Department of Computer Engineering

Computer Networks

Homework 3

Dr. Jafari

Parsa Mohammadian — 98102284

July 10, 2022

Contents

1

Software-Defined Networking or SDN is a networking architecture that uses software-based controllers or APIs[1] to implement network functions for hardware-based infrastructure.
The architecture of SDN is shown in Figure 1. Since the control layer of network is decoupled from the infrastructure layer, SDN has the following criterias:

- Directly programmable
- Agile
- Centerally managed
- Programmatically configured
- Open standard-based and vendor neutral

SDN can be implementd in three main ways:

- Open SDN: The controller comunicates with the switches using OpenFlow protocol as shown in Figure 2.
- SDN via APIs: The functions in remote devices are implemented using APIs(SNMP, CLI, Rest, ...) as shown in Figure 3.
- SDN via hypervisor-based overlay network: Hypervisor-based overlay networks are created over the top of the physical network. The hypervisor is responsible for the control layer of the network as shown in Figure 4.

SDN advantages:

- Centeralized controller
- Reduced lines of configuration code due to seperation of data and control plane
- The code is written only once and can be used in multiple environments
- The controller features can eliminate the need for external devices and overaly reduce the cost of the network
- Openness

Resources

- https://youtube.com/watch?v=Z5Gi2Bpd82M
- https://youtube.com/watch?v=Nh2hXUuKXyQ
- https://opennetworking.org/sdn-definition/
- https://www.geeksforgeeks.org/types-of-software-defined-networks-implementation/

[1] Application Programming Interface

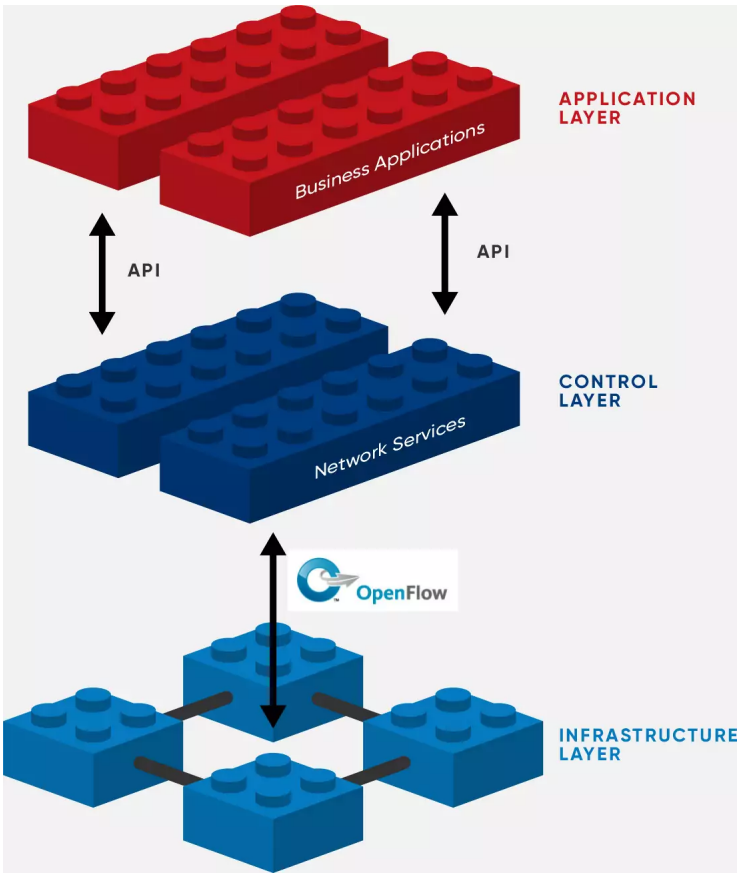


Figure 1: SDN Architecture

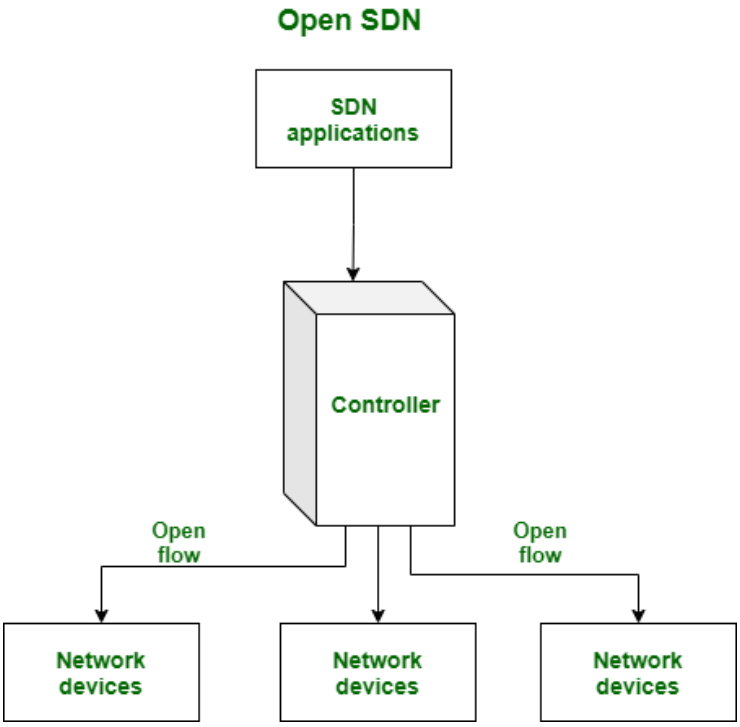


Figure 2: Open SDN

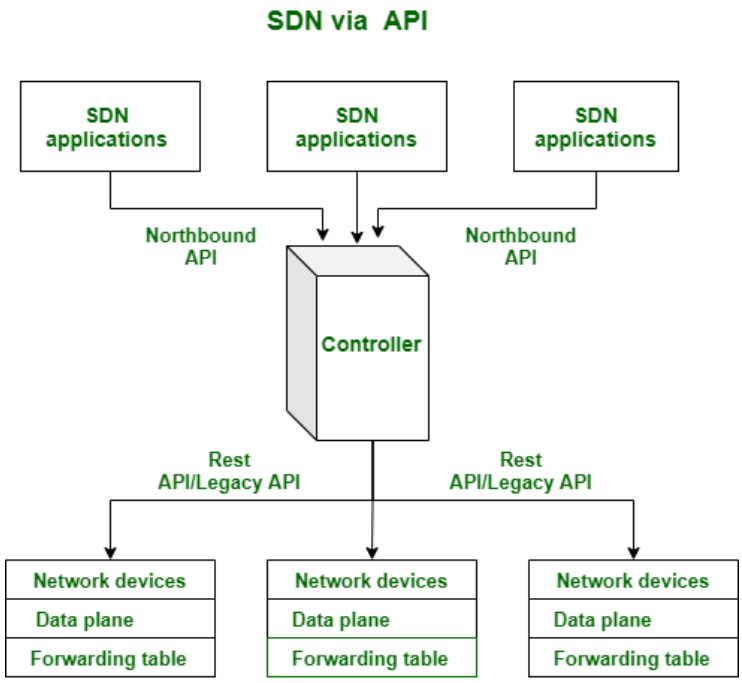


Figure 3: SDN via APIs

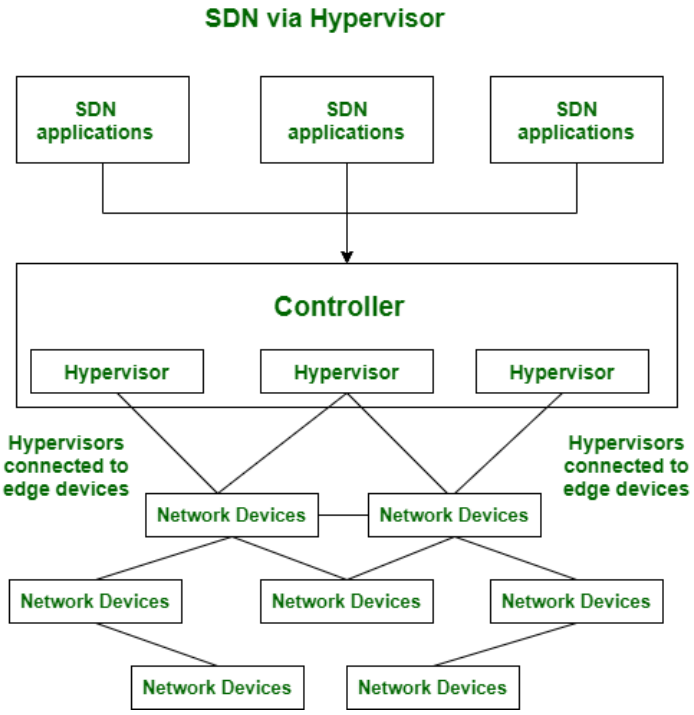


Figure 4: SDN via hypervisor-based overlay network

2

2.1

Telegram normal messages use server-client/client-server encryption. It means they are secure while transmitting between clients and servers in a way that only the server and the client can decrypt the message. Then in the server all this messages are stored in a secure database. This way Telegram allows multiple clients to connect as a same user. On the other hand, Telegram secret chats are client-client encrypted. At the start of the secret chat, participants exchange encryption keys. Then the messages are encrypted using the keys exchanged. This means that only two sides of the conversation can decrypt the message. In such a case, the server can not decrypt the message at all. Because of that, newly connected clients can not access the secret chat.

Resources

- https://telegram.org/faq

2.2

As mentioned in official WhatsApp documentation, WhatsApp uses end-to-end encryption. This is just like Telegram secret chats. Actually, this is the main reason why WhatsApp can not restore your messages on newly connected clinet if you lost all your logged in devices. So WhatsApp is not peer-to-peer.

Resources

- https://www.whatsapp.com/security/

3

3.1

Table 1: Link State Routing

Step	N'	$D(R_2), P(R_2)$	$D(R_3), P(R_3)$	$D(R_4), P(R_4)$	$D(R_5), P(R_5)$
1	R_1	$5, R_1$	∞	$2, R_1$	∞
2	R_1R_4	$3, R_4$	$3, R_4$		$5, R_4$
3	$R_1R_4R_2$		$3, R_4$		$5, R_4$
4	$R_1R_4R_2R_3$				$5, R_4$
5	$R_1R_4R_2R_3R_5$				

3.2

Table 2: Distance Vector Step 1

R_1 Table	R_1	R_2	R_3	R_4	R_5
R_1	0	5	∞	2	∞
R_2	∞	∞	∞	∞	∞
R_3	∞	∞	∞	∞	∞
R_4	∞	∞	∞	∞	∞
R_5	∞	∞	∞	∞	∞
R_2 Table	R_1	R_2	R_3	R_4	R_5
R_1	∞	∞	∞	∞	∞
R_2	5	0	4	1	∞
R_3	∞	∞	∞	∞	∞
R_4	∞	∞	∞	∞	∞
R_5	∞	∞	∞	∞	∞
R_3 Table	R_1	R_2	R_3	R_4	R_5
R_1	∞	∞	∞	∞	∞
R_2	∞	∞	∞	∞	∞
R_3	∞	4	0	1	2
R_4	∞	∞	∞	∞	∞
R_5	∞	∞	∞	∞	∞
R_4 Table	R_1	R_2	R_3	R_4	R_5
R_1	∞	∞	∞	∞	∞
R_2	∞	∞	∞	∞	∞
R_3	∞	∞	∞	∞	∞
R_4	2	1	1	0	3
R_5	∞	∞	∞	∞	∞
R_5 Table	R_1	R_2	R_3	R_4	R_5
R_1	∞	∞	∞	∞	∞
R_2	∞	∞	∞	∞	∞
R_3	∞	∞	∞	∞	∞
R_4	∞	∞	∞	∞	∞
R_5	∞	∞	2	3	0

Table 3: Distance Vector Step 2

R_1 Table	R_1	R_2	R_3	R_4	R_5
R_1	0	3	3	2	5
R_2	5	0	4	1	∞
R_3	∞	∞	∞	∞	∞
R_4	2	1	1	0	3
R_5	∞	∞	∞	∞	∞
R_2 Table	R_1	R_2	R_3	R_4	R_5
R_1	0	5	∞	2	∞
R_2	3	0	2	1	4
R_3	∞	4	0	1	2
R_4	2	1	1	0	3
R_5	∞	∞	∞	∞	∞
R_3 Table	R_1	R_2	R_3	R_4	R_5
R_1	∞	∞	∞	∞	∞
R_2	5	0	4	1	∞
R_3	3	2	0	1	2
R_4	2	1	1	0	3
R_5	∞	∞	2	3	0
R_4 Table	R_1	R_2	R_3	R_4	R_5
R_1	0	5	∞	2	∞
R_2	5	0	4	1	∞
R_3	∞	4	0	1	2
R_4	2	1	1	0	3
R_5	∞	∞	2	3	0
R_5 Table	R_1	R_2	R_3	R_4	R_5
R_1	∞	∞	∞	∞	∞
R_2	∞	∞	∞	∞	∞
R_3	∞	4	0	1	2
R_4	2	1	1	0	3
R_5	5	4	2	3	0

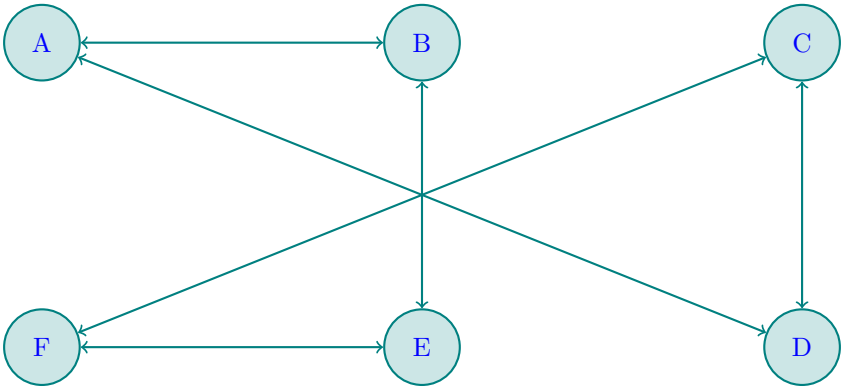
Table 4: Distance Vector Step 3 (Final)

R_1 Table	R_1	R_2	R_3	R_4	R_5
R_1	0	3	3	2	5
R_2	3	0	2	1	4
R_3	3	4	0	1	2
R_4	2	1	1	0	3
R_5	5	4	2	3	0
R_2 Table	R_1	R_2	R_3	R_4	R_5
R_1	0	5	∞	2	∞
R_2	3	0	2	1	4
R_3	3	4	0	1	2
R_4	2	1	1	0	3
R_5	5	4	2	3	0
R_3 Table	R_1	R_2	R_3	R_4	R_5
R_1	∞	∞	∞	∞	∞
R_2	5	0	4	1	∞
R_3	3	2	0	1	2
R_4	2	1	1	0	3
R_5	5	4	2	3	0
R_4 Table	R_1	R_2	R_3	R_4	R_5
R_1	0	5	∞	2	∞
R_2	5	0	4	1	∞
R_3	3	4	0	2	2
R_4	2	1	1	0	3
R_5	5	4	2	3	0
R_5 Table	R_1	R_2	R_3	R_4	R_5
R_1	∞	∞	∞	∞	∞
R_2	∞	∞	∞	∞	∞
R_3	3	4	0	1	2
R_4	2	1	1	0	3
R_5	5	4	2	3	0

Reverse poisoning is a technique to solve loop issue in distance vector routing. The problem is when a router goes offline, its neighbors update their routing table and brodcast the new table to all of their neighbors. If a router happen to transmit a packet to the offlien router through the updated router, it brodcast it to updated nodes and they will send packets with offline router destination back to these nodes. In order to prevent this, we ensure that a packet can not turn back to the same router.

4

A possible network graph is shown below.



4.1

$D \to C \to F$

4.2

$A \to B \to E \to F$

5

Actually, there where two problems with the 2021 Facebook(aka Meta) outage. First, the backbone network of the company was disconnected from the internet because of a command mistake by an employee. Second, becuase of the later issue, the company authorative name server refused to give out the IP addresses. This is because of a implemented feature, in which authorative name servers disable those advertisements that they can not speak to. The final result was the unaccessible network of the company.
The most trivial possible attack in this situation, is the physical access to data-centers becuase of the chaos. Aside from that, some attacker could poison DNS servers and add records to his/her own servers. This way he/she can steal login credentials of the company users.
While Facebook was out of access in all of the world, the Facebook IP addresses could be resolved in the Iran. Despite there is not a official reason for this, I can say that this was because of the intensive internet filtering in the country. Since Iran has mapped Facebook IP addresses to their own filtering servers prior to the outage, the IP addresses could be resolved in Iran, but the servers were not associated with the company.

Resources

- https://engineering.fb.com/2021/10/05/networking-traffic/outage-details/