International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Application Aware Routing in SDN

Harishkumar Bhokare[1], Mansi Bhonsle[2]

[1]ME Student, G.H. Raisoni College of Engineering and Management, Chas, Ahmednagar- 414001, India

[2]Professor, G.H. Raisoni Institute of Engineering and Technology, Wagholi, Pune- 412207, India

Abstract: ***Software-Defined Networking (SDN) is an emerging architecture that is dynamic, manage-able, cost-effective, and adaptable, making it ideal for the high-bandwidth, dynamic nature of today's applications. This architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. The Open Flow protocol is a foundational element for building SDN solutions. Moreover it is based on Object oriented concepts where entire network architecture is observed as objects. Software Defined Networks (SDN) promises to make high-capacity networks cheaper to build and easier to configure, not to mention faster and more efficient. As more and more computing moves to the cloud, those network improvements will be critical to keep everything affordable and available.***

Keywords: SDN- Software Defined Networks, Open Flow Protocol, POX, CAPEX/OPEX

1. Introduction

Traditional networks were designed to forward packets from source to destination using the shortest route possible. Routers and switches were mostly agnostic to the applications being served by the network. Today's networks are forced to be application-aware, to improve user experience, provide service differentiation, and reduce operational costs. Software-defined network (SDN) architecture allows service providers to build networks with increased application awareness, which can be built into the network by developing SDN controller applications that keep track of application-level characteristics and use that intelligence to provision flow into the network switches.

2. Related Work

Traditional routing techniques, the ones in use today, are not application-aware. The application-deliver infrastructure in today's internet is loosely-coupled with the packet forwarding infrastructure, which leads to duplication of functions in the network. This, in turn, impacts user experience, and increases operational costs. The request routing techniques in use today are-

1) Policy-Based Routing (PBR)
2) BGP Route Advertisement
3) Use of a Centralized Request Router

These techniques have some limitations, which are listed below-

1) The absence of mechanisms for the centralized request router to communicate with the L3 router to dynamically increase the bandwidth availability for a given user session. The network is always provisioned for peak bandwidth usage, resulting in unused bandwidth.
2) Latency requirements of the application are not taken into consideration by L3 routers when routing the requests.
3) CAPEX is increased as a result of maintaining a dedicated infrastructure for L7 request routing in addition to the existing routing infrastructure.

Software-defined network (SDN) architecture allows service providers to build networks with increased application awareness, which can be built into the network by developing SDN controller applications that keep track of application-level characteristics and use that intelligence to provision flow into the network switches. SDN technology opens up an array of opportunities for network service providers. It helps them roll out monetizable services in their networks faster without having to depend on network equipment manufacturers. Application-aware routing architecture using SDN allows service providers to lower CAPEX/OPEX, and improve the overall end-user experience.

3. Design and Implementation

3.1 Methodology

When the client transmits an IP packet, the switch inspects the packet and depending on the policy/rule installed in it, forwards the packet on a particular route. If the switch doesn't have any policy/rule installed, it sends the packet to the controller. The controller inspects the packet header and/or the payload, determines the type of packet (TCP, UDP, HTTP, etc.), and installs a policy/rule on the switch instructing it to forward packets along a particular route.

3.2 Implementation

We have created different paths for TCP, UDP and ICMP traffic. We have further classified TCP traffic in to HTTP and FTP traffic. For each different type of traffic, we have assigned a dedicated path. Our implementation could easily be extended to be adaptive, so that the features of an application-aware network mentioned above could be realized.

Figure 1: SDN Methodology

When the switch forwards the packet it receives to the controller, the controller checks the Ethernet packet for IP content.

1) If it's indeed an IP packet, the controller then checks the protocol header to determine if it's a TCP, UDP or an ICMP packet.
2) If the packet is a TCP packet, then it further determines if the packet is HTTP or FTP by inspecting the port number.
3) After the controller determines the type of packet, it instructs the switch to send the packet along a specific path/route.
4) According to our topology-
   - HTTP packets are routed along S3.
   - FTP packets are routed along S6.
   - Other TCP packets are routed along S7.
   - UDP packets are routed along S4.
   - ICMP packets are routed along S5.

We identify different packets as being TCP/UDP using the identifier in the IP header for the transport protocol, after IP packets themselves are identified using an identifier in the Ethernet header. The applications are identified using the server port being contacted by the client. We class packets as belonging to an FTP connection if the server port number is 21, and as HTTP if the server port being contacted is 8000

4. Conclusion

The controller installs a rule on the switch, instructing it to forward the packets along the different paths depending on the type of the packet. The controller inspects the Ethernet packet for IP content and further for TCP, UDP and ICMP content based on the protocol header. TCP packets are further inspected for HTTP and FTP traffic by performing a string search in the payload for keywords (HTTP 1.1 for HTTP traffic and FTP/SFTP for FTP traffic).

Only the first packet from a flow is sent to the controller, as in Pyretic/POX. Thereafter, a rule is installed on the switch for the flow. This allows the flow to be dealt with by the switch itself, saving the controller from having to deal with too many packets.

References

[1] Avalanche: Data Center Multicast using Software Defined Networking
[2] http://www-03.ibm.com/systems/networking/sdn/
[3] https://www.opennetworking.org/sdn-resources/sdn-definition
[4] http://en.wikipedia.org/wiki/Software-defined_networking
[5] http://www.opendaylight.org/
[6] http://archive.openflow.org/wp/documents/

Author Profile

Harishkumar Bhokare received the B.E degrees in Electronics and Telecommunication Engineering from Sinhgad College of Engineering, Pune in 2013. He was working with QLogic India Private Limited, Pune from 2013 to 2015. He is currently working with IBM India Private Limited.