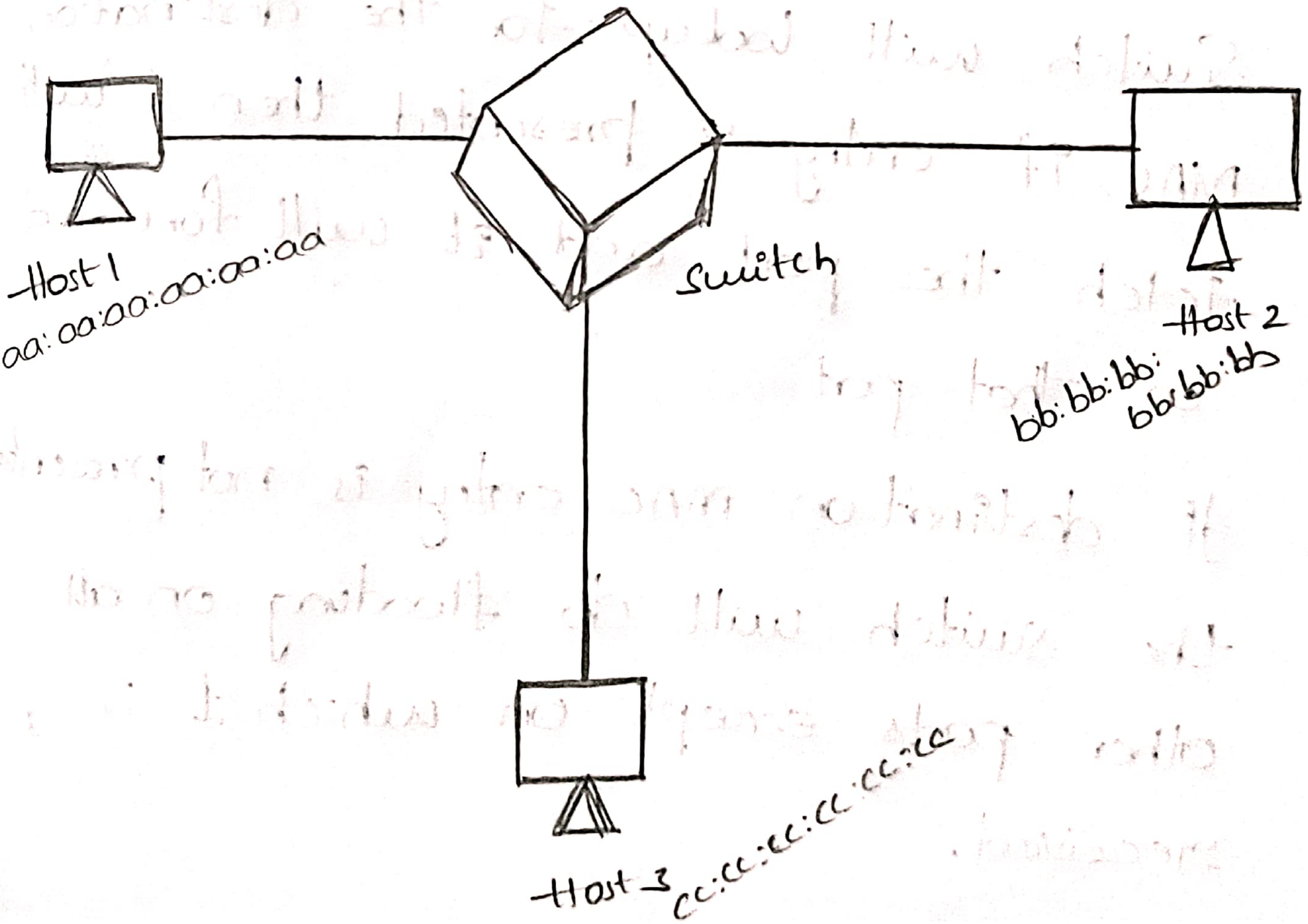
25/11/2024

Networks

Switch:-

Switch is a layer 2 device, which is used to communicate between the hosts. with in the network based on the MAC address.



MAC Table

MAC Address	port	type	vlan.

- It will check the source mac, is there any entry for this mac in the MAC Table, if not we need to add the entry.

- Switch will lookup for the destination MAC, if entry is presented then it will fetch the port and it will forward on that port

- If destination MAC entry is not presented the switch will do flooding on all other ports except on which it is received.

Broadcasting: source MAC is unicast,
destination MAC is ff:ff:ff:ff:ff:ff

flooding: Source MAC is unicast,
destination MAC is unicast.

- Each host, they will validate their system MAC with destination MAC of the frame

If it is not matched, host will discard (drop) the frame, if matched then it respond with reply.

- When mapped host is replying back he will keep source mac as host 2. and destination MAC is host 1. and Switch will receive reply frame on port 2.

- Switch will validate source MAC, in the MAC Table, if not existed adds into the MAC table.

Eg:-

MAC Address	port.	type	VLAN	
aa:aa:aa:aa:aa:aa	1	Dynamic	1	} default value
bb:bb:bb:bb:bb:bb	2	Dynamic	1	

Linux Commands

① less — View file content interactively

Eg:- less file

— It opens file in Viewer.

② more — View file content page-by-pag.

Eg:- more file

— Views file by paging.

③ head — Show the first few lines of a file

Eg:- head -n 5 file

— It shows first 5 lines

④ tail — Show the last few lines of a file

Eg:- tail -n 5 file

— It show last 5 lines.

⑤ find - Searches for files and directories

Eg. find Directory -name "*.txt"

- It searches for .txt files.

⑥ Locate - Find files by name (needs mlocate package)

Eg:- locate file

- It searches for 'file'.

⑦ du - Disk usage of files and directories

Eg:- du -h Directory

- It shows the size of that directory.

⑧ df — Display disk space usag

Eg:- df -h

— Disk usag in human-readable format

⑨ stat — Display detailed file or directory status.

Eg:- Stat file

— It shows status of 'file'

⑩ tree — Show directory structure in tree format.

Eg:- tree Directory-Name.

— It display that directory structure

C Programming

Pointers

In C, pointers are the variables that stores the address of another variable.

It allowing efficient manipulation of data and memory.

- A pointer is declared using * operator.

Eg:- int $\boxed{*\, ptr;}$. It stores the address of integer variable

- We can initialize the pointer with the address of another variable using the &(address-of) operator.

Eg:- int x = 10;

int *ptr = &x;

ptr → 5002 (address)

$x \rightarrow \boxed{10}$ (value)

- Dereferencing the pointers by using * (dere-ference) operator.

To access the value stored at the memory location:

Eg:- printf (" Value of x: %d \n", *ptr);

- Pointers are essential for managing dynamically allocated memory using functions like malloc, calloc, and free

Advantages

- Efficient array and structure manpulation
- Enables dynamically memory allocation.
- Allows functions to modify variables passed as arguments using their addresses.
- Facilitates the creation of data structures like linked lists, trees and graphs.

Types of pointers,

1. NULL Pointer:-

- A pointer that does not point to any valid memory location.
- It is initialized to NULL
- Useful for initializing pointers and error checking.

Eg:-
```
int *ptr = NULL
if (ptr == NULL)
{
    printf(" Pointer is NULL\n");
}
```

2. Void Pointer:-

- A pointer that can point to any data type.
- It is also known as generic pointer.
- Cannot be dereferenced directly. it must be type cast to the correct type.

Eg:-
```
int x = 10;
void *ptr = &x;
printf("Value is %d \n", *(int*)ptr);
```

If, we try to derefenrenced directly it shows an error.

3. Wild Pointer

A pointer that is not initialized and points to random memory location

- Using wild pointer can lead to undefined behaviour.

Eg:- int *ptr; → wild pointer

*ptr = 10; → undefined behaviour

To avoid these behaviour always initialize pointers before using them.

4. Dangling pointer:-

A pointer that points to a memory location that has been freed or deallocated.

- Accessing or dereferencing pointers leads to undefined behaviour.

Eg:- int *ptr = (int *) malloc (sizeof(int));

```
*ptr = 42;
free (ptr); → Memory freed.
ptr= NULL
   ↓
prevent dangling pointer
```