CS358 Endsem Assignment

Name: M. Maheeth Reddy

Roll: 1801CS31

Maheeth

Date: 01-May-2021

Answers

Ans 1: Match-plus-action tables

For S1

Match	Action
IP Src: 128.119/16 IP Protocol: TCP IP Dst: 128.121/16	Forward(4)
IP Src: 128.119/16 IP Protocol: UDP IP Dst: 128.121/16	Forward(2)

For S2

Match	Action
Ingress port = 3 IP Src = 128.119/16 IP Dst = 128.121/16 IP Protocol = UDP	Forward(2)

For S3

Match	Action
Ingress port = 2 IP Dst = 128.121/16	Forward(1)
Ingress port = 4 IP Dst = 128.121/16	Forward(1)

For S4

Match	Action
ingress port = 2 IP Src = 128.119/16 IP Dst = 128.121/16 IP protocol = TCP	Forward(3)

(a) The value of Dst Port can be 'any' for router S1

(b) Both TCP and UDP protocols are used for router S1.
For interface 2, UDP is used for S1 → S2 → S3 path.
and for interface 4, TCP is used in S1 → S4 → S3 path

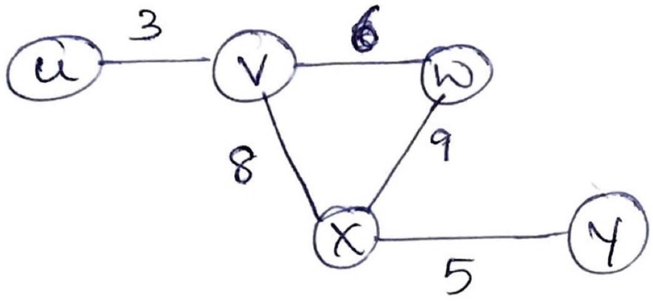
(c) IP Src for router S2 will be 128.119/16

(d) The action of rule must be forward as it forwards to 128.121/16

(e) S4 must forward packets to interface 3 to reach S3.

Ans 2

Given network is



(a) When the algorithm converges,

the distance vector from router X to all ~~other~~ routers is

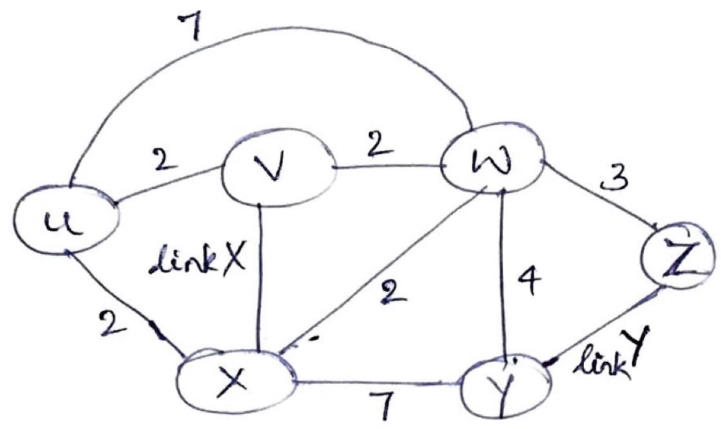
$$\boxed{(u, v, w, x, y) = (11, 8, 9, 0, 5)}$$

(b) The initial distance vectors for router Y ~~are~~ is

$$\boxed{(u, v, w, x, y) = (\infty, \infty, \infty, 5, 0)}$$

(c) The name of the problem that occurs when link costs increase is $\boxed{\text{Count to infinity problem}}$

Ans 3: Given 6-node network,



ⓐ For link X,

the node before V in the path to V is X.

It is given that the shortest path from X to V is 3

Hence, $\boxed{\text{the cost associated with link X is 3}}$

ⓑ As per the table, the shortest path to Z from X

is 5, and is along the path $X \rightarrow W \rightarrow Z$

So, the link Y will never be used.

Hence, $\boxed{\text{the answer is n/a}}$

Ans4

The LAN consists of 10 computers connected by two self-learning Ethernet switches.

Switch Table entries are provided. Based on the tables,

(a)

At $t=4$, the node communicating is C.

For TTL = 4, only one entry is there in table on left corresponding to C

(b) At $t=8$, nodes C, F are communicating

For TTL = 8, we have 2 entries, each corresponding to C and F in table on right

(c) At $t=3$, node J is communicating, because at TTL = 3, there is an entry for J in each table on left & right

(d) At $t=2$, node H is communicating.

At TTL = 2, we have an entry for H in each table on left & right.

Ans 5

Filled Switch Tables

Switch 2

Switch table for the left interfaces (8, 10, 11, 12, 13, 14, 15)

MAC Addr	Interface	TTL
F	8	1
I	12	1
C	8	2
J	13	2
E	8	3
G	10	3

Switch 1

Switch table for the right interfaces (1, 2, 3, 4, 5, 6, 7)

MAC Addr	Interface	TTL
F	6	1
I	7	1
C	3	2
J	7	2
E	5	3
G	7	3

(a) At t=1, source entry for switch 1 is (F, 6)

(b) At t=1, destination entry for switch 1 is (I, 7)

(c) At t=2, destination entry for switch 2 is (J, 13)

(d) At t=3, source entry for switch 1 is (E, 5)

(e) At t=3, destination entry for switch 2 is (G, 10)

Ans 6: Given,

$G = 1001$ (4-bit generator)

$D = 10011010$ (data payload)

$r = 3$

We need to find CRC bits, R for given data.

To find R, we need to get the remainder when $D \cdot 2^r$ is divided by G.

$D \cdot 2^r$ is equal to D appended by 'r' no. of zeroes.

$\Rightarrow$ $\boxed{D \cdot 2^r = 1001\,1010\,000}$ and $\boxed{G = 1001}$

Division b/n $D \cdot 2^r$ and G

```
          10001011
1001 ) 1001 1010 000
       1001
       --------
          0 1010
            1001
          ------
             1100
             1001
            -----
              1010
              1001
             -----
              0011
```

So, the remainder is 011

$\therefore$ $\boxed{\text{CRC bits, } R = 011}$

I have written the verification on next page.

Maheeth, 1801CS31, CS358, Maheeth

Verification that R=011 are correct CRC bits

First we append D with R, then divide the resultant with G. If the remainder is zero, then R is the correct CRC bits, and transmission will be successful.

Division of [DR] by G

```
          10001011
1001 ) 10011010011
       1001
       -----
        01010
         1001
         ----
          1101
          1001
          ----
           1001
           1001
           ----
              0    Remainder is zero
```

Hence, R=011 are the correct CRC bits

For, datapayload (D) = 10011010 and generator (G) = 1001 the CRC bits (R) are 011 for r=3

Ans 7: Since, the question doesn't specify the source and destination nodes, I have assumed source node as D and destination node as A.

We have been given the IP and MAC addresses for A,B,C,D.

(a) The source MAC address at point 5 is

1A-79-BC-E9-E6-3A

(b) The destination MAC address at point 5 is

D0-AE-4D-72-CC-F0

(c) The source IP address at point 5 is 128.119.58.43

(d) The destination IP address at point 5 is 128.119.190.195

(e) The source MAC address at point 2 is

E6-E3-64-04-2A-D3

(f) The destination MAC address at point 2 is

C2-32-12-7D-4E-86

Ans8: Given the packet payload,

Packet Payload Bits

Parity (row-wise)

1	0	1	0	0	1	1	1	0	1	0	0	1	0	0	0	1
1	0	1	1	1	1	0	1	0	0	1	1	0	0	0	0	0
1	0	1	1	1	0	1	0	0	1	0	1	0	1	1	0	1
0	1	1	0	0	0	0	1	1	0	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	1	1	0	1	0	0	0	0
1	1	1	0	1	1	0	0	0	1	0	1	1	1	1	0	

Packet Payload

Parity (column-wise)

(a) Parity bit is one if no. of ones in column is odd. otherwise it is zero.

∴ $\boxed{\text{2D parity for 16 columns is 1110110001011110}}$

(b) Same rule for rows gives the

$\boxed{\text{2D parity for 5 rows as 10100}}$

Ans 9

Oscillation problems faced in traditional computer network

Consider an example of Dijkstra's algorithm where oscillation could be possible

Link costs ~~b/w~~ between any two nodes in the network are dependent upon the amount of traffic volume passing through the link at any specific time. This is called dynamic cost policy.

Above said example,

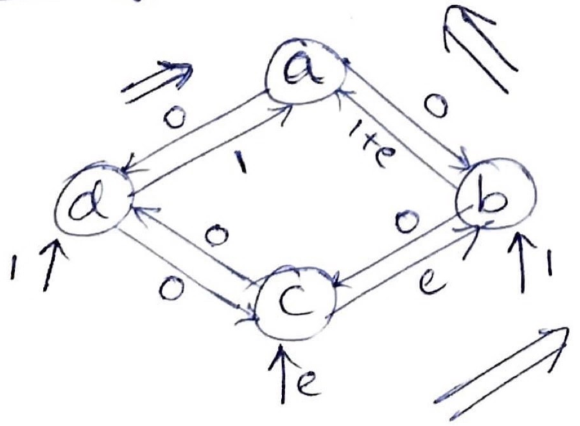
There are 4 nodes a, b, c, d. Link cost is equal to amount of load carried on the link.

Nodes b, c, d route the traffic only to node a.

Load received by a from b & d is one unit each.

Node c sends 'e' amount of load to node a

Initial routing



Double arrows represent optimal path

From the network topology in previous page, we can see that:

At this instant

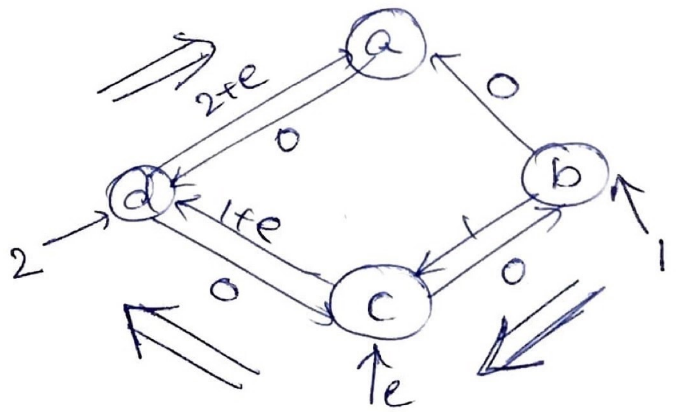
(i) Cost for clockwise path from C to a = 1

(ii) Cost for anticlockwise path from C to a = 1+2e

So, least cost path from C to a is in clockwise, so we need to change the initial route

Similarly, least cost path from b to a is also in clockwise, so we need to change the current path which is in anticlockwise. (1+e)

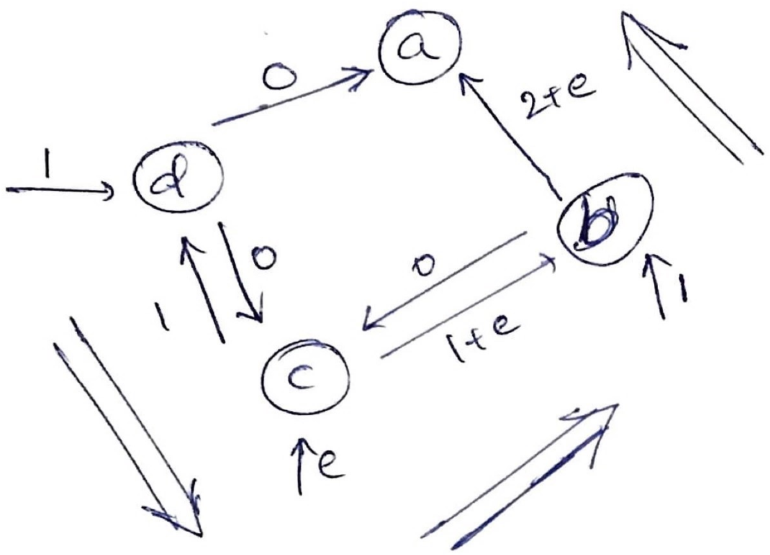
Modified Routing 1:



On running the algorithm for second time, nodes b,c,d detect an anticlockwise path with zero cost to a.

So, the traffic is routed along minimum cost direction.

Modified Routing 2:



Again on the next run, nodes b, c, d detect a zero cost path to a in clockwise direction and traffic is routed exactly similar to the previous routing (modified routing 1)

These modifications to the path will oscillate from clockwise to anticlockwise or viceversa as long as routing is required

This scenario is called as Oscillation problem in traditional computer network. Due to this problem, many routing do not define link costs based on short term load levels.

This problem has been overcome by the use of Software Defined Networking (SDN).

Maheeth, 1801CS31, CS358, Maheeth

Ans 10 Traditional networking is rooted in fixed-function network devices such as a switch or router.

If the network functions are implemented as hardware constructs, then its speed is usually bolstered.

Flexibility is a recurring hurdle for traditional networks. Few APIs are exposed for provisioning software, but this software can't be quickly modified as needed.

Traditional networking has the following traits:

(i) It is implemented from dedicated devices using one or more switches, as well as routers and application delivery controllers.

(ii) functionality implemented in dedicated hardware, such as specific ICs. Hence, it has a hardware centric limitation.

SDN on the other hand has the advantage of bolstering data intensive applications such as big data and virtualization. In addition to centralizing and simplifying the control of enterprise network management, SDN offers the following advantages:

i) Traffic Programmability

ii) Greater agility

iii) Capacity to generate policy driven network supervision

iv) Ability to implement network automation.

Along with these, the primary advantages SDN's offer are:

i) Centralized networking provision
ii) Holistic enterprise management
iii) Maximum granular Security
iv) Lower operating cost & hardware savings, hence reduced capital expenditure

Due to recurring problems in traditional networks like oscillation, difficult traffic engineering and the vast majority of advantages of SDN, SDN can be adopted.

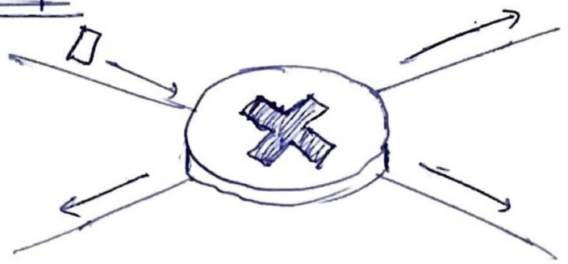
Network Substrate Formation :

The concept of network substrates comes when we talk about SDN. It is used for being able to mimic the success in the software industry.

Characteristics of network substrates are:

(i) Having simple common substrate
(ii) Building OS on top of hardware, which enable easy deployment of networking applications.

Router Example



Basic job of a router is receiving packets, checking the routing table, forwarding the packets out. For building the routing table, the router has to understand BGP, OSPF, RIP etc. We can use network substrates to get the routing table from somewhere else (centralized). This can also be seen in seperate data and control plane based architecture of SDN. Since, control plane is centralized and sits on top of all the software and hardware, it is the network's substrate.

Maheeth, 1801CS31, CS358, Maheeth

Ans 11

Software Defined Networking (SDN) is the physical separation of the network control plane from the forwarding plane and where a control plane controls several devices. SDN makes networks agile and flexible. It provides better network control for the cloud computing service providers to respond quickly to ever changing business requirements. Hence it is preferred over traditional network architecture.

A typical representation of SDN architecture includes 3 layers

1. Infrastructure Layer / Data Plane
2. Control layer / Control Plane
3. Application layer / Application Plane.

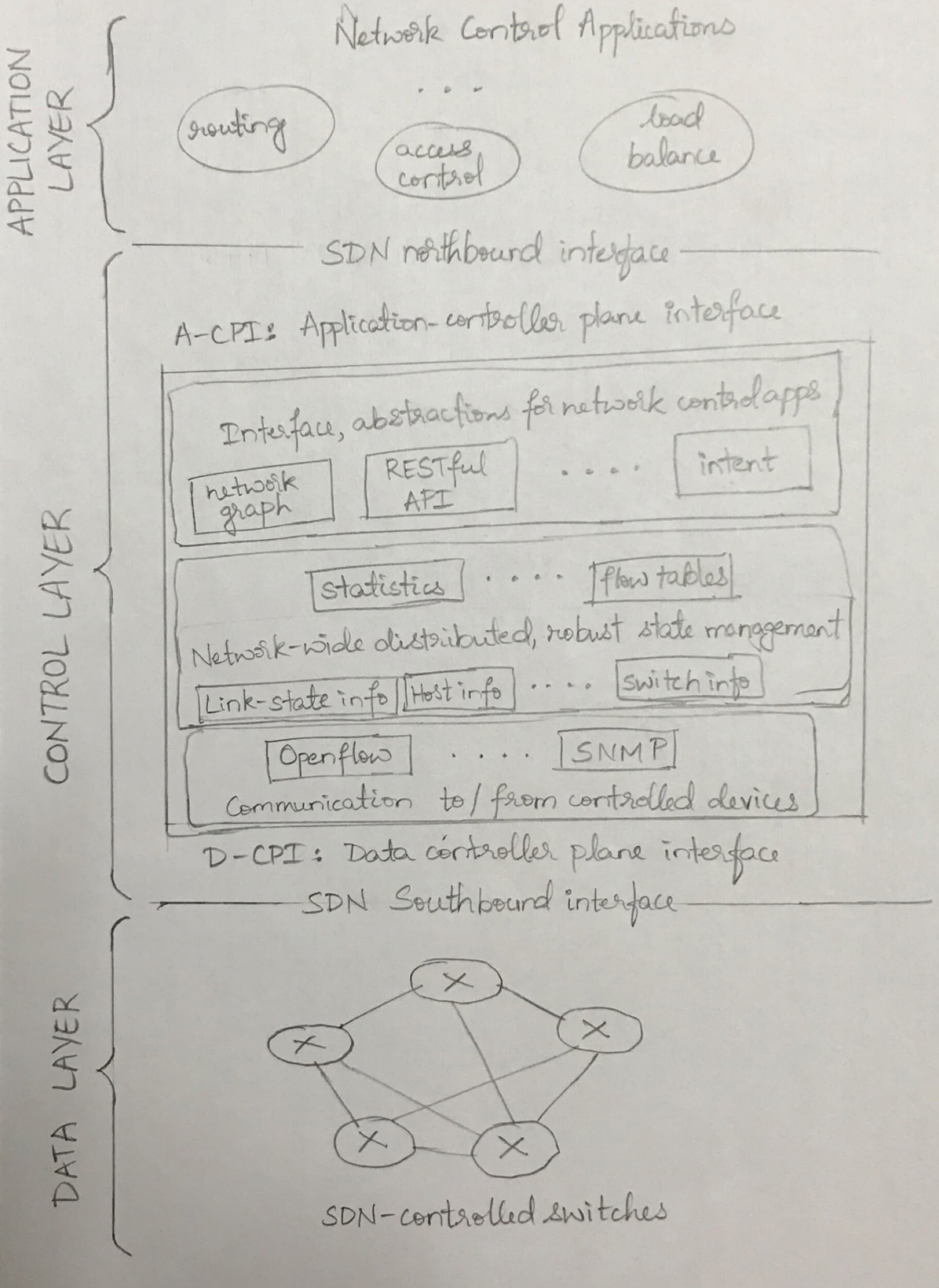
The data plane consists of network elements, which expose their capabilities to the control plane via southbound interface.

The SDN applications are in the application plane and communicate their network requirements towards the control plane via northbound interface.

Maheeth, 1801CS31, CS358, Maheeth

11.2

SDN architecture diagram



The control plane sits in the middle and has the following functions

→ translate the applications' requirements and exerts low-level control over the network elements

→ Provide network information to the applications

→ Orchestrate different applications.

Data Plane: It consists of data sources & sinks. It functions as a traffic forwarding/ processing engine and may have the ability to handle protocols like ARP, LLDP. It has interfaces communicating to the control plane. The core functionality of data plane is:

→ Programmatic control of all functions offered by network element

→ Capability advertisement.

→ Event notification.

Control Plane: It is considered as the brain of SDN. It communicates with the data plane using southbound APIs. The intelligence to this layer is provided by centralized SDN controller software. This controller resides on a server and manages policies and the flow of traffic throughout the network. The physical switches in the network constitute the data layer. It interfaces to the application layer.

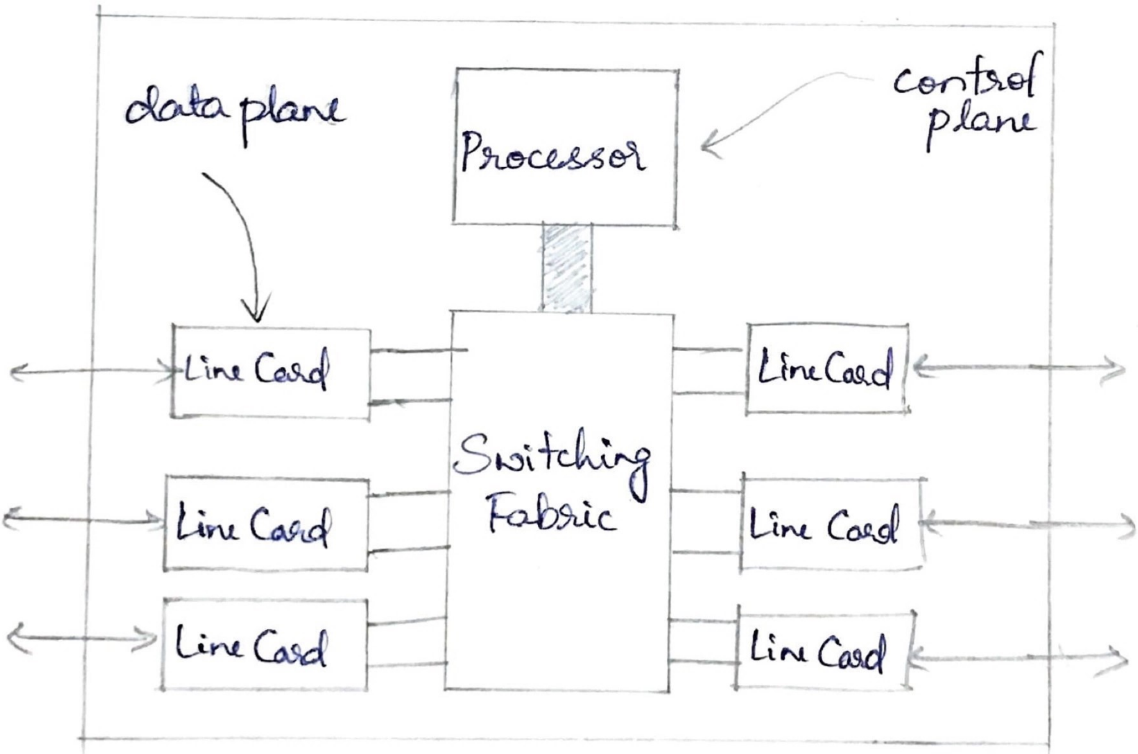
It's core functionality is:

→ Topology and network state information

→ Device discovery

→ Path computation

→ Security mechanism



Interaction between Data and Control planes via Switching Fabric.

Application Plane: The SDN application layer consists of network applications or functions such as intrusion detection systems, load balancing or firewalls. It communicates with control layer using the northbound API

Applications specify the resources and behaviours required from the network, with the context of business and policy agreement. The applications are developed using specific programming languages.

It may need to orchestrate multiple controllers to achieve the objectives.

Ans 12:

Data center transport impairments & requirements

The shallow packet buffers cause 3 specific performance impairments

① Incast:

If many flows converge on the same interface of a switch over a short period of time, packets may exhaust either switch memory or maximum permitted buffer, resulting in data losses. This traffic pattern arises naturally from use of partition/aggregate pattern as request for data creates incast at queue of switch port connected to aggregator.

We capture incast instances via packet-level monitoring. Since, size of each individual response is small, which leads to loss of packet almost invariably results in a TCP timeout. Thus, whenever timeout occurs response almost always misses aggregator's deadline.

② Queue buildup

Long-lived, greedy TCP flows will cause length of bottleneck queue ~~to~~ grow until packets are dropped resulting in familiar sawtooth patterns.

Two impairment occurs when short and long flows traverse same queue. Even when no packets are lost, short flows experience increased latency as they are in queue behind packets from large flows. This queue builds up, impairment occurs & this traffic pattern becomes more frequent.

③ Buffer pressure

It is very common for short flows on one port to be imparted by activity on any of many other ports. Instead the loss rate of short flows in this traffic pattern depends on no. of long flows traversing other ports. The long, greed TCP flows build up queues on their interfaces. Since, buffer space is a shared resource, the queue build up reduces the amount of buffer space available to absorb bursts of traffic from partition/aggregate traffic. We term this impairment as buffer pressure. The result is packet loss and timeouts as in incast but without requiring synchronized flows.

*

Tension Between requirements

High throughput High burst tolerance	Low latency
Deep buffers Reduced RTO_{min} (⇒ Doesn't help latency)	Shallow buffers AQM-RED (⇒ Average queue not fast enough for incast)

Case Study

Microsoft Bing

1. Measurements from 6000 server production cluster
2. Instrumentation passively collects logs.
   → Application-level
   → Socket-level
   → Selected packet-level
3. More than 150TB of compressed data over a month.

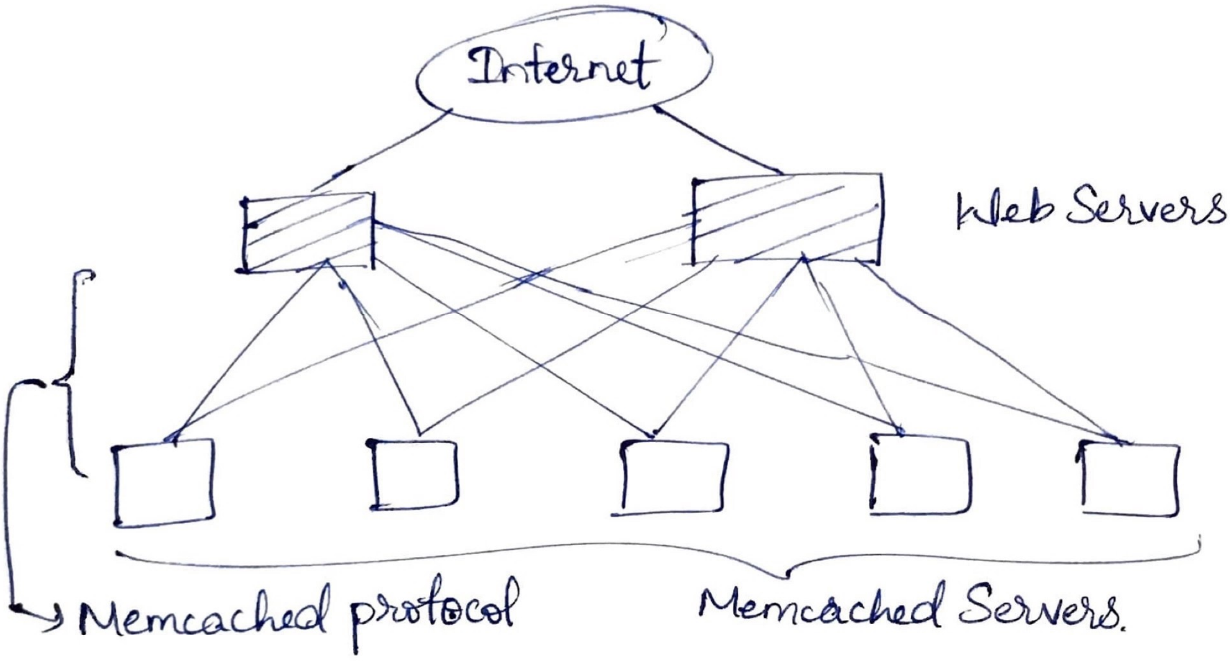
In case of Facebook, it uses Partition/aggregate which is the foundation for many large scale web apps.

Facebook DC

① Uses partition/aggregate ~ multiget

Aggregators : Webservers

Workers : Memcached Servers.



② Workloads are

i) Partition/aggregate (query) ⟶ Delay Sensitive

ii) Short messages [50KB, 1MB] ⟶ Delay Sensitive
(coordination, control state)

iii) Large flows [1MB, 50MB] ⟶ Throughput Sensitive
(Data Update)

Ans 13 (a) Forwarding Abstraction

Forwarding Abstraction is used to abstract away forwarding hardware while still being able to express how and where to forward a packet.

The first abstraction refers to how network elements forward packets. At a high enough level of abstraction all network elements perform the same task.

Abstraction 1: Packet forwarding as a computational problem.

The function of any network element is to:

(i) Receive a packet
(ii) Observe Packet fields
(iii) Apply algorithms
(iv) Optionally edit the packet
(v) Forward or discard the packet.

Features:

⇒ Flexible: ① forwarding behaviour specified by control plane
② built from basic set of forwarding primitives

⇒ Minimal: ① Streamlined for speed & low power
② Open & not vendor specific.

Openflow is an example of forwarding abstraction

~~Featu~~ Features of Openflow:

- Protocol independent
  i) Construct ethernet, IPv4, VLAN, MPLS, ----
  ii) Construct new forwarding methods.
- Backward compatible
  i) Run in existing networks
- Technology independent
  i) Switches, routers, WiFi APs
  ii) Cellular base stations
  iii) WDM/TDM circuits

(b) Network Functions Virtualization

It is a way to virtualize network services such as routers, firewalls & load balances, that have traditionally been run on proprietary hardware. These services are packaged as virtual machines on commodity hardware, which allows service providers to run their network on standard servers instead of proprietary ones. It is one of the primary components of a telco cloud, which is reshaping the telecommunication industry. With NFV we don't need to have dedicated Hardware for each network function. NVF improves scalability and agility by allowing service providers to deliver new network services and application on demands without requiring additional Hardware resources.

~~NVF~~ NFV architecture:

① Virtualized network functions: These are software applications that deliver network functions like file sharing, directly services & IP configuration.

② Network function virtualization infrastructure: It consists of the infrastructure components – compute, storage, networking.

③ Management, automation and network orchestration: It provides a framework for managing NFV and VNF's.

Benefits of using NFV: With NFV, service providers can run network functions, on standard hardware, instead of dedicated hardware. NVF gives providers the flexibility to run ~~NVFs~~ VNFs across different servers or move them around as needed when demand changes. This flexibility lets service providers deliver services and app faster. For example, if a customer request a new network function, they can spin up a new VM to handle that request. If the function is no longer needed, the VM can be decomissioned. this can also be a low risk way to test the value of a potential new service.

(c) Service Function Chaining

It uses the capabilities of SDN to create a chain of connected network services such as 24-7 services like firewalls. Network Address Translation (NAT) and intrusion protection. Network operators can use this to set up suites or catalogs of pathways for traffic to travel through. Any one path can consist of any combination of connected services, depending on the traffic's requirements. Different traffic requirements might be more security, lower latency, or an overall high quality of service(QoS) The primary advantage of network service chaining is to automate the way virtual network connections can be setup to handle traffic flows for connected services. For example, an SDN controller could take a chain of services and apply them to different traffic flows depending on the source, destination or type of traffic.

The chaining capability automates what traditional network administrators do when they connect up a series of physical 24-7 devices to process incoming and outcoming network traffic which traditionally may require a no. of manual steps. Service chaining is being included in many SDN and network function virtualization use cases & deployments, including data centres.

Another benefit is when used with SDN, is optimizing the use of network resources and improving application performance. SDN analytics and performance tools can use the best available network resources and help negotiate around network congestion issues.

Service chaining is facing a new networking technology that is purposefully trying to get away from service chaining. That technology is called SASE.

The chain in service chaining represents the service that can be connected across the network using software provisioning. This is especially important in NFV world.

The technology's capabilities mean that a large number of virtual network functions can be connected together in a NFV environment. Because its done in software using virtual circuits, these connections can be set up and torn down as needed with service chain provisioning through the NFV.

Ans 14 TCP Incast Problem in Datacenters

The performance of the TCP protocol degrades when it is run in a ~~many~~ many-to-one traffic pattern, like in datacenters. This performance degradation of TCP is called TCP Incast Problem.

In a situation, where many senders concurrently transmit datablocks to a single receiver, a catastrophic collapse occurs and none of the senders is able to send their next block untill all senders finish transmitting. This delay is not at all desired.

Hence, in delay-sensitive online application which employ partition/aggregation computing, very senders are given time extension, so the system responses are excluded from the final result, due to the missing of aggregators deadline.

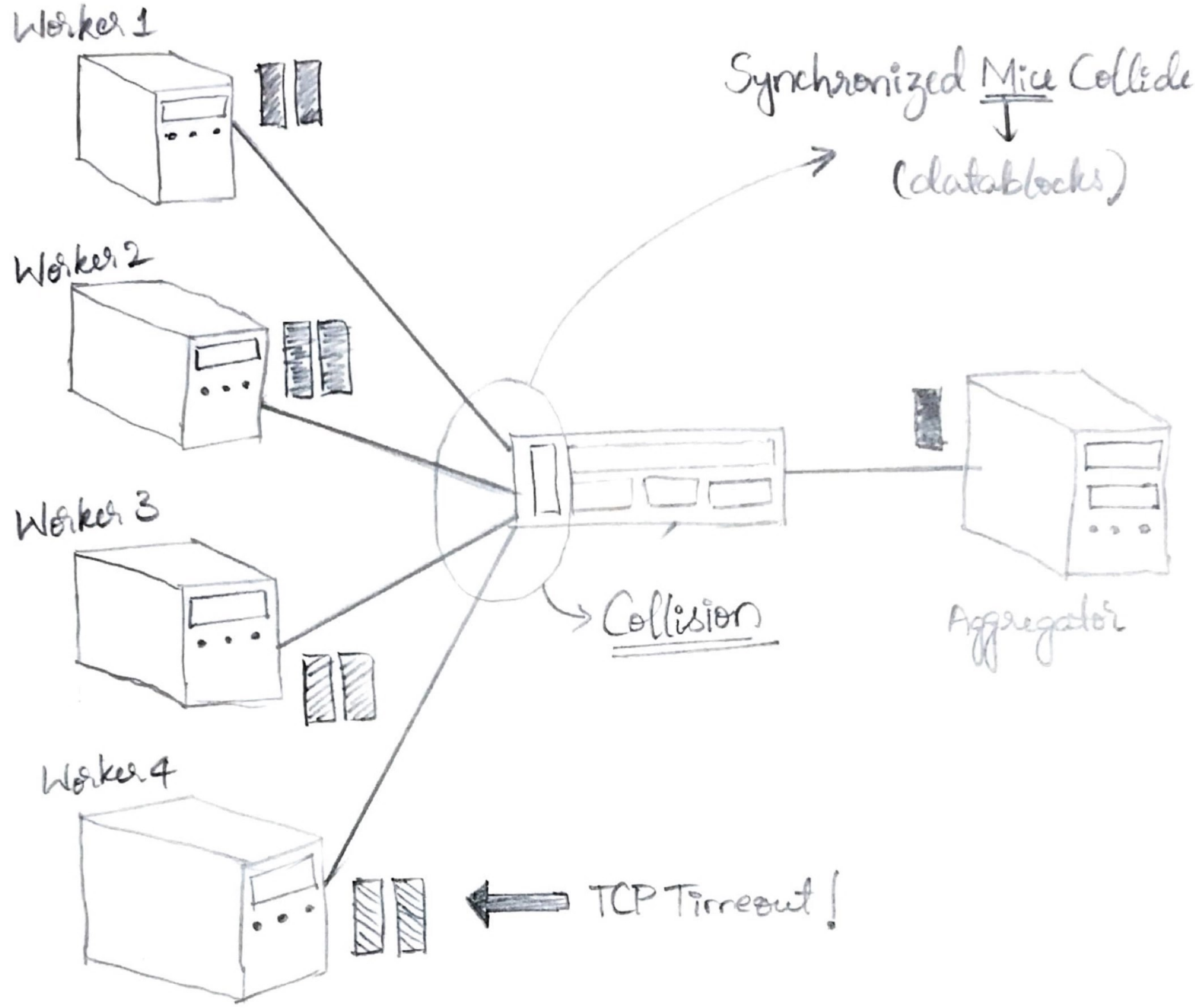
Solutions to solve TCP Incast Problem

① Adjusting System Parameters like changing block size, increasing buffer size, ~~and~~ increasing capacity, and limiting the no. of concurrent flows can help solve TCP Incast Problem.

Maheeth, 1801CS31, CS358, Maheeth

14.2

Diagram depicting TCP Incast Problem



(2) Designing enhanced mechanisms to reduce RTO_{min} will lead to reduction of throughput collapse. Another way is by decreasing the Max. transmission unit (MTU) [RTO is TCP retransmission timeout]

(3) Replacing loss-based TCP version: Adjusting congestion window properly according to the delay information generated by RTT measurement. Slight adjustment of window size is beneficial to buffer protection.

Maheeth, 1801CS31, CS358, Maheeth 14·3

(4) We can use the DCTCP algorithm. This is a TCP protocol variant that was designed especially for datacenters. It works because of the following characteristics

i) High Burst Tolerance

Because of large buffer headroom and aggresive masking.

ii) Low latency

There are small buffer occupancies ~~so to~~ so low queuing delay.

iii) High Throughput

Ans 15 **Markov Decision Process - MDP**

Markov Decision Processes are discrete time stochastic control processes. They provide a mathematical framework for modelling decision making in situations where outcomes are partly random and partly under the control of a decision maker. They are usually used for studying optimization problems utilizing Dynamic Programming.

MDP's are modelled in the following manner:

There is 4-tuple (S, A, P_a, R_a) in which

(i) S is a set of states called state space

(ii) A is a set of actions called action space.

(iii) $P_a(s, s') = P_r(S_{t+1} = s' \mid S_t = s, a_t = a)$

P_a is the probability that action a in state s at time t will lead to state s' at time t+1

(iv) $R_a(s, s')$ is the immediate reward (expected) received after transitioning from state s to s', due to action a.

NOTE: The state and action spaces may be finite/infinite

A policy function π is a probabilistic mapping from state space to action space.

Objective :

The objective of an MDP process is simply to find the optimal policy ($\pi(s)$) mapping from state space (S) to action space (A).

To do this, we use a function called value function,

$$V(S) = \sum_{S'} P_{\pi(S)}(S,S')\left(R_{\pi(S)}(S,S') + \gamma . V(S')\right)$$

This value function would return the expected returns of all rewards possible from state s. And, we would want a policy function which would maximise the expected return.

$$\pi_*(S) = \arg\max_a \left\{ \sum_{S'} P(S'|S,a)\left(R(S'|S,a) + \gamma . V(S')\right)\right\}$$

We can converge to this optimal policy by iterating through 2-steps. (i) Value Update (ii) Policy updates

These steps are repeated in some order for all states until no further change takes place. Both recursively update a new estimation of the optimal policy and state value using an older estimation of those values.

Example

There is a robot whose purpose is to pick up litter.

State Space $S = \{high, low\}$ for battery charge

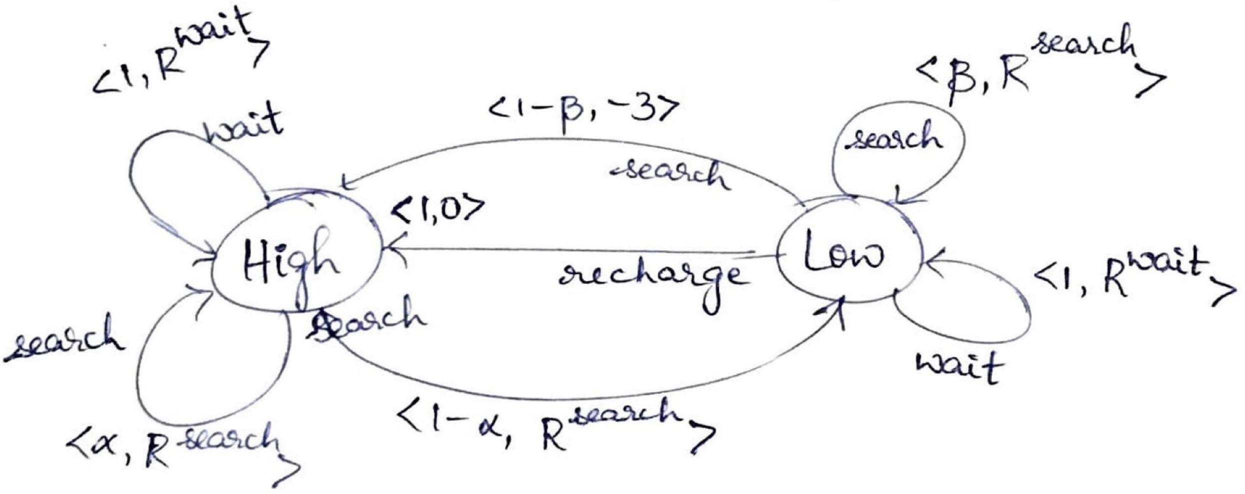
Action Space $A = \{wait, search, recharge\}$

wait → Hold still

search → Search for litter

recharge → Charge itself

We can model the MDP for this robot in the following manner, (each edge is represented as $<P, R>$ (P: probability of action, R: reward)



Q-Learning

Q-Learning is a model free reinforcement learning algorithm, to learn the value of an action in a particular state. It does not require a model of the environment and it can handle problems with stochastic transitions and rewards without requiring adaptations.

For any finite MDP, Q-learning finds an optimal policy in the sense of maximizing the expected value of the total reward over any and all successive steps, starting from the current state. Q-learning can identify an optimal action-selection policy for any given FMDP, given infinite exploration time and a partly-random policy.

NOTE: Q in Q-learning refers to the expected rewards for an action taken in a given table, which is computed by Q-learning.

Objective

Through this algorithm, we estimate the optimal Q-function.

We assume a Q table, mapping state action pairs to values. i.e., $Q : S \times A \longrightarrow R$

Before learning begins, Q is initialized to a possibly arbitrary value. Then at each time 't', the agent selects an action a_t, observes a reward r_t, enters a new state s_{t+1} and Q is updated.

$$\underbrace{Q'(s_t, a_t)}_{\text{new value}} = \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \Big(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{optimal future value(expected)}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \Big)\Big)$$

Three factors are involved here:

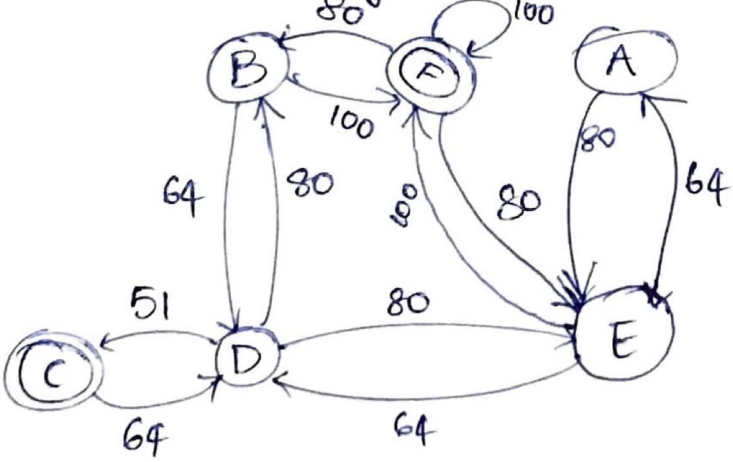
(i) $(1-\alpha)\, Q(s_t, a_t)$ → The current value weighed by learning scale.

(ii) αr_t → Reward if action a_t is taken when in s_t

(iii) $\alpha \gamma \max_a Q(s_{t+1}, a)$ → maximum reward that could be obtained from s_{t+1}

Example,

Consider the following situation



Maheeth, 1801CS31, CS358, Maheeth

15·6

Initial Q-table,

$$Q = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Final Q-table

$$Q = \begin{bmatrix} - & - & - & - & 80 & - \\ - & - & - & 64 & - & 100 \\ - & - & - & 64 & - & - \\ - & 80 & 51 & - & 80 & - \\ 64 & - & - & 64 & - & 100 \\ - & 80 & - & - & 80 & 100 \end{bmatrix}$$

Differences b/n MDP and Q-Learning:

	Markov Decision Process	Q-Learning
1	Discrete Time Stochastic Control Process	It is a continuous time stochastic control process
2	It is policy/model based algorithm	It is a model free algorithm
3	It utilizes dynamic programming.	It utilizes Bellman's equation
4	It is a non-temporal difference problem	It is a temporal difference problem.
5	Uses value & policy functions for optimization	Q-Table is used instead.
6	Used if R, P are known before hand	Used when R, P can't be predicted

Ans 15 Markov Decision Problem - MDP vs Q-Learning

① MDP performs better than Q-Learning in non-temporal environments: when the episodes end at a time stamp.

② MDP doesn't suffer from bias as each update is made using a true sample of what $Q(s,a)$ should be.

Ans 16

Exploration and Exploitation in Q-Learning

The Q-Learning algorithm does not specify what the agent should actually do. The agent learns a Q-function that can be used to determine an optional action. These are two things that are useful for the agent to do:

Exploit: The knowledge that it has found for the current state S by doing one of the actions 'a' that maximises Q(s,a)

Explore: In order to build a better estimate of the optimal Q-function, it should select a different action from the one that it currently thinks is best.

Tradeoff

During learning, the agent might choose the action value with the highest estimate (greedy action) or something else (non-greedy). Doing only one of these doesn't help in convergence.

Balancing rightly between exploration & exploitation is the key. Only exploitation might be good in the short run, but might mess up the long run optimum. Therefore, even though initially exploration gives bad results it will help in convergence.

Some algorithms that balance these both are:

i) ϵ - greedy algorithm

ii) Upper confidence bound.

iii) Thompson sampling.

<u>Note:</u> The property that needs to be held by exploration is <u>Degradation</u>. Exploration must happen in the initial stages of learning and must die down eventually. Otherwise, it would account to high variance.

<u>Eg:</u> Decaying ϵ greedy policy

Ans17 The Q-Matrix is given in the question

Starting from 0, we need to find the highest Q-value from Q-Matrix until we reach 10.

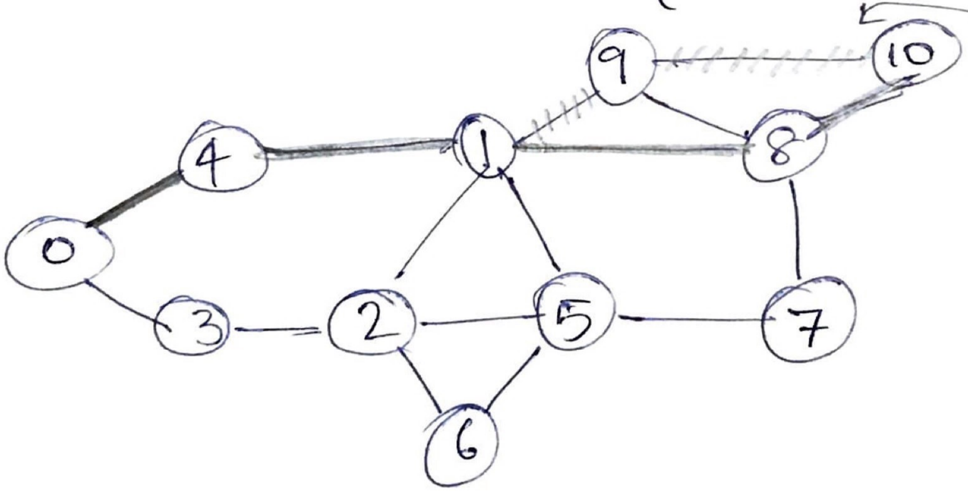
For 0, max Q-value is 138.0 to node 4

From 4, max Q-value is 174.0 to node 1

From 1, max Q-value is 218.0 to both nodes 8 and 9

From both 8&9, max Q-value is 274.0 to node 10.

So there are two shortest paths. $\begin{cases} 0 \rightarrow 4 \rightarrow 1 \rightarrow 8 \rightarrow 10 \\ 0 \rightarrow 4 \rightarrow 1 \rightarrow 9 \rightarrow 10 \end{cases}$



▬ → path 0→4→1→8→10

//// → path 0→4→1→9→10