Srishyla Educational Trust (R), Bheemasamudra

GM INSTITUTE OF TECHNOLOGY, DAVANGERE

DEPARTMENT OF COMPUTER SCIENCE &ENGINEERING

(Accredited by NBA, New Delhi valid till 30-06-2022)

with B++ Grade

FALCON Activities-Gallery 2019-20

Event Name: Hadoop installation and Fundamentals of Big data

Date: 13-03-2020

Event Coordinator: Mr. Rudresh N C, Asst. professor, Dept. of CSE

GM INSTITUTE OF TECHNOLOGY
HADOOP- INSTALLATION AND FUNDAMENTALS OF BIG-DATA

Inauguration of the workshop with the presence of principal, chief guest Dr. vinutha H P and HOD of the CSE

GM INSTITUTE OF TECHNOLOGY
HADOOP- INSTALLATION AND FUNDAMENTALS OF BIG-DATA

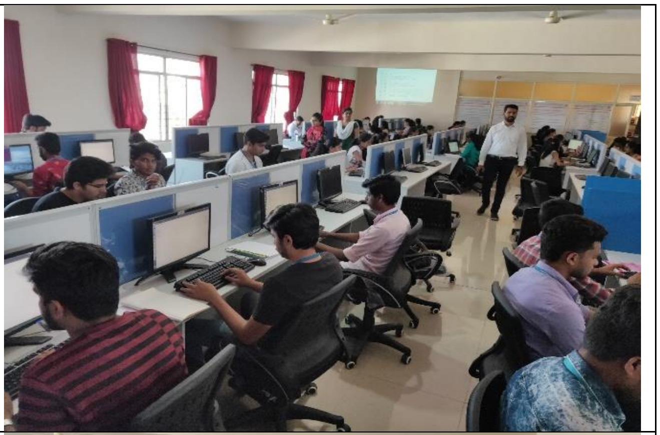
Mr. Rudresh N C, Asst. Professor dept. of CSE interaction with the student as a resource person in the session

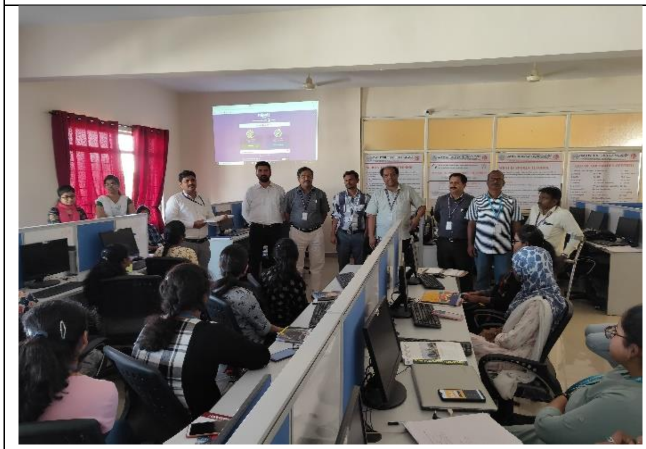
Group photo at the end of the workshop with HOD of CSE and all staffs and participants

Event Name: workshop on java4Beginners

Date: 18.02.2020 and 19.02.2020

Event Coordinator: Dr. Mouneshachari S and Mr. Veerabhadrappa
Assoc. Professor Instructor
Dept. of CSE Dept. of CSE





Event Name: Aptitude Test

Date: 14-02-2020

Event Coordinator: All staff of CSE

Event Name: Test on C-Skills

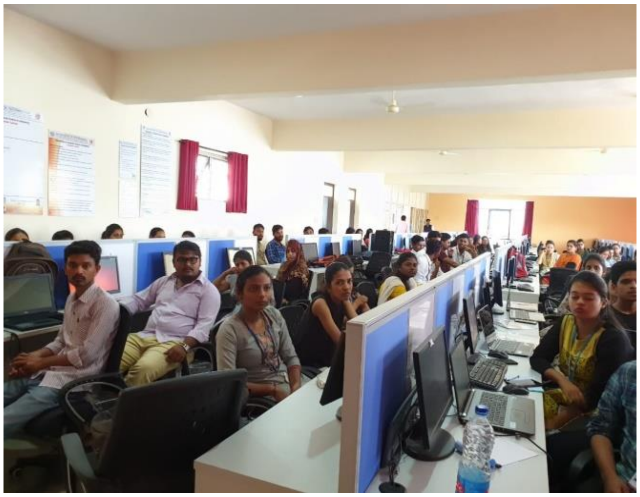
Date: 14.02.2020

Event Coordinator: All staff of CSE

Event Name: Two day hands-on Python for Everyone and Everywhere

Date: 7-11-2019 and 8-11-2019

Event Coordinator: Dr. Mouneshachari S and Mr. Veerabhadrappa
Assoc. Professor Instructor
Dept. of CSE Dept. of CSE



Event Name: DBMS Case Study design using ER diagram

Date: 9-09-2019

Event Coordinator: Mr. Kotreshi S N
Asst. Professor, Dept. of CSE

Event Name: Explore the PC

Date: 4-08-2019

Event Coordinator: Mr. Maruthi S T

Asst. Professor, Dept. of CSE

Principal Dr. P Prakash and Vice Principal Dr. B R Sreedhar during the Inauguration of "Explore the PC program".

Mr.Maruthi S T Explaining about the importance of hardware parts with demo during the program conduction.

Mr.Shreeganesh, 6th sem, CSE sharing his knowledge on computer and its hardware parts to the students.

Mr.Ojus V Tudvekar, 6th sem, CSE explaining about the charts and procedure of hardware lab conduction during the program conduction.

Group Photo with students and Ms.Archana, faculty coordinator of RSGG PU college, Davanagere

Group Photo with students and Ms.Akshatha, faculty coordinator of DRM PU college, Harihar

Event Name: Role plays in data structure and its applications

Date: 04-09-2019

Event Coordinator: Mr. Santoshkumar M

Asst. professor, Dept. of CSE

Activity: Poster Presentation on Analog & Digital Electronics

Date: 20-09-2019

Event Coordinator: Mr. Arunkumar B T

Asst. Professor, Dept. of CSE

DECODER
ENCODER
MULTIPLE

Description Language
Karnaugh Map

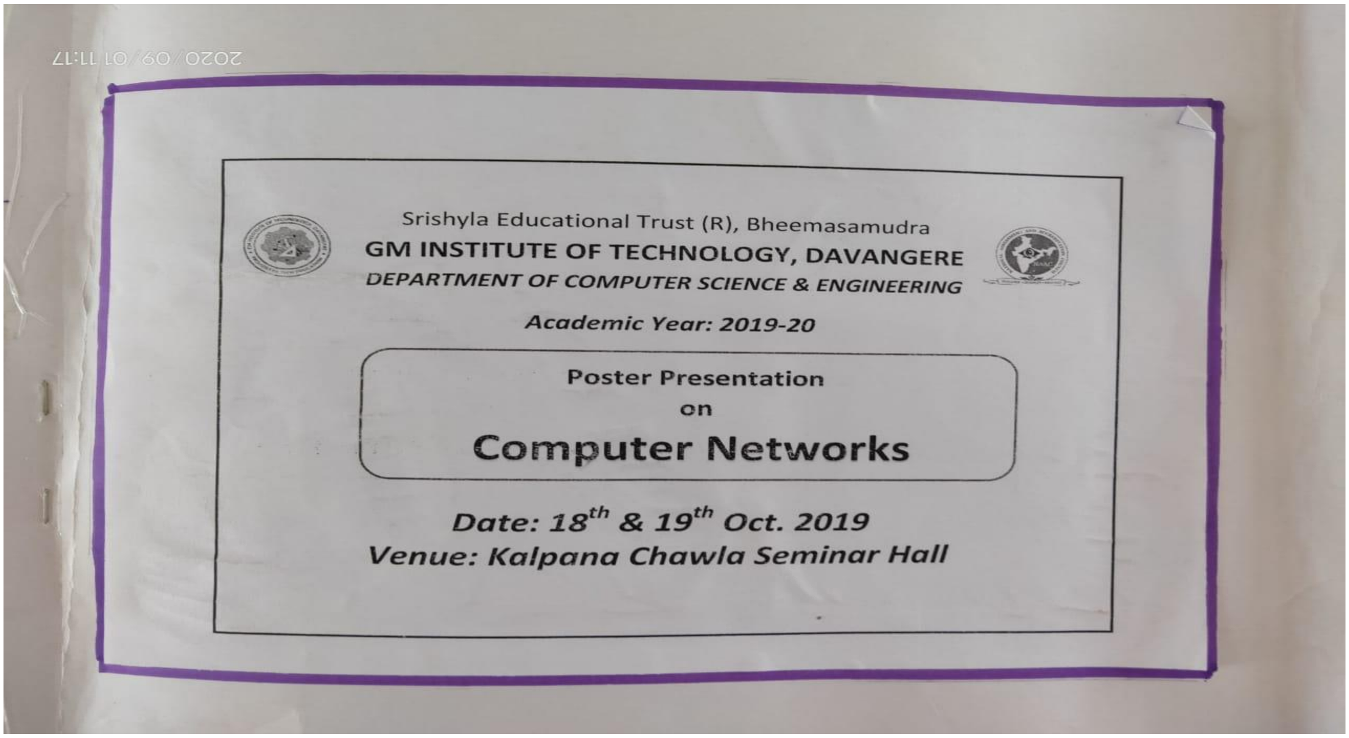
ANALOG AND DIGITAL ELECTRONICS
COMBINATIONAL
SEQUENTIONAL
PRODUCT OF SUM METHOD
HAZARD AND HAZARD COVER

HAZARD AND HAZARD COVER
SUM OF PRODUCT
QUINE MCCLUSKEY METHOD
ANALOG AND DIGITAL ELECTRONICS
MULTIPLEXER
ENCODER
DECODER

Placement Activity: Poster presentation on Computer Networks

Date: 18-10-2019 and 19-10-2019

Event Coordinator: Dr. Mouneshachari S
Assoc. Professor, Dept. of CSE

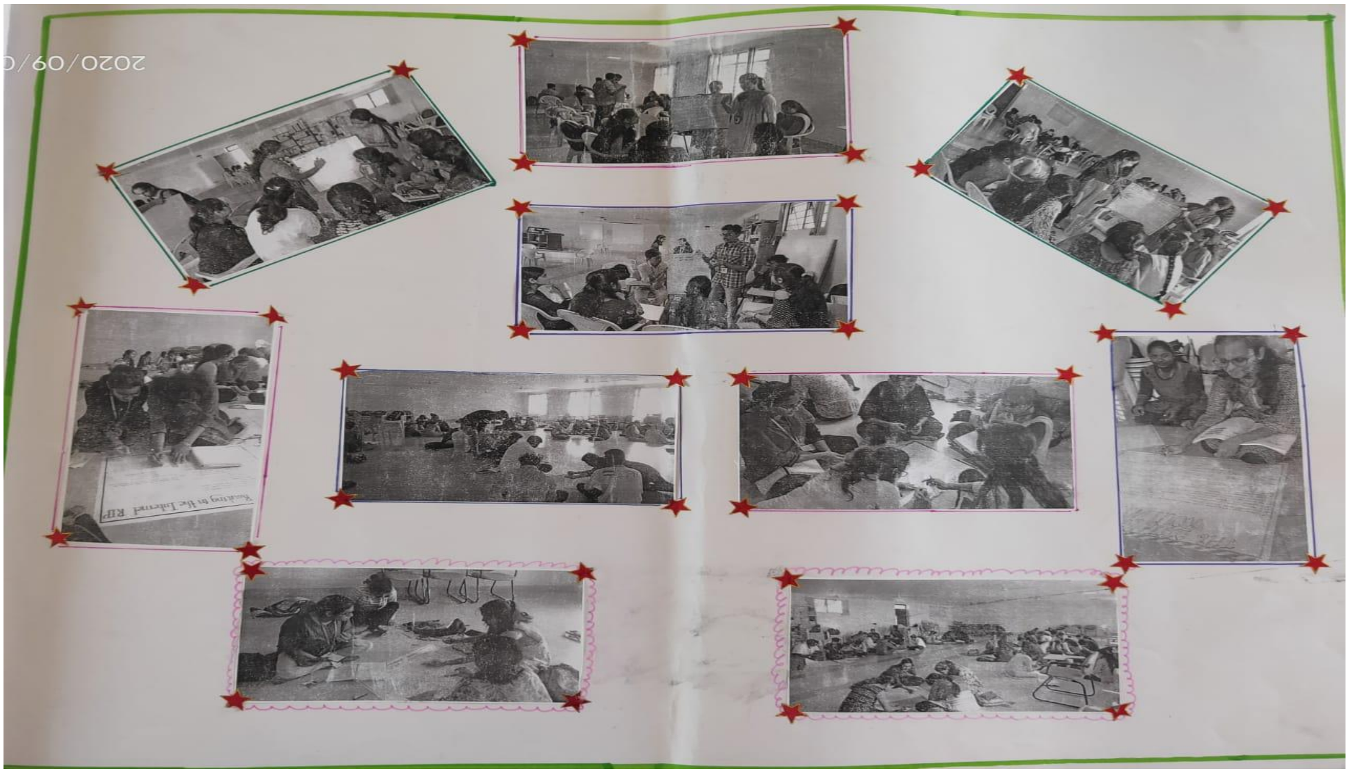


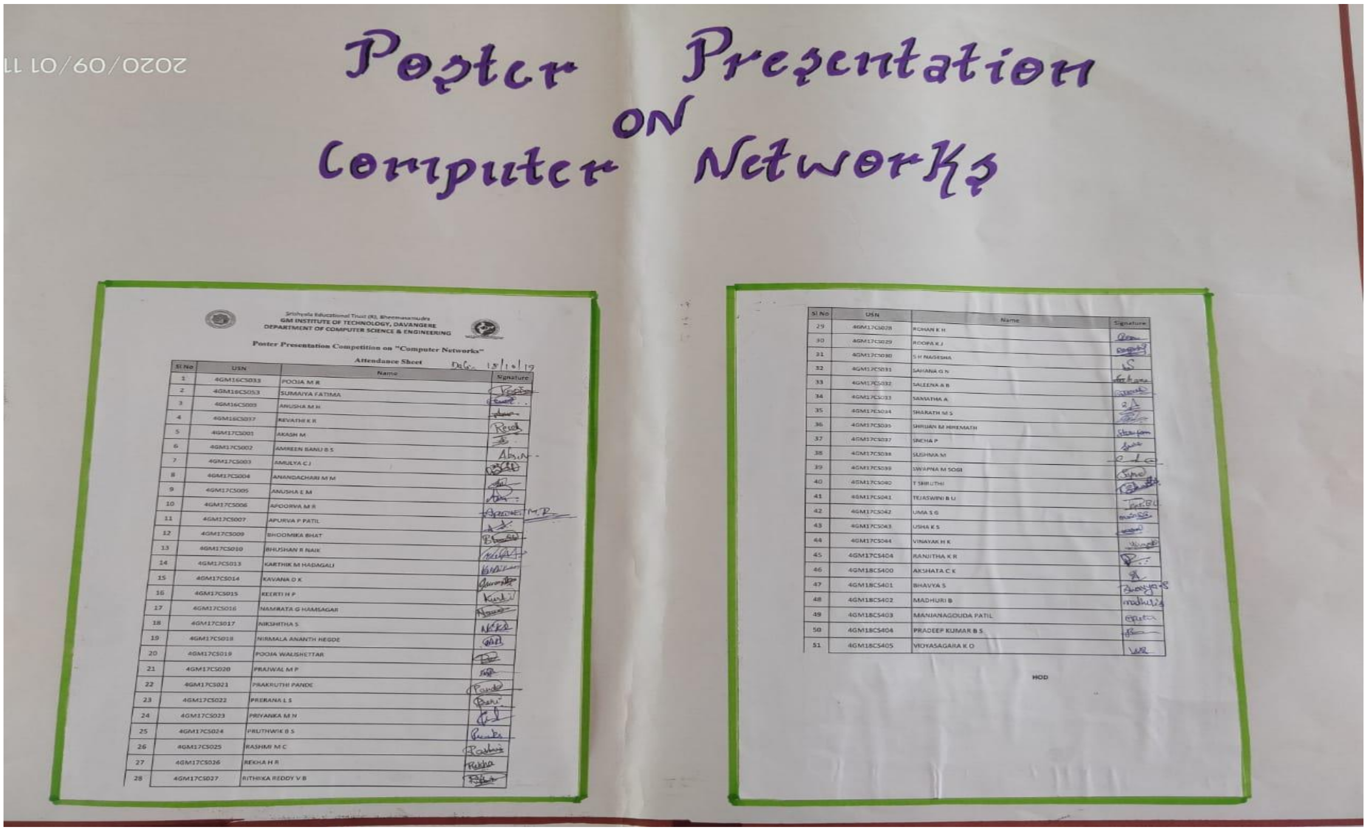
Srishyla Educational Trust (R), Bheemasamudra
GM INSTITUTE OF TECHNOLOGY, DAVANGERE
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Academic Year: 2019-20

Poster Presentation
on
Computer Networks

Date: 18th & 19th Oct. 2019
Venue: Kalpana Chawla Seminar Hall





Poster Presentation ON Computer Networks

Srishyla Educational Trust (R), Bheemasamudra
GM INSTITUTE OF TECHNOLOGY, DAVANGERE
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

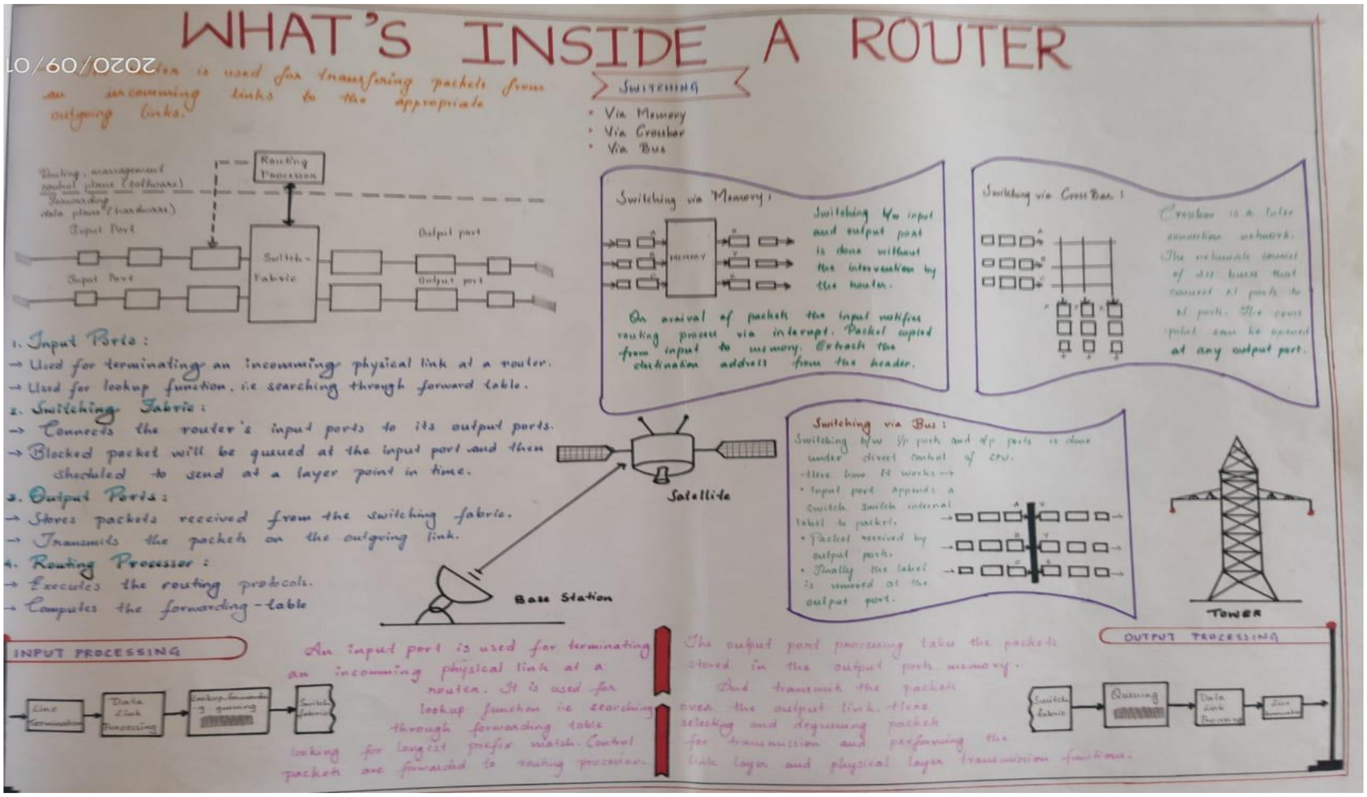
Poster Presentation Competition on "Computer Networks"

Attendance Sheet

Date: 18/10/19

Sl No	USN	Name	Signature
1	4GM16CS033	POOJA M R	
2	4GM16CS053	SUMAIYA FATIMA	
3	4GM16CS003	ANUSHA M H	
4	4GM16CS037	REVATHI K R	
5	4GM17CS001	AKASH M	
6	4GM17CS002	AMREEN BANU B S	
7	4GM17CS003	AMULYA C J	Absent
8	4GM17CS004	ANANDACHARI M M	
9	4GM17CS005	ANUSHA E M	
10	4GM17CS006	APOORVA M R	Apoorva M.R
11	4GM17CS007	APURVA P PATIL	
12	4GM17CS009	BHOOMIKA BHAT	
13	4GM17CS010	BHUSHAN R NAIK	
14	4GM17CS013	KARTHIK M HADAGALI	
15	4GM17CS014	KAVANA O K	
16	4GM17CS015	KEERTI H P	Keerti
17	4GM17CS016	NAMRATA G HAMSAGAR	
18	4GM17CS017	NIKSHITHA S	
19	4GM17CS018	NIRMALA ANANTH HEGDE	
20	4GM17CS019	POOJA WALISHETTAR	
21	4GM17CS020	PRAJWAL M P	
22	4GM17CS021	PRAKRUTHI PANDE	Pande
23	4GM17CS022	PRERANA L S	
24	4GM17CS023	PRIYANKA M N	
25	4GM17CS024	PRUTHWIK B S	
26	4GM17CS025	RASHMI M C	Rashmi
27	4GM17CS026	REKHA H R	Rekha
28	4GM17CS027	RITHIKA REDDY V B	
29	4GM17CS028	ROHAN K H	
30	4GM17CS029	ROOPA K J	
31	4GM17CS030	S H NAGESHA	
32	4GM17CS031	SAHANA G N	
33	4GM17CS032	SALEENA A B	
34	4GM17CS033	SAMATHA A	
35	4GM17CS034	SHARATH M S	
36	4GM17CS035	SHRIJAN M HIREMATH	
37	4GM17CS037	SNEHA P	
38	4GM17CS038	SUSHMA M	
39	4GM17CS039	SWAPNA M SOGI	
40	4GM17CS040	T SHRUTHI	
41	4GM17CS041	TEJASWINI B U	Tej.B U
42	4GM17CS042	UMA S G	
43	4GM17CS043	USHA K S	
44	4GM17CS044	VINAYAK H K	
45	4GM17CS404	RANJITHA K R	
46	4GM18CS400	AKSHATA C K	
47	4GM18CS401	BHAVYA S	Bhavya S
48	4GM18CS402	MADHURI B	madhuri B
49	4GM18CS403	MANJANAGOUDA PATIL	
50	4GM18CS404	PRADEEP KUMAR B S	
51	4GM18CS405	VIDYASAGARA K O	

HOD



WHAT'S INSIDE A ROUTER

[illegible] is used for transfering packets from an incomming links to the appropriate outgoing links.

Routing Processor
Routing, management control plane (software)
Forwarding data plane (hardware)
Input Port
Input Port
Switch Fabric
Output port
Output port

1. Input Ports :
→ Used for terminating an incomming physical link at a router.
→ Used for lookup function, i.e searching through forward table.
2. Switching Fabric :
→ Connects the router's input ports to its output ports.
→ Blocked packet will be queued at the input port and then sheduled to send at a layer point in time.
3. Output Ports :
→ Stores packets received from the switching fabric.
→ Transmits the packets on the outgoing link.
4. Routing Processor :
→ Executes the routing protocols.
→ Computes the forwarding-table

SWITCHING

- Via Memory
- Via Crossbar
- Via Bus

Switching via Memory:

Switching b/w input and output port is done without the intervention by the router.

On arival of packets the input notifies routing process via interupt. Packet copied from input to memory. Extract the destination address from the header.

Switching via CrossBar :

Crossbar is a inter connection network. The network consist of 2N buses that connect N ports to N ports. The cross point can be opened at any output port.

Switching via Bus :

Switching b/w I/p ports and o/p ports is done under direct control of CPU.

Here how it works →

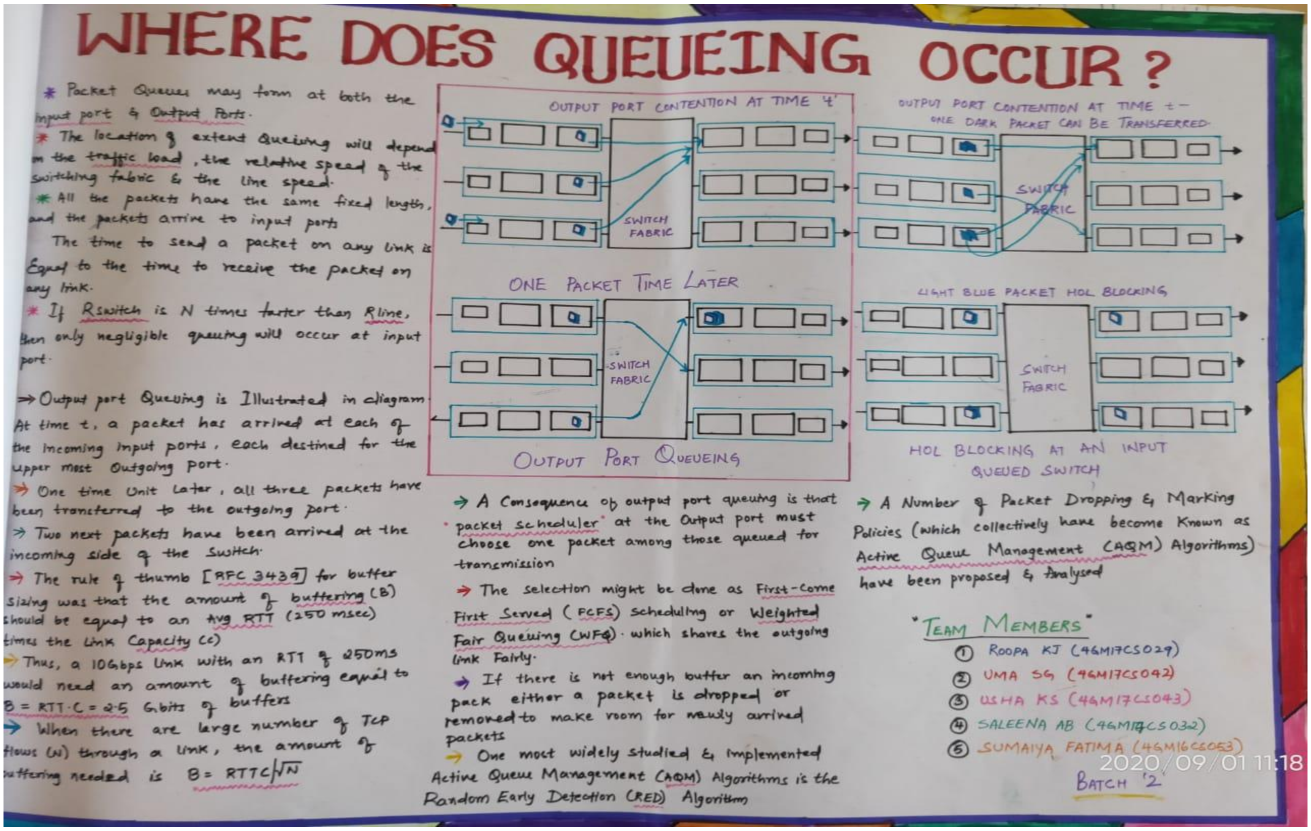
- Input port appends a switch switch internal label to packet.
- Packet received by output ports.
- Finally the label is removed at the output port.

INPUT PROCESSING

An input port is used for terminating an incomming physical link at a router. It is used for lookup function ie searching through forwarding table looking for longest prefix match. Control packets are forwarded to routing processor.

The output port processing take the packets stored in the output ports memory. And transmit the packets over the output link. Here selecting and dequeuing packets for transmission and performing the link layer and physical layer transmission functions.

OUTPUT PROCESSING



WHERE DOES QUEUEING OCCUR ?

* Packet Queues may form at both the input port & Output Ports.

* The location & extent Queuing will depend on the traffic load, the relative speed of the switching fabric & the line speed.

* All the packets have the same fixed length, and the packets arrive to input ports

The time to send a packet on any link is Equal to the time to receive the packet on any link.

* If Rswitch is N times faster than Rline, then only negligible queuing will occur at input port.

→ Output port Queuing is Illustrated in diagram. At time t, a packet has arrived at each of the incoming input ports, each destined for the upper most Outgoing port.

→ One time Unit Later, all three packets have been transferred to the outgoing port.

→ Two next packets have been arrived at the incoming side of the Switch.

→ The rule of thumb [RFC 3439] for buffer sizing was that the amount of buffering (B) should be equal to an Avg RTT (250 msec) times the Link Capacity (c)

→ Thus, a 10Gbps Link with an RTT of 250ms would need an amount of buffering equal to B = RTT·C = 2·5 Gbits of buffers

→ When there are large number of TCP flows (N) through a link, the amount of buffering needed is $B = RTTC/\sqrt{N}$

OUTPUT PORT CONTENTION AT TIME 't'

ONE PACKET TIME LATER

OUTPUT PORT QUEUEING

OUTPUT PORT CONTENTION AT TIME t — ONE DARK PACKET CAN BE TRANSFERRED

LIGHT BLUE PACKET HOL BLOCKING

HOL BLOCKING AT AN INPUT QUEUED SWITCH

→ A Consequence of output port queuing is that "packet scheduler" at the Output port must choose one packet among those queued for transmission

→ The selection might be done as First-Come First Served (FCFS) scheduling or Weighted Fair Queuing (WFQ). which shares the outgoing link Fairly.

→ If there is not enough buffer an incoming pack either a packet is dropped or removed to make room for newly arrived packets

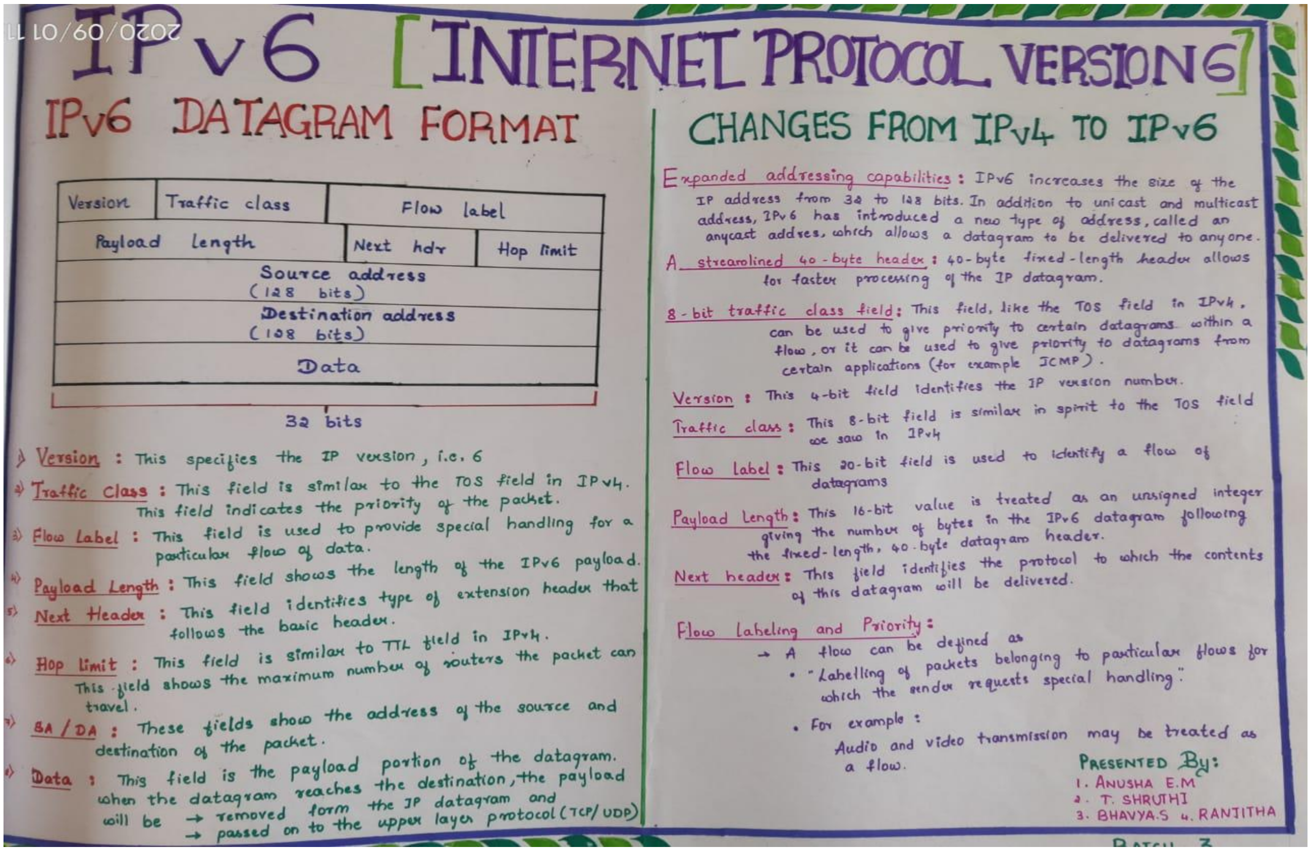
→ One most widely studied & implemented Active Queue Management (AQM) Algorithms is the Random Early Detection (RED) Algorithm

→ A Number of Packet Dropping & Marking Policies (which collectively have become Known as Active Queue Management (AQM) Algorithms) have been proposed & Analysed

"TEAM MEMBERS"

① ROOPA KJ (4GM17CS029)
② UMA SG (4GM17CS042)
③ USHA KS (4GM17CS043)
④ SALEENA AB (4GM17CS032)
⑤ SUMAIYA FATIMA (4GM16CS053)

BATCH '2'



IPv6 [INTERNET PROTOCOL VERSION 6]

IPv6 DATAGRAM FORMAT

<table>
<tr><td>Version</td><td>Traffic class</td><td colspan="2">Flow label</td></tr>
<tr><td colspan="2">Payload length</td><td>Next hdr</td><td>Hop limit</td></tr>
<tr><td colspan="4">Source address
(128 bits)</td></tr>
<tr><td colspan="4">Destination address
(108 bits)</td></tr>
<tr><td colspan="4">Data</td></tr>
</table>

32 bits

1) Version : This specifies the IP version, i.e. 6

2) Traffic Class : This field is similar to the TOS field in IPv4. This field indicates the priority of the packet.

3) Flow Label : This field is used to provide special handling for a particular flow of data.

4) Payload Length : This field shows the length of the IPv6 payload.

5) Next Header : This field identifies type of extension header that follows the basic header.

6) Hop limit : This field is similar to TTL field in IPv4. This field shows the maximum number of routers the packet can travel.

7) SA / DA : These fields show the address of the source and destination of the packet.

8) Data : This field is the payload portion of the datagram. when the datagram reaches the destination, the payload will be → removed form the IP datagram and → passed on to the upper layer protocol (TCP/UDP)

CHANGES FROM IPv4 TO IPv6

Expanded addressing capabilities : IPv6 increases the size of the IP address from 32 to 128 bits. In addition to unicast and multicast address, IPv6 has introduced a new type of address, called an anycast addres, which allows a datagram to be delivered to anyone.

A streamlined 40-byte header : 40-byte fixed-length header allows for faster processing of the IP datagram.

8-bit traffic class field : This field, like the TOS field in IPv4, can be used to give priority to certain datagrams within a flow, or it can be used to give priority to datagrams from certain applications (for example ICMP).

Version : This 4-bit field identifies the IP version number.

Traffic class : This 8-bit field is similar in spirit to the TOS field we saw in IPv4

Flow label : This 20-bit field is used to identify a flow of datagrams

Payload Length : This 16-bit value is treated as an unsigned integer giving the number of bytes in the IPv6 datagram following the fixed-length, 40-byte datagram header.

Next header : This field identifies the protocol to which the contents of this datagram will be delivered.

Flow Labeling and Priority :

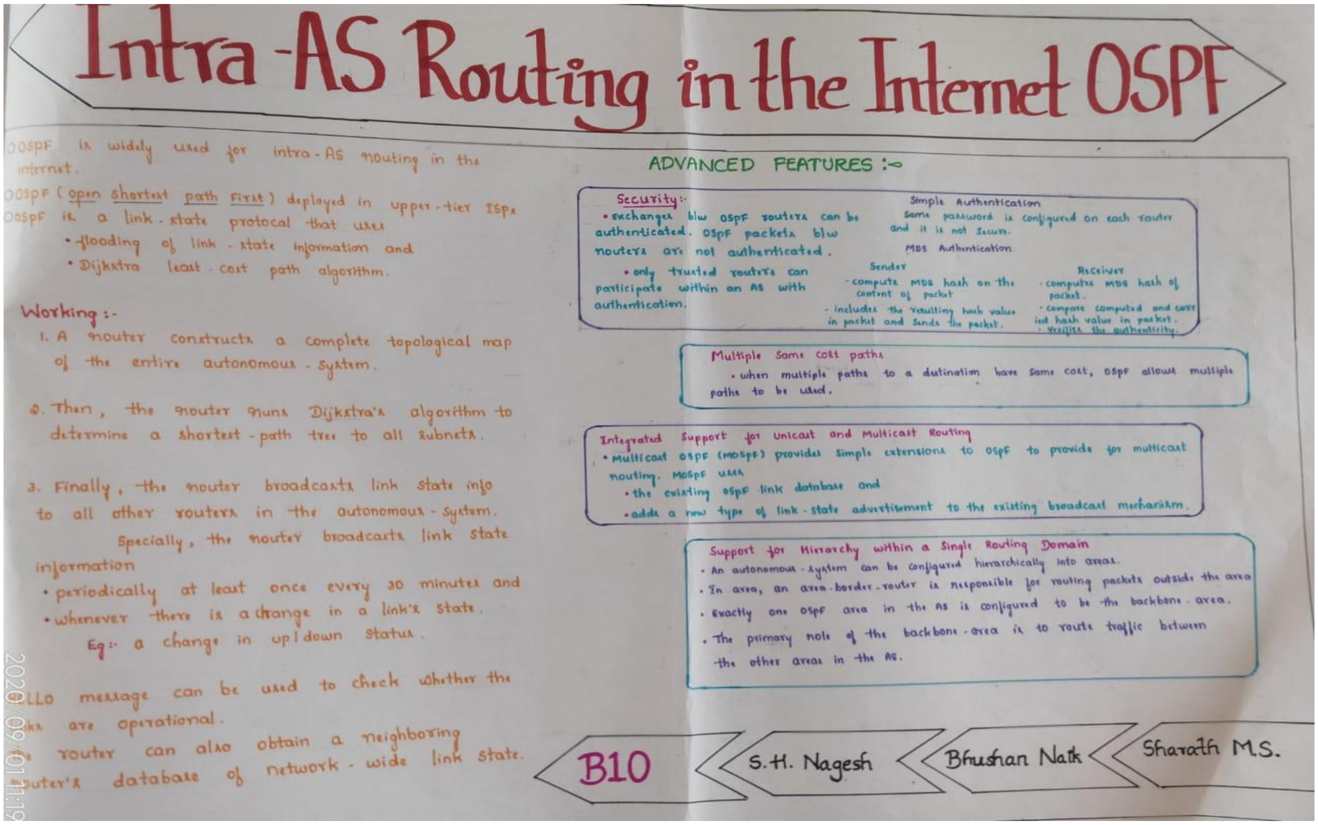
→ A flow can be defined as

- "Labelling of packets belonging to particular flows for which the sender requests special handling".
- For example : Audio and video transmission may be treated as a flow.

PRESENTED By:

1. ANUSHA E.M
2. T. SHRUTHI
3. BHAVYA.S 4. RANJITHA

BATCH 3



Intra-AS Routing in the Internet OSPF

- OSPF is widely used for intra-AS routing in the internet.
- OSPF (open shortest path First) deployed in upper-tier ISPs
- OSPF is a link-state protocal that uses
  - flooding of link-state information and
  - Dijkstra least-cost path algorithm.

Working :-

1. A router constructs a complete topological map of the entire autonomous-system.
2. Then, the router runs Dijkstra's algorithm to determine a shortest-path tree to all subnets.
3. Finally, the router broadcasts link state info to all other routers in the autonomous-system.

   Specially, the router broadcasts link state information
   - periodically at least once every 30 minutes and
   - whenever there is a change in a link's state.

     Eg:- a change in up/down status.

LLO message can be used to check whether the
ks are operational.
router can also obtain a neighboring
uter's database of network-wide link state.

ADVANCED FEATURES :-

Security:-

- exchanges b/w OSPF routers can be authenticated. OSPF packets b/w routers are not authenticated.
- only trusted routers can participate within an AS with authentication.

Simple Authentication

Same password is configured on each router and it is not secure.

MD5 Authentication

Sender	Receiver
- compute MD5 hash on the content of packet - includes the resulting hash value in packet and sends the packet.	- computes MD5 hash of packet. - compare computed and carried hash value in packet. - verifies the authenticity.

Multiple Same cost paths

- when multiple paths to a destination have same cost, OSPF allows multiple paths to be used.

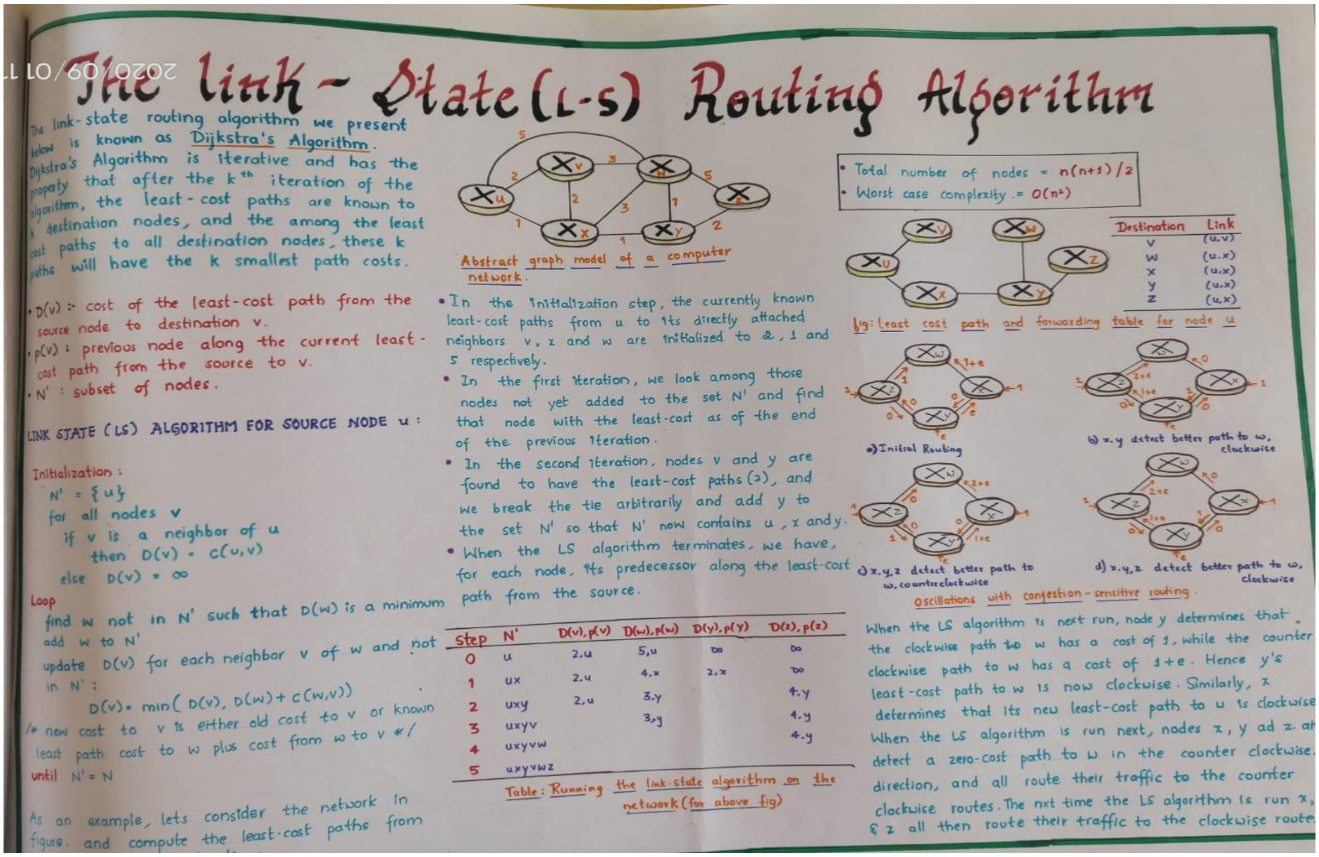
Integrated Support for Unicast and Multicast Routing

- Multicast OSPF (MOSPF) provides simple extensions to OSPF to provide for multicast routing. MOSPF uses
  - the existing OSPF link database and
  - adds a new type of link-state advertisement to the existing broadcast mechanism.

Support for Hierarchy within a Single Routing Domain

- An autonomous-system can be configured hierarchically into areas.
- In area, an area-border-router is responsible for routing packets outside the area
- Exactly one OSPF area in the AS is configured to be the backbone-area.
- The primary role of the backbone-area is to route traffic between the other areas in the AS.

B10 · S.H. Nagesh · Bhushan Naik · Sharath M.S.



The link-State (L-S) Routing Algorithm

The link-state routing algorithm we present below is known as Dijkstra's Algorithm. Dijkstra's Algorithm is iterative and has the property that after the k^{th} iteration of the algorithm, the least-cost paths are known to k destination nodes, and the among the least cost paths to all destination nodes, these k paths will have the k smallest path costs.

- D(v) :- cost of the least-cost path from the source node to destination v.
- p(v) : previous node along the current least-cost path from the source to v.
- N' : subset of nodes.

LINK STATE (LS) ALGORITHM FOR SOURCE NODE u :

```
Initialization :
  N' = {u}
  for all nodes v
    if v is a neighbor of u
      then D(v) = c(u,v)
    else D(v) = ∞
Loop
  find w not in N' such that D(w) is a minimum
  add w to N'
  update D(v) for each neighbor v of w and not
  in N' :
       D(v)= min( D(v), D(w)+ c(w,v))
/* new cost to v is either old cost to v or known
 least path cost to w plus cost from w to v */
until N' = N
```

As an example, lets consider the network in figure. and compute the least-cost paths from

Abstract graph model of a computer network.

- In the initialization step, the currently known least-cost paths from u to its directly attached neighbors v, x and w are initialized to 2, 1 and 5 respectively.
- In the first iteration, we look among those nodes not yet added to the set N' and find that node with the least-cost as of the end of the previous iteration.
- In the second iteration, nodes v and y are found to have the least-cost paths (2), and we break the tie arbitrarily and add y to the set N' so that N' now contains u, x and y.
- When the LS algorithm terminates, we have, for each node, its predecessor along the least-cost path from the source.

Step	N'	D(v),p(v)	D(w),p(w)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	∞	∞
1	ux	2,u	4,x	2,x	∞
2	uxy	2,u	3,y		4,y
3	uxyv		3,y		4,y
4	uxyvw				4,y
5	uxyvwz				

Table: Running the link-state algorithm on the network (for above fig)

- Total number of nodes = $n(n+1)/2$
- Worst case complexity = $O(n^2)$

Destination	Link
v	(u,v)
w	(u,x)
x	(u,x)
y	(u,x)
z	(u,x)

fig: least cost path and forwarding table for node u

1+e
1
0
0
e
1
a) Initial Routing
b) x,y detect better path to w, clockwise
2+e
1+e
c) x,y,z detect better path to w, counterclockwise
d) x,y,z detect better path to w, clockwise

Oscillations with congestion-sensitive routing.

When the LS algorithm is next run, node y determines that the clockwise path to w has a cost of 1, while the counter clockwise path to w has a cost of 1+e. Hence y's least-cost path to w is now clockwise. Similarly, x determines that its new least-cost path to w is clockwise. When the LS algorithm is run next, nodes x, y ad z. at detect a zero-cost path to w in the counter clockwise direction, and all route their traffic to the counter clockwise routes. The nxt time the LS algorithm is run x, y & z all then route their traffic to the clockwise route.