

4/4/22

Unit - 3

Sensors, Actuators And Sensor N/w

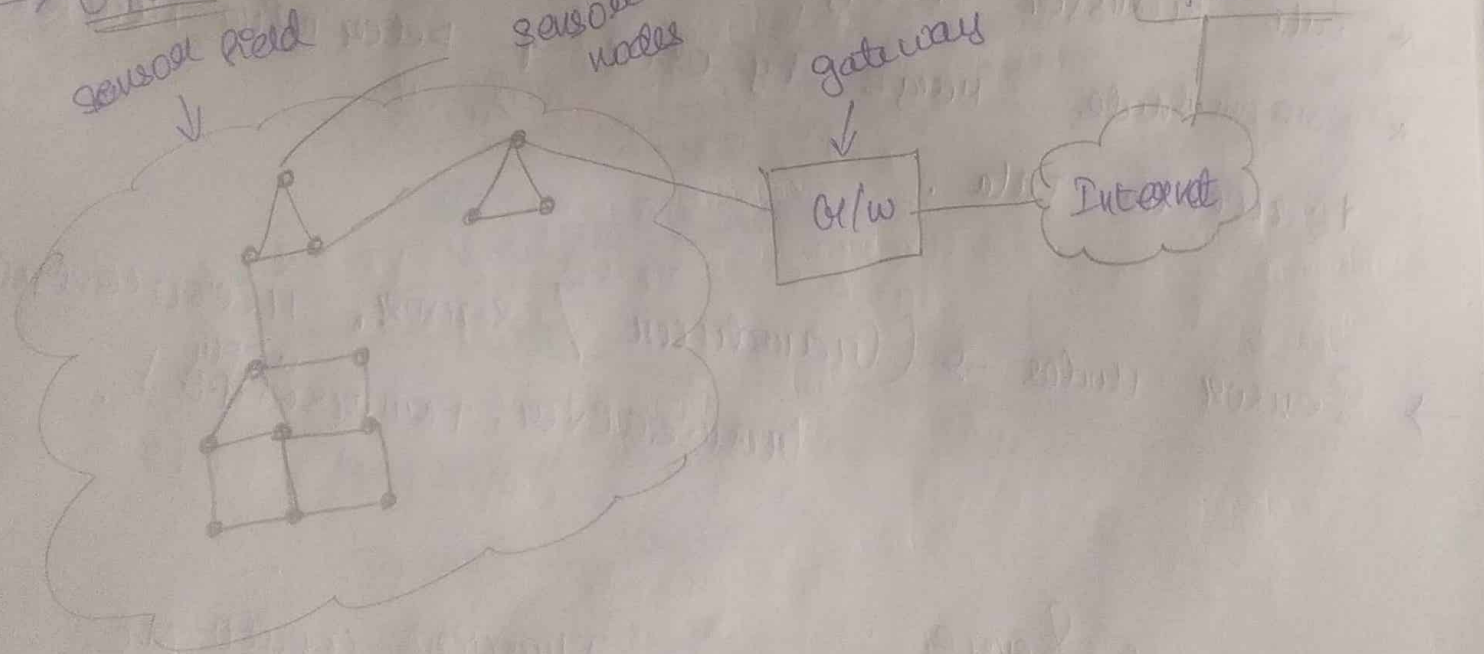
→ Definition of Sensor N/w :-

* Comprises group of small, powered device network infra structures, record conditions in any no of

Environment, including industrial facilities, farms, hospitals etc...

* Sensor network connect to the network to transfer information.

→ Operation of Sensor network :-



→ Sensor Network Include :-

- * Sensor / Actuator Node

- * Gateway & clients

- * sensor node group inside sensor field with diff topology.

→ Sensor Network Operation :-

- * Sensor nodes - monitor & collect data by sensor + transfer to other sensor node.

- * During data transfer data forwarded through multiple nodes to reach software.

- * data transfer through Internet to magnetic node.

- * management node merge by user + determine monitoring + collect data.

→ Sensor nodes → (Transducer / sensor, Microcontroller, transceiver, power supply).

⇒ Sensor :-

- * Sense physical condition & change into electrical signal.

- * signal sent to microcontroller for processing.

- * central processor send commands to be out of view
- * Data received

⇒ Self-driving car * RADAR sensor, Lidar

⇒ Network topology : *

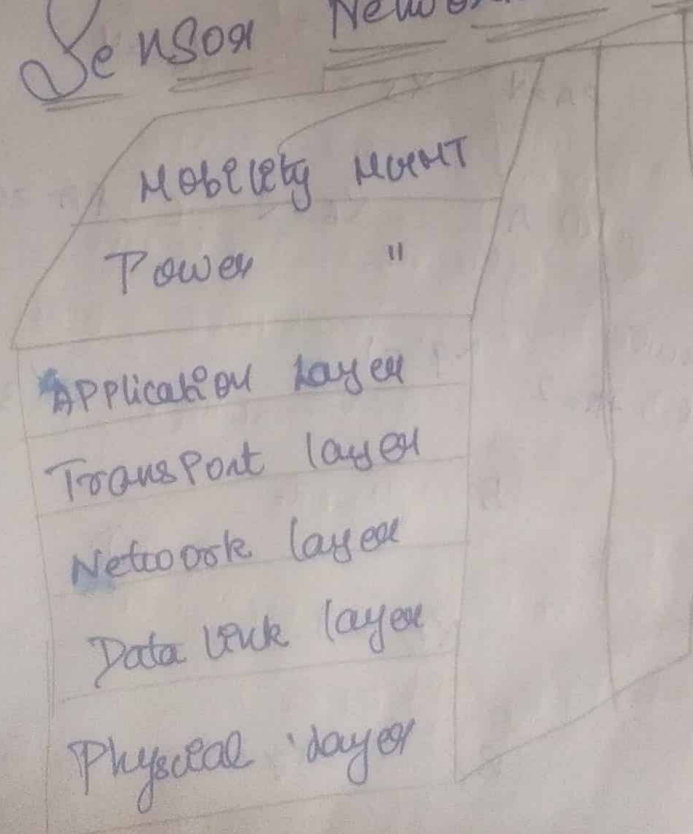
- * Point-to-point, star, mesh

⇒ Applications :

- * Automotive
- * Medical * Health care
- * Navigation
- * Aviation

11/1/22

Sensor Network Protocol Stack



→ moving one place to another place

HLR

→ Physical layer (collecting the info)

* Modulation (Modulator → Sensor)
 De-modulation → Receiver
 Optical → Light signal

→ Data link layer

* MAC (Media Access Control)

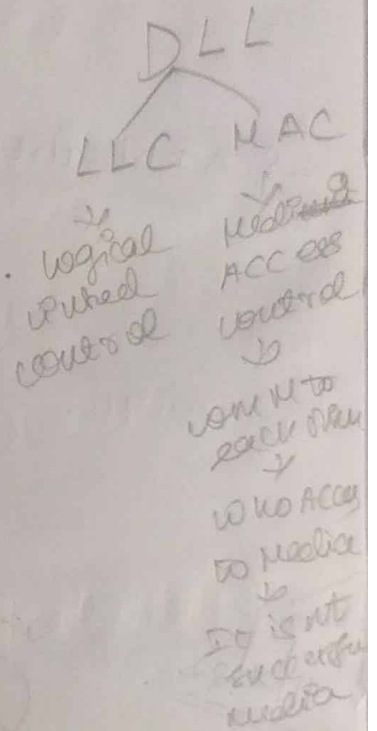
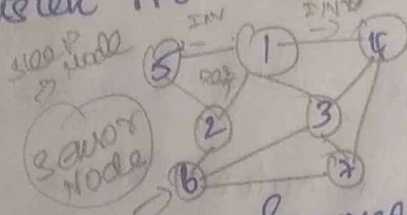
* TDMA (Time Division Multiple Access).

* NODE 1 wake up listen no response.

* Send Invitation

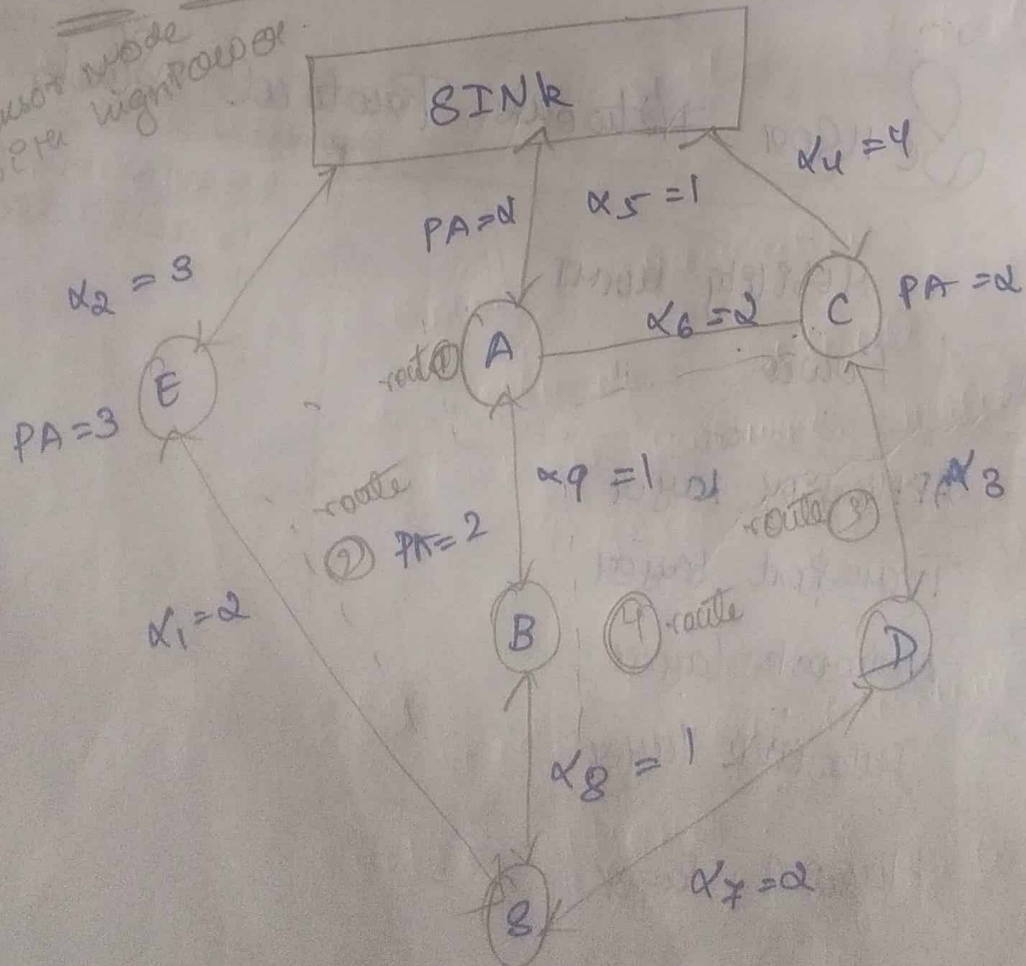
Node 1 → Node 2

Active Node Respond link formed.



→ Network Layer :- (No routing)

→ Sensor node with high power

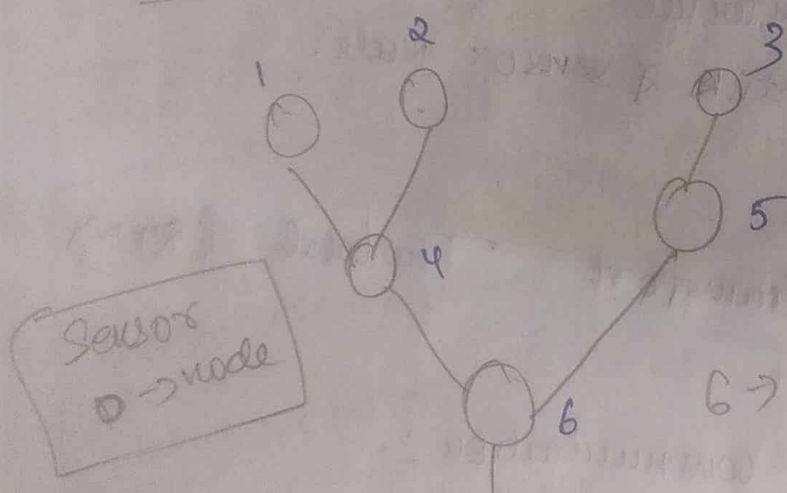


$x \rightarrow$ Distance
 $PA =$ Available Power

⇒ Route :- Shortest Path

- ① → Sink → A → B → S PA = 4 → 3
- ② → Sink → E → S PA = 3 → 5
- ③ → Sink → C → D → S PA = 3 → 7
- ④ → Sink → C → A → B → S PA = 6 → 8

⇒ Data Aggregation :- (Collection of Average)
(min, max, average)



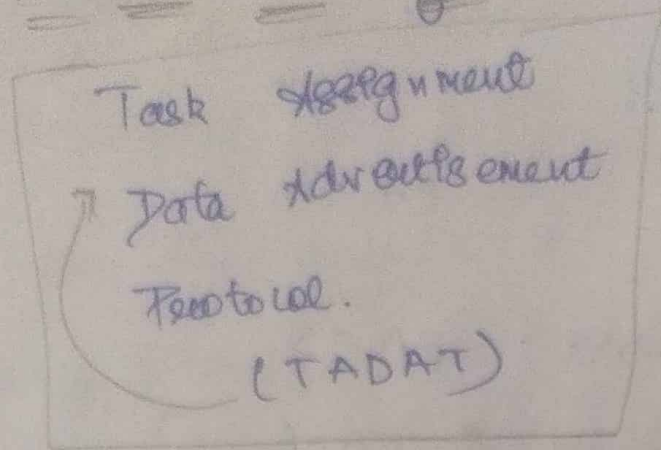
6 → Data Aggregation

Processing (Signal)

What Action To be done.

With Agg → It is called collection
all cluster → WPU do here process
from receiving
processing.

→ Application Layer:-



- * Sensor Management Protocol.
- * Multiple network Protocol.

→ Transport Layer:-

TCP → Internet

UDP → sink & sensor node.

12/7/22

→ Real time Transport Protocol (RTP)

⇒ Real time communication:-

- * Software Protocols.

- * Common hardware media.

→ gives real time guarantees

- * Eg:-

voice, over landline.

- * Mobile phones.

- * Voice Over Internet Protocol.

- * Instant messaging (whatsapp etc..
8/b mess...)

→ NO delivery Mechanism. connection loss.

* RTP \rightarrow SUPPLIES
EXPENSE HPE OR JPECH etc...
Address

- * sensitive to PK + Lact
- * less sensitive to PK + Lact

- * sensitive to PK + Lact
- * less sensitive to PK + Lact

* Version is 2-bit field using latest version

* P → It is 1 bit.

↓
It is used for padding (32 bit).

0000
↓
padding

* X → Extension header (1 bit)
↓
after a basic header.

0 → No extension header.

* Centricator count ⇒ 4-bit ⇒ 2^4
→ 0 to 15 possible (max)

* N ⇒ END marker by the APPL to indicator
the end of the data.

* Sequence N ⇒ It is used to rearrange the message.

* Payload → 7-bit
payload encoding

0 - 8A used
1 - 1016...

	PLM			
0	1016			
1		4	26	motion speed
3	GSM	4	30	H 216 4
5-6	DN 14		32	" "
4	CPL	"	33	" "
		"		
8	PCMA	"		
9	AC 22	"		
10-11	" 16	"		
14	MPED	"		
15	AC 22	"		

→ Timestamp: [32 bit]
 a relationship b/w Time & stamp of RTP
 Package.

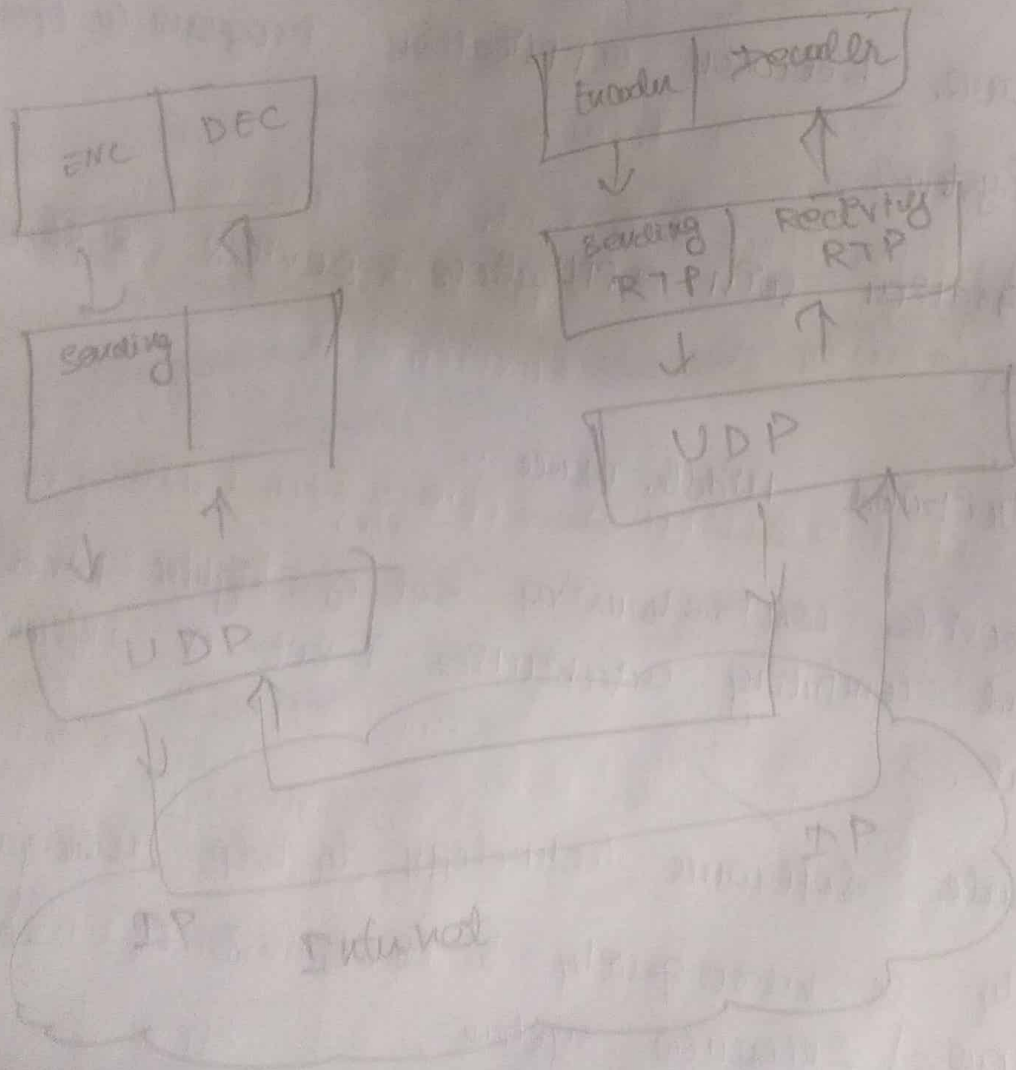
IP = random
 NEX = some of previous
 one's.

(32 bit)

→ Synchronization source
 * Defined source.
 * It is also random
 * Maximum 0-15

Use for where there is more than one source is present

→ RTP → use UDP!!



DEPT 2000
 819 50000 Block 1

RTP RTT
 Header Block 1

UDP
 RTT
 Header RTP Header Block 1

14/12/22



Middle Ware (MW) :-

* Software between application Program & operating Operating System.

* Integration join applications, Devices, system Software & Data. connecting

→ Distributed Middle Ware :- Each one have one work (distributed work)

* It Provide comprehensive set of higher level distributed computing capabilities & set of interface access that.

* It Include software technology to help manageable complexity or heterogeneity to Development of Distributed system / Interface system.

APPLN
NW
OS
H/W

* Higher level programming abstraction for developing distributed applications.

* Higher than lower level abstraction, such as sockets, monitors provided by operating system.

Transmission →
Read
Receive →
Write

Socket:- A common End Point
→ from which data can be read/write

→ Middle ware system view:-

OS → Software makes underlying hardware usable.

* Middleware system make "distributed system" programmable & manageable.

* Base microcontroller without operating system can be programmed.

* Programs written in assembly language.

→ Distributed Applications:-

* Can be developed without middle ware.

* But cumbersome.

→ Various Middle ware :-

* JVM Java virtual	* SOAP Simple object access protocol	* XML Extended machine language	* HTTP message queue in telephony training
* COAP Constrain application protocol	* IaaS Infrastructure as a service	* PaaS Platform as a service	* NOOS Network as a service

→ Distributed System :-

* NO. of Inter connected autonomous computer host provide services to meet the inference processing needs of Modern Enterprises.

* Connection of the Independent computer but appears to the user as single computing

→ Characteristics :-

① Multiple computers Each has CPU, memory, storage, I/O Path connect to Environment

② Inter connection → Some I/O Path but all connected talk to each other.

→ Shared state :- System cooperate to maintain shared state requires global invariant & coordinated operation of multiple computing

→ Reason for Distributed Computing :-

* Inherent Distribution

→ Bridge customer, supplier & companies at different site.

* Speed up :-

→ Improve Performance.

* Fault tolerance.

* Resource sharing

* Scalability.

* Flexibility.

→ Reason for Distributed System Hardware :-

* Scale :-

→ numerical, geographic, administrative

* Loss of control over path of the system.

* unreliability of message passing

* Failure.

→ Design Goals of Distributed System :-

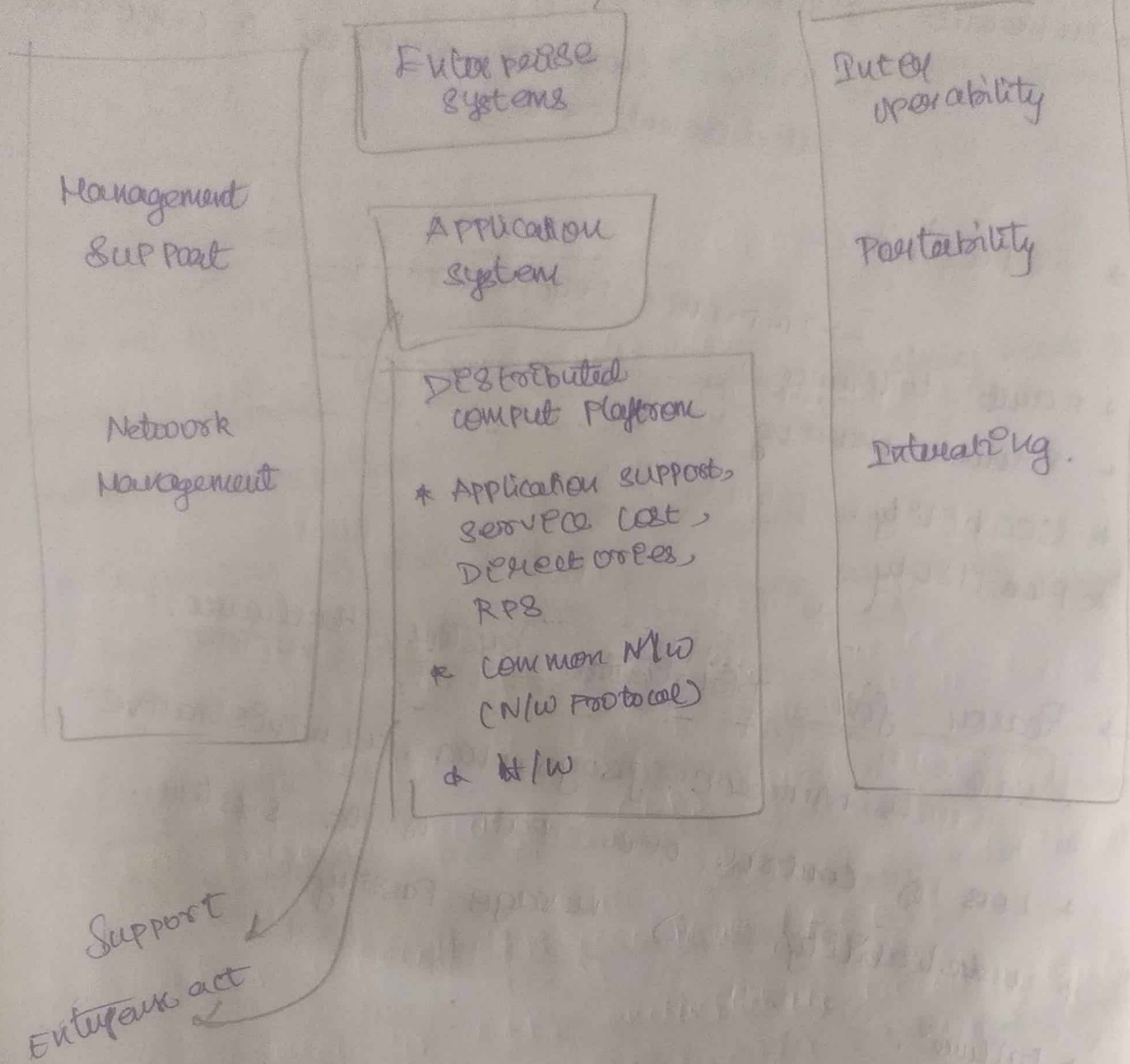
* Sharing :- Hardware, software, application services.

* openness :- extensibility.

* concurrency :- complete vs cooperative transport.

→ Distributed System :-

Platform Enterprise
activities.



Support
Enterprise act

16/1/22



Distributed Middle ware! -

Applications

Domain specific services

Common middle ware service

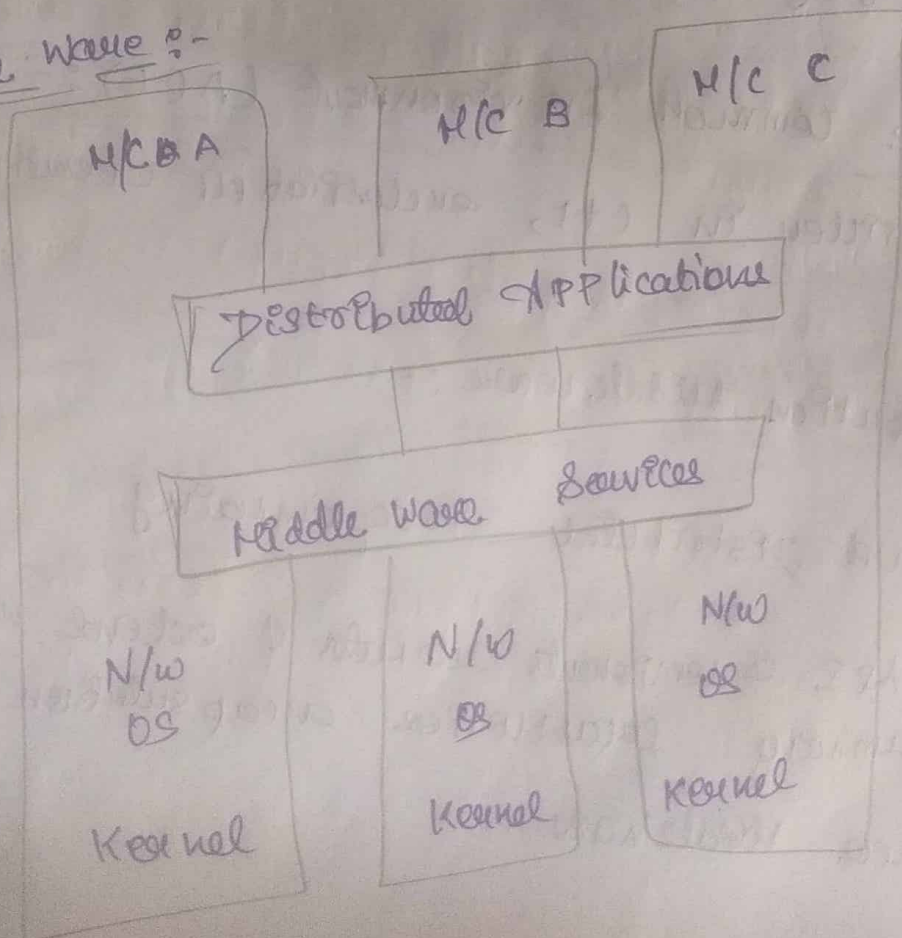
Distributed middle ware service

Host Infrastructure middle ware

OS & Protocols

Hardware devices

→ Middle ware :-



→ Host Infra structure :-

* encapsulate & enhance native OS & concurrently mechanism to create reusable.

* Networking component

reactors, monitor objects, acceptors, active objects.

→ Sun JVM → Platform Independent
→ execute code → by abstraction different OS &

CPU architecture.

→ NET :-
* Microsoft Platform XNL service.
* Designed to connect information devices
Common language Routine (CLR) - Middle ware.

→ Adaptive Common Environment (ACE) :-
written in C++, encapsulation operating system.

→ Distribution Middle ware :-

* higher level distributed programming

* Reusable API & components automate & extend native OS
N/w programming capabilities encapsulation by host

Infrastructure Middle ware.

→ OHCI CORBA :-

- * Common Object Request Broker Architecture
- * object + + component Integrate across middle ware.

→ Sun JAVA RMI :- (Remote Method Invocation)
* It is the methods Invoked from other JVMs.

→ Message PPS to PPUTEN Component Object Model CD COM :-
* software component communicate via Remote component.

→ SOAP ⇒ (Simple Object Access Protocol) :-
* It enable auto web services
* shared open web Infrastructure.

→ Common UW Services :-
* allow application Developer to contribute Program business logic without working codes to Develop Distributed system use low level middle ware Directly.

→ ORBI CORBK:- event notification logging, multimedia
streaming security, global RT, RT scheduling,
fault tolerance concurrency ENT & transaction

→ Sun Extended Java Bean:-
* Distributed system → prebuilt
Software source called Bean (not code from
scratch)

→ NET:-
Application specific into components Domain

→ Domain specific middle ware:-
* used for particular Domain such as telecom,
E-commerce, e-commerce, healthcare, process automation
Personal Hardware Mach devices...

→ ORBI:- (Object Mgmt group)

→ DTF:- (Domain Task forces)

→ E-commerce DTF:-

* development of E-commerce & electronic
marketing screens medical.

→ Smart Medical solution of saving:

* Medical Image include ultrasound, URF, CP SCAN etc...

18/1/22

- * collaborative signal processing
- * Data gathering
- * Time Dependent Systems
- * clock Synchronization

→ collaborative signal processing (CSP)

* Signal processing (SP) is an Electrical Engg sub field forces on analysing, modifying & Synthesise signals such as sound, image, scientific measurement

CSP & SP → process signals → to Extract info from that collaboration means work together

* CSP → process signal to extract information from that

→ collaboration : Means work together.

* CSP :- It is a process signal received from group of Element.

* Reasons - CSP :-

- * unity is strength.
- * goal → Identify, Detect & track any object.
- * Each sensor → process event for certain time
- * Detect & store → local info

→ IN Sensor Network:-

Elements → sensors

Input → location of target.

→ Time Dependent & Clock Sync:-

* Distributed System (DS) :-

It is a collection of computers via high speed common system.

* In Distributed system, Hardware & software components communicate, coordinate by message passing.

* Each node share information with other nodes.

* Need proper allocation of Resources & help coordinator between different processes.

* To resolve conflicts sync is used.

* Sync achieved in DS via clock.

* Time set based on universal time coordination (UTC) ▲

clk sync ways

- External clk sync
- Internal

9) External clk syn :-

External clock is present, used for reference for all nodes adjust the time accordingly.

10) Internal clk syn :-

* Each node has its own local time & share with other nodes.
* all other nodes set time the accordingly.

→ Two types of clock syn algorithm :-
① centralized
② Distributed

① centralized :-

* Time server used as reference.
* Single time server propagates its time to all the nodes.
* Single time systems → Node fails → It lose sync

Eg:- Berkeley algo, passive time service, Active time service etc.