two WMNs to transmit control signal and data traffic, however, the wireless links between forwarders may fail occasionally, if this happens the controller cannot send control signal to the corresponding forwarders, which means these forwarders cannot route any data traffic any more. The network performance is bad because the control signal cannot reach forwarders. [7] points out the wired control network solution. This configuration may not be optimal, but our goal is to prove that the SDN architecture has the ability to manage the WMNs and improve network performance. Using the wired control network is to make sure the control signal is effective, and on that basis, we can measure the effect that SDN architecture has on WMNs more accurately. Therefore, we deploy the wired out-of-band control network over WMNs in our experiments.

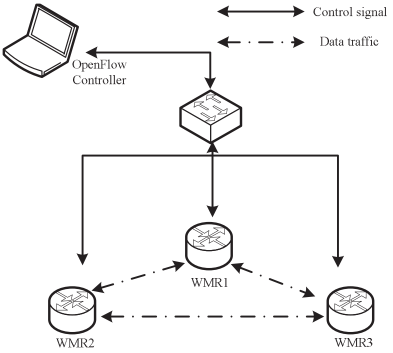

Fig. 2. Out-of-band control scheme.

3. The SDN-based Link Saturation Routing Algorithm

One of the most notable characteristics of SDN architecture is that controller can obtain the full topology of the network through OpenFlow protocol. Controller needs a link quality model to find the optimal path from the overall topology of the network for the coming flows. We introduce link saturation to SDN architecture, from which controller can figure out the link congested situation and reroute the following flows to other better paths. At the same time, the congested path still forwards the flows which were active before the congested situation happened. This is because that once the forwarding rules of flows have been configured in the forwarders, the same flows will be forwarded according to the rules in the forwarders and never be sent the controller. Therefore, the traffic on the congested path will gradually reduced and the congested situation will be improved. After a while, when the controller figures out that the metric of this path becomes optimal again, it will allocate new coming flows onto this path again.

FAMTAR in [8] also uses the forwarding table to find better paths when congestion happens, however, due to the limitations of the traditional flooding routing method, the fine-grained routing algorithms will lead to severe increase of control overhead. Therefore, they only use two thresholds to represent the congested situation and non-congested situation respectively, which are $0.9\times C$ and $0.7\times C$, where C is the link capacity. The proposed link saturation algorithm, by contrast, can reflect the link congested situations more accurately. Moreover, our scheme just need to add the one parameter, which is link saturation, to the standard OpenFlow protocol in the control network, which means our scheme will have little influence on the increase of the control overhead.

Figure 3 shows the calculation process of the link saturation. The forwarder traverses its flow table and Num represents the number of existing flows on the link. $B_{\max}$ is the capacity of the link, B_{left} is the bandwidth which is still available on the link, $Flow(i).B$ is the bandwidth of the ith existing flow, γ is a constant and $\gamma B_{\max}$ represents the threshold we consider that the link starts to become congested, $Overload[i]$ is a binary variable means that whether the ith flow makes the link become congested or not, Pol represents the link saturation.

```
procedure SDN_Pol( )
    init i Num Overload[ ]
    whlie $i \le Num$ do
        $B_{left} = B_{max} - Flow(i).B$
        i++
        if $B_{left} \le \gamma B_{max}$
        $Overload[i] = 1$
    end while
    $Pol = \frac{\sum_{i=0}^{Num} Overload[i] \cdot Flow(i).B}{B_{max}}$
end procedure
```

Fig. 3. Calculation process of Pol

The forwarders report the link saturation to the controller periodically. The controller collects the information from all the forwarders and chooses the optimal path between the source site and destination site according to the link quality model, which can be summarized as follow:

$$\begin{cases} Cog_{path} = 1 - \prod_{l \in path} (1 - Pol_l) \\ T_{path} = \sum_{l \in path} T_l \\ M_{path} = (1-\alpha) T_{path} + \alpha Cog_{path}, \qquad 0 \le \alpha \le 1 \end{cases} \tag{1}$$

where Pol_l represents the saturation parameter of lth link of the path, Cog_{path} represents the congested situation of the path and $Cog_{path} > 0$ implies that the path has been considered as congested. The path will be considered as better if it has smaller Cog_{path} because it means that the path still has more bandwidth resources available. T_{path} represents the delay of the path, T_l represents the delay of the link l, which belongs to the path, α is a scale factor, it determines the relative importance of the delay and congested situation depending on the network and traffic characteristics. For large α, the path selection will be more sensitive to congested situation. On the contrary, for small α the path selection will be more sensitive to delay. M_{path} is the matrix of the path, and the smaller M_{path} value the path has, the better it is.

4. Simulation Results

We have built a test network to evaluate the performance of the SDN-based link saturation routing algorithm using the topology in Figure 4. Opendaylight [9] is an open platform for network programmability to enable SDN, therefore we use the Opendaylight controller to manage the out-of-band control network structure and determine the behavior of the data network in Figure 4. The controller directs the data traffic to the optimal route between Host1 and Host2 according to the SDN-based link saturation routing algorithm. The data network Open vSwitch [10] topology is generated using Mininet [11], which can create Open vSwitch topology on a single Linux kernel efficiently. The capacity of the wireless links between Open vSwitches is set to 100 Mbit/s, the delay of theses links ranges from 1ms to 3ms, and the packet loss rate of these links is set to 2%.

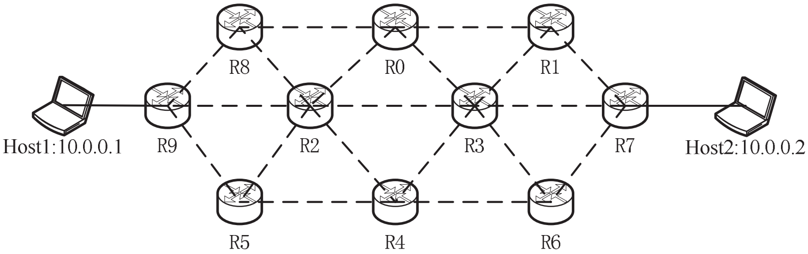

Fig. 4. Simulation topology

We use the Iperf tool to generate the data traffic that goes through the data network. Specifically, Host2 runs as the Iperf server which listens on a specific port while Host1 runs as the Iperf client which generates the data traffic to the corresponding port on Host2. The data traffic volume ranges from 0 to 400Mbps and we observe the packets generated at Host1 and packets received at Host2 to evaluate the performance of the routing algorithm. Moreover, we compare SDN-based routing strategy to OLSR and FAMTAR to demonstrate its superiority.

Figure 5 shows the result of average throughput of the three routing schemes during the observed time period in relation to the data traffic that passes through on the data network. We can see that the amount of average traffic increases to the limit of 100Mbps, it is because that the traditional OLSR routing protocol always uses only the best path and route all the traffic onto the path even if it becomes congested. Therefore the average throughput of the data network is quite low in spite of the fact that the data network has much more bandwidth resources. In contrast, the maximum average throughput of FAMTAR can reach 250Mbps as the data rate increases. It is because that when the optimal path becomes congested, FAMTAR stops pushing the following data traffic to the congested path and reroutes it to sub-optimal path while the congested path still forwards all the flows which were active before the congestion was noticed. Moreover, the proposed SDNR has the best average throughput due to the link saturation routing algorithm in contrast with the double thresholds scheme of FAMTAR. We can clearly see that the maximum average throughput of SDNR is nearly 300Mbps which is close to the bandwidth capacity of the paths in the topology.

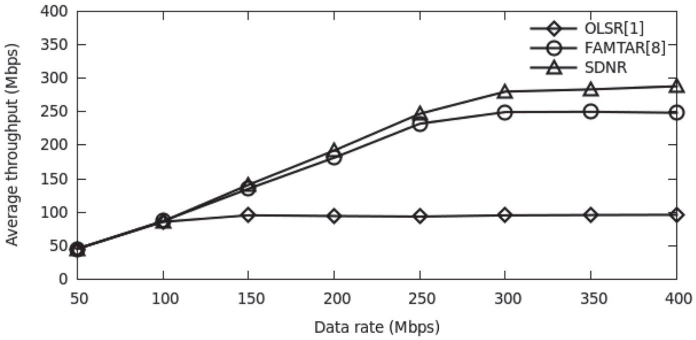

Fig. 5. Average throughput of three routing schemes in relation to data rate

Figure 6 shows the result of packet delivery ratio of the three routing schemes in relation to the data traffic. We can see that the packet delivery ratio of OLSR decreases seriously when the data rate is greater than 100Mbps. It is because that the optimal path has been congested since the data traffic was greater than its bandwidth capacity. The longer and longer queue time leads to the more and more serious packet loss. Similar situation happens to FAMTAR when the data rate exceeds 250Mbps. It is because that when all the paths become congested, FAMTAR pushes all the following data traffic to the original optimal path which does not improve the queuing situation of the following data traffic in contrast with OLSR. Therefore the downward trend of the two curves is the same. By contrast, SDNR has better performance on PDR because when the data traffic is greater than bandwidth capacity it will also push flows to the least congested path according to the link saturation and different paths may be selected when the network congested situation is serious. The queue time of each path will be shortened in contrast with OLSR and FAMTAR, therefore the downward trend of PDR curve is improved.

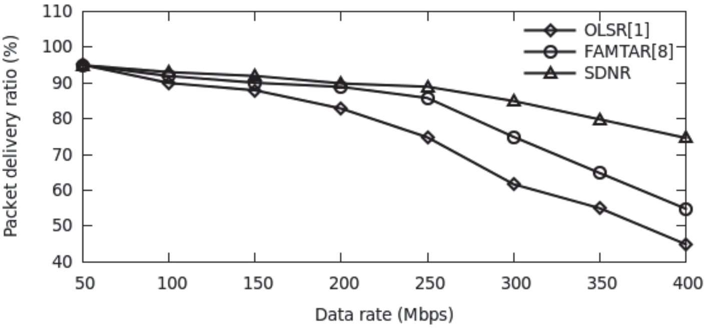

Fig. 6. Packet delivery ratio of three routing schemes in relation to data rate.

Figure 7 shows the result of normalized routing overhead of the three routing scheme in relation to the data traffic. Firstly, we can see that OLSR has a stable control overhead as the data rate increases, it is because that OLSR always routes the traffic to the optimal path regardless of the congested situation. Moreover, we can clearly find out that SDNR always has smaller normalized routing overhead than FAMTAR as the data rate increases. It is because that the FAMTAR builds and maintain the routing policy with the traditional flooding method, while SDNR uses the controller to construct all the routing policy to all the forwarders. With the centralized control structure, the forwarders just need to contact with the one controller instead of all the other forwarders in FAMTAR.

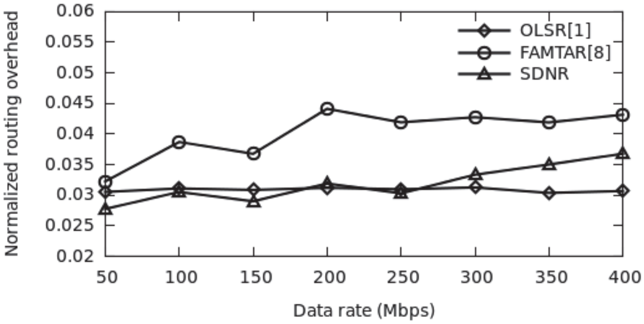

Fig. 7. Normalized routing overhead of three routing schemes in relation to data rate.

5. Conclusions

In this paper, we propose a new approach to route packets based on SDN control structure. We introduce link saturation to SDN, and with which we design our link quality model and advocate SDNR. We deployed out-of-band control structure over WMN with Opendaylight controller and Mininet to conduct our experiments on SDNR. Simulation results show that SDNR has good performance on average throughput, packet delivery ratio and normalized routing overhead.

Acknowledgments

This Work is jointly supported by Beijing Higher Education Young Elite Teacher Project (YETP0440), National Natural Science Foundation of China. (No.61272518 and 61302083)

References

1. T. Clausen and P. Jacquet, *Optimized Link State Routing Protocol (OLSR)*, IETF Network Working Group: RFC3626-2003, October 2003.
2. P. Jacquet, P. Minet, A. Laouiti, L. Viennot, T. Clausen, and C. Adjih, *Multicast optimized link state routing*, IETF Internet Draft: draft-ietf-manet-olsr-molsr, November 2001.
3. Royer E. M. and Perkins C. E., *Multicast Ad Hoc On-Demand Distance Vector (MAODV) Routing*, IETF, Internet Draft: draft-ietf-manet-maodv-00.txt, July 2000.
4. Hilmi E. Egilmez, Seyhan Civanlar, and A. Murat Tekalp, *An Optimization Framework for QoS-Enabled Adaptive Video Streaming Over OpenFlow Networks*, IEEE Trans on Multimedia, vol. **15**, No. 3, April 2013, pp. 710–715.
5. Hilmi E. Egilmez, and A. Murat Tekalp, *Distributed QoS Architectures for Multimedia Streaming Over Software Defined Networks*, IEEE Trans on Multimedia, vol. **16**, no. 6, October 2014, pp. 1597–1609.
6. Andrea Detti, Claudio Pisa, Stefano Salsano, Nicola Blefari-Melazzi, *Wireless Mesh Software Defined Networks (wmSDN)*, 2nd International Workshop on Community Networks and Bottom-up-Broadband, 2013, pp. 90–95.
7. J. Chung, G. González, I. Armuelles, T. Robles, R. Alcarria, and A. Morales, *Experiences and Challenges in Deploying OpenFlow over a Real Wireless Mesh Network*, IEEE Latin America Trans, vol. **11**, no. 3, May 2013, pp. 955–961.
8. Robert Wójcik, Jerzy Domzał, and Zbigniew Dulinski, *Flow-Aware Multi-Topology Adaptive Routing*, IEEE Communications Letters, vol. 18, no. 9, September 2014, pp. 1539–1542.

9. Opendaylight controller website, https://www.opendaylight.org/.
10. Open vSwitch website, http://openvswitch.org/.
11. Mininet website, http://mininet.org/.