Research on Traffic Engineering in SDN

Siddhant Sawant, Jenil Joseph, Bonyl Santhmayor, Jayson Ruzario
Department of Electronics and Tele-Communication,
St. John College of Engineering and Management, Palghar, India

Abstract: Common objective of network traffic engineering is to minimize the maximum link utilization in network in order to accommodate more traffic and reduce the chance of congestion. TE plays a critical role in determining the performances and reliability of a network. Major challenge in Traffic engineering is how to cope with unpredictable changes in traffic demand. Traffic engineering (TE) is an important network application, which studies measurement and management of network traffic, and designs reasonable routing mechanisms to guide network traffic to improve utilization of network resources, and better meet requirements of the network quality of service (QoS).

IndexTerms- Traffic, SDN, network topology, forwarding rules.

I. Introduction

A. Software Defined Networking (SDN) opens new possibilities in the field of networking. It allows the definition of new control plane protocols in software and provides the possibility to make use of centralized network information, like topology, available paths, or current traffic distribution. It also allows the forwarding of packets at different granularities, providing the possibility of flow based forwarding, and the ability to define entire path of all packets.

The main idea of TE is to study how to measure and analyse real-time network traffic, and design reasonable routing mechanisms to schedule and guide network traffic to improve utilization of network resources, or better meet requirements of the network Quality of Service (QoS)[1]. The main goal of most applications is to engineer traffic with the aim of minimizing power consumption, maximizing aggregate network utilization, providing optimized load balancing, and other generic traffic optimization techniques. By using specialized optimization algorithms and diversified configuration options, it is possible to meet the infrastructure goals of latency, performance, and fault tolerance, for instance, while reducing power consumption[2]. With the use of simple techniques, such as shutting down links and devices intelligently in response to traffic load dynamics, data centre operators can save up to 50% of the network energy in normal traffic conditions. One of the important goals of data centre networks is to avoid or mitigate the effect of network bottlenecks on the operation of the computing services offered[3].

B. Related Work

The survey about SDN entirely focused on its applicability to TE, which studies the impact of the different SDN interfaces on TE. Surveys like [4] and [5] provide a general overview of SDN. Most surveys are focused entirely on OpenFlow and its applications [6] and [7], some of which deal briefly with TE. However, none of the surveys analyse how different SDN protocols impact TE, and how the impact depends on the interface at which the protocol operates.

II. Definition of TE

In communication networks, TE consists in the application of strategies and scientific principles to optimise the performance of operational networks[4]. The general objective of TE is to route traffic in a data network so that traffic demands are met, by optimising a selected performance objective. This usually involves the computation of a path between a given source-destination pair, or the computation of multiple paths to share the load according to specific traffic-splitting ratios.

SDN controllers are usually logically centralized even if their implementation is distributed. This centralization can cause some problems. When the number of incoming messages are too big to handle it creates bottle neck at the controller.

We can eradicate these problems by using a logically hierarchical control plane that separates the control plane of the core, where switches deal with more traffic, from the control plane in the access of the network where each switch deals only with local traffic.

III. Traffic engineering in SDN

Traditionally traffic is allocated to network resources (e.g. links, bandwidths) by each network node based on local decisions. These decisions are based in the result of distributed shortest path algorithms calculated locally according to centrally assigned links weights. If more than one path is possible the decision is again local. The lack of global knowledge about the network and the inability of any network forwarding element to control anything except the next hop where traffic is directed to translates into poor network utilization rates with some links becoming congested while some parts of the network are under used. In such a setting the only way to influence traffic distribution is to change the link weights to obtain the best set of paths.

Software Defined Networking (SDN) opens new possibilities in this context. It allows the definition of new control plane protocols in software and provides the possibility to make use of centralized network information, like topology, available paths, or current traffic distribution. It also allows the forwarding of packets, providing the possibility of flow based forwarding, and the ability to define the entire path of all packets of a flow.

IV. Traditional network TE technologies

Traditional network TE technologies mainly include IP-based TE and MPLS based TE.

1. In general, IP-based TE solves the problem of multipath traffic load balance by Traffic load balancing is properly balancing the data load on the different paths.

The equivalence multipath routing technology (ECMP) based on a hash algorithm is an effective load balancing solution.

When the packet arrives at a switch or router, switch or router extracts the header fields of the packet to make hash calculations and then selects one of the forward paths, according to the hash value.

Now, IP packets having the same head are forwarded along the same path.

A clear defect of the ECMP is that a lot of large flows, called elephant flows, are forwarded to the same path, which results in load imbalance and bandwidth waste[1].

optimizing the IP routing algorithm to avoid network congestions.

IP-based TE technology has two clear drawbacks:

1. When Open shortest path first(OSPF) link weights are used to control routing of a network, traffic cannot be split in an unpredictable proportion, leading to inability to make full use of network resources.
2. Second, when links fail of the network topology change, the OSPF protocol takes some time to converge to a new network topology, which possibly leads to network congestions, packet losses, delays, and even routing loops.

In order to avoid these defects of IP-based TE, researchers proposed another solution, in which network packets are forwarded by the Multi-Protocol Label Switching (MPLS), instead of IP headers. However, the protocol mechanism of the MPLS is too complex, and can lead to a high performance overhead, so it is difficult to satisfy requirements of data centre networks demanding high link bandwidth utilization, green energy saving, and high reliability[5].

V. Forwarding rules definition

The number of flow entries in a switch is equal to the number of equally short paths for the reachable destinations (access switches), which is exactly the same as with traditional routing[4]. However, since the rules are set by the controllers based on the results of the path discovery algorithm, the tag header value in a packet will determine the entire path that it will travel.

A. Control logic for core switches

Low controller-node communication is achieved by the use of a set of predefined paths calculated by the common services centres(CSC). Rules are installed pro-actively and communication is only needed if the network changes or failures occur. Packets that do not match any rule are dropped and no OpenFlow packet In messages are sent to the CSC, since it does not perform any type of re-active function[5]. There is also no gathering of flow statistics or other information by the CSC.

B. Control logic for access switches

Access switches deal with traffic from the hosts directly connected to them. Upon the reception of a new untagged packet the access switch sends a Packet In OpenFlow message to the ASC. The ASC then defines and installs an OpenFlow flow entry. The entry has a match clause that matches all traffic with the same source IP, destination IP and transport port triplet. It also has an action that sets the value of the tag header in the matching packets and therefore defines the path those packets will use. It is the definition by the ASC of these flow entries that enforces the traffic placement algorithm. Upon the reception of the first packet from the switch, the ASC checks the available paths for the destination. When more than one path is available the flow is distributed according to the paths centralities among all paths that are not in a near congestion situation.

VI. Traffic Engineering Measurement and Management in SDN

A. Network Parameters Measurement:

1. **Network Topology Parameters** basically refers to analysing which switches are connected directly to each other and construct the global topology using Link Layer Discovery Protocol (LLDP).
2. **Network traffic parameters** refer to the number or speed of network packets that pass through network equipment or a network port, such as the total number of IP packets, the number of bytes per second and so on.
3. **Network performance parameters** mainly include the network latency, bandwidth utilization, throughput rates, packet loss rates etc[1].

B. Traffic Management in SDN

A simple solution is to split traffic from the package level not the flow level and to identify elephant flows first, and then choose the right path by the controller.

2. **QoS-guarantee scheduling**

Hi QoS is a efficient QoS-guarantee solution in the SDN. The Hi QoS identifies multiple paths between source and destination nodes by queuing mechanisms to guarantee QoS for different types of traffic.Also Hi QoS can recover from link failure very quickly by rerouting traffic from failed paths to other available paths.

3. **Energy Saving Scheduling**

There are two types of methods to save energy, link rate adaptive and sleep models. Link rate adaptive methods dynamically adjust link rates according to traffic demand, because the energy consumption of links depends on their data transmission loads, rather than utilization[5], On the other hand, the sleep model powers off some network components, or translates some non-working components into sleep mode to save energy

VII. RESEARCH CHALLENGES

Traffic Analysis: Efficiently handle big data in the context of user behaviour, locality in SDN-TE needs to be addressed.

Traffic Monitoring: Significantly reduce the network overhead when SDN controller(s) or monitoring devices collect the network statistics with high accuracy is a topic for further study.

Network Check and Program Debugging Method: Methods to quickly detect or prevent intrusions by using network verification or programming error checking approaches for SDN are still largely unexploited.

VIII. CONCLUSION

This paper presents the development of the traffic engineering technology of the SDN is an important aspect. Regarding network management, the main goal is to provide reasonable traffic scheduling strategies according to network parameters obtained using flow measurement technologies to satisfy specific requirements.

References

1. Traffic Engineering in Software-Defined Networking: Measurement and Management, June 21, 2016

2. SDN based Traffic Engineering without Optimization: a Centrality based Approach IEEE ICC 2017 SAC Symposium SDN & NFV Track.

3. D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for traffic engineering over MPLS," Internet Requests for Comments, RFC 2702, Sep. 1999.

4. Y. Jarraya, T. Madi, and M. Debbabi, "A survey and a layered taxonomy of software-defined networking," IEEE Commun. Surveys Tuts., vol. 16, no. 4, pp. 1955–1980, 4th Quart., 2014.

5. Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," IEEE Commun. Surveys Tuts., vol. 18, no. 1, pp. 602–622, 1st Quart., 2016.

6. A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using OpenFlow: A survey," IEEE Commun. Surveys Tuts., vol. 16, no. 1, pp. 493–512, 1st Quart., 2014.

7. I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A roadmap for traffic engineering in SDN-OpenFlow networks," Comput. Netw., vol. 71, pp. 1–30, Oct. 2014

8. A. Pathak, M. Zhang, Y. C. Hu, R. Mahajan, and D. Maltz, "Latency inflation with MPLS-based traffic engineering," in Proc. ACM SIGCOMM Conf. Internet Meas. (IMC), Berlin, Germany, 2011, pp. 463–472.

9. W.-Y. Huang, T.-Y. Chou, J.-W. Hu, and T.-L. Liu, ''Automatical end to end topology discovery and flow viewer on SDN,'' in Proc. 28th Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA), May 2014, pp. 910–915