# Basic Concepts

Prof. Ananthanarayana V.S.
Dept. of Information Technology
N.I.T.K., Surathkal

# Objectives of This Module

- This module gives evolution of Distributed Computing Systems (DCS) from hardware and software infrastructure point of view.
- This module also discusses what are the design goals, transparencies and fundamental issues in DCS
- Comparison of various Operating Systems from historical point of view is discussed

# Road Map

- Generic Component (H/W) of Distributed Computing System:
  - Computer system
  - Networking
- Evolution of operating systems
- Introduction to distributed computing systems (DCS)

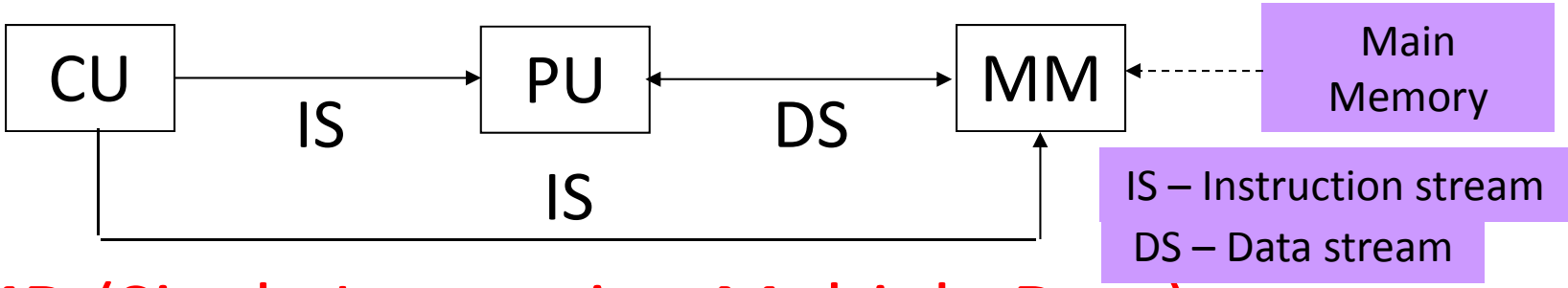
# Generic Components (H/W) of a DCS

- (1) Computer System
  - CISC
    - Complex instruction set computer
    - A single instruction performs multiple actions (hence complex in operation)
    - Ex: VAX, Intel® X86, IBM 360/370
  - RISC
    - Reduced instruction set computer
    - Each instruction performs only a simple operation
    - Ex: MIPS, DEC Alpha, SUN Sparc, IBM 801
  - Multi Core
    - More processors on chip, so multiple instructions can run at the same time
    - Intel Core Duo, Intel i3, i5, i7, AMD Phenom II X4

# Flynn's Classification of Computer System

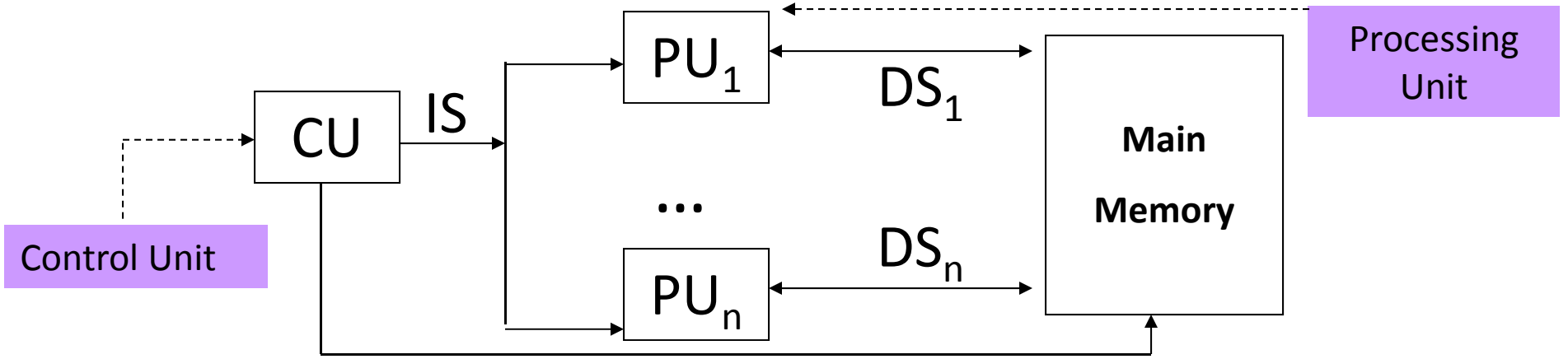
- SISD (Single Instruction Single Data)

**- General purpose computer**



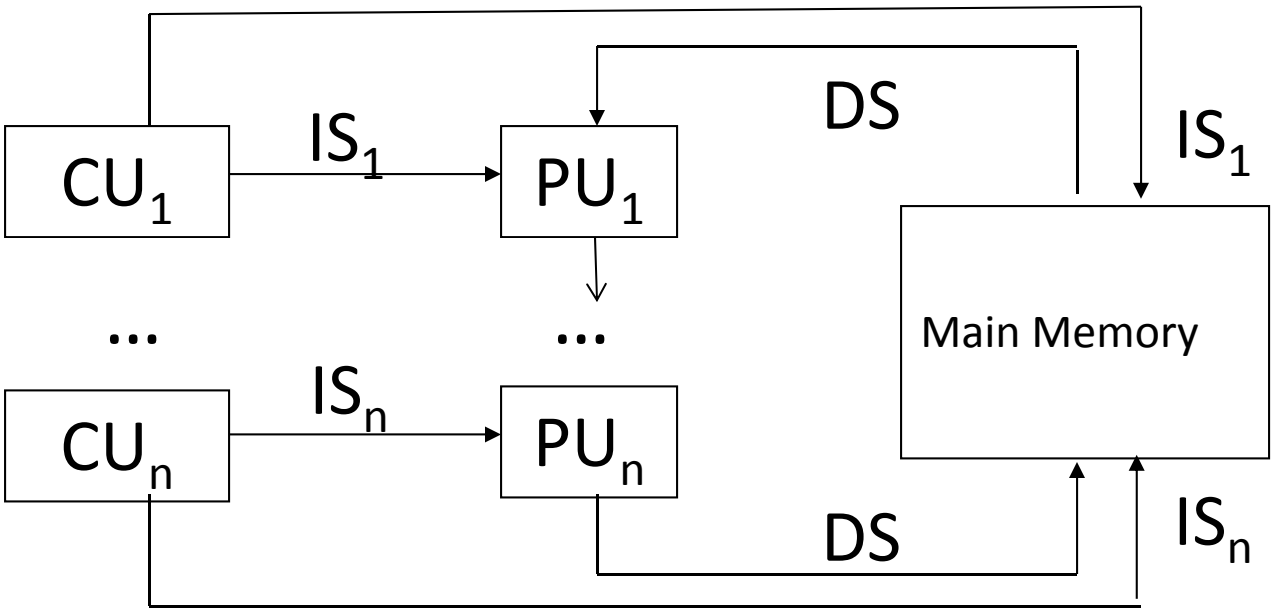

- SIMD (Single Instruction Multiple Data)

**-Vector processing computer**



# Flynn's Classification Contd…

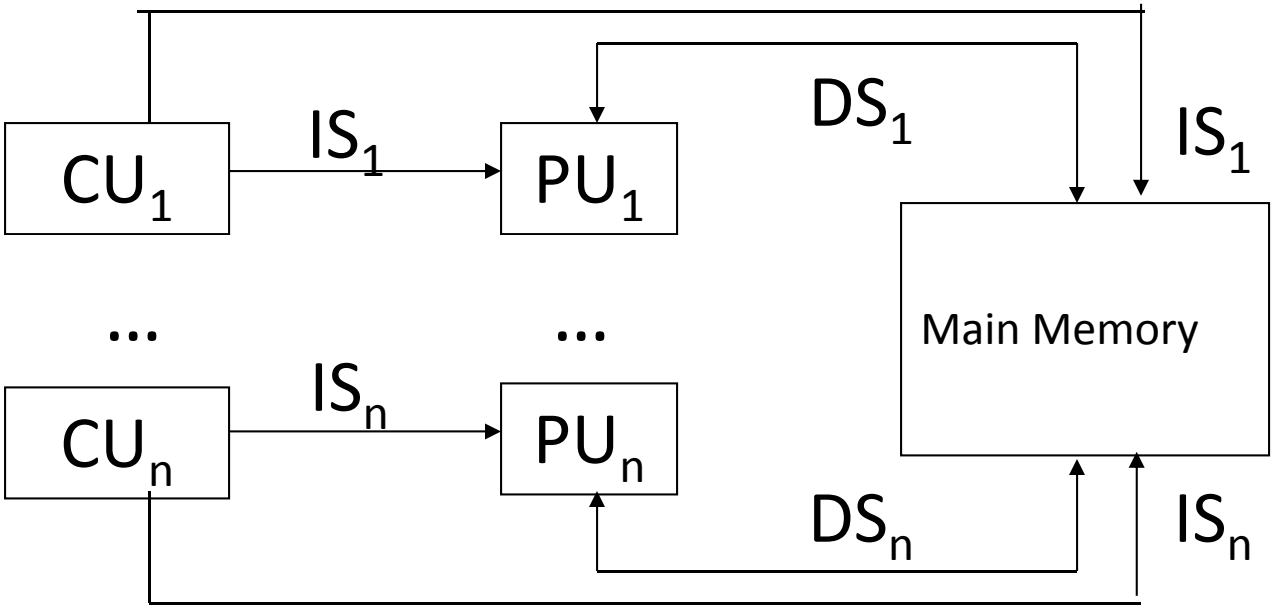
- MISD (Multiple Instruction Single Data)



# Flynn's Classification Contd…

- MIMD ( Multiple Instruction Multiple Data)

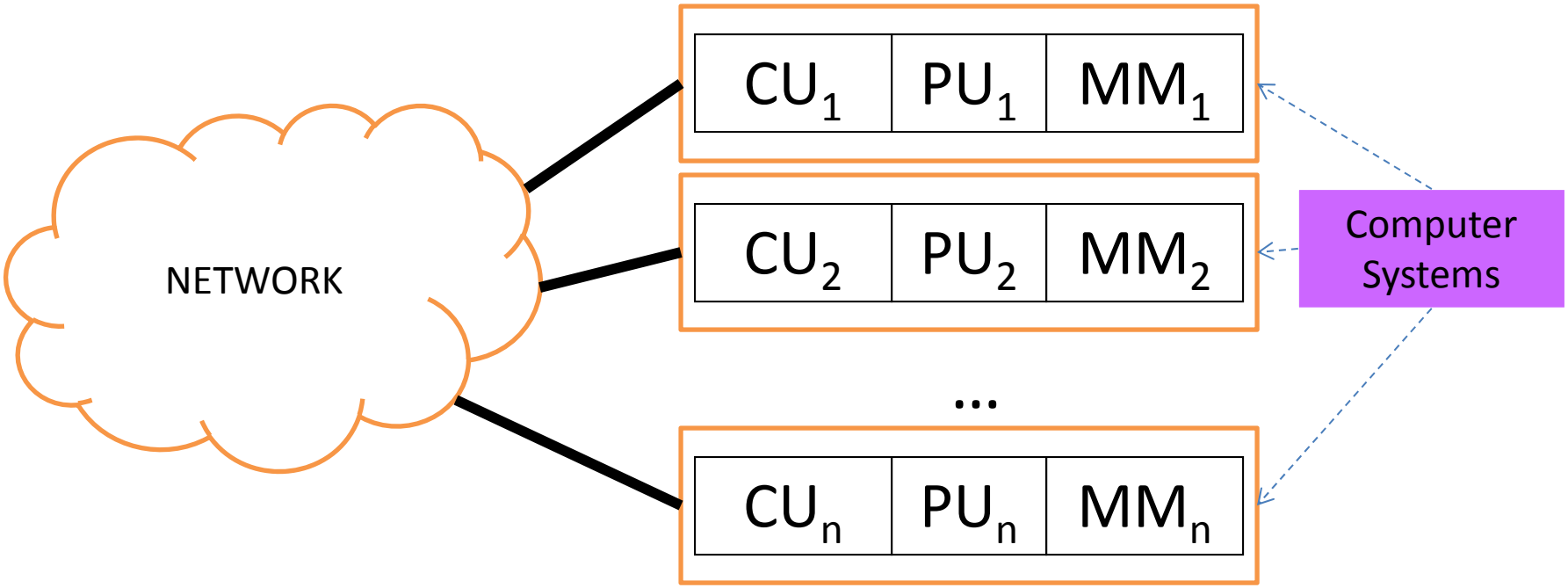
  **- Multi Processor with shared Memory**



- MIMD ( Multiple Instruction Multiple Data)

**- Multi Processor with non-shared memory**

**- Communication is by passing messages between them via network**



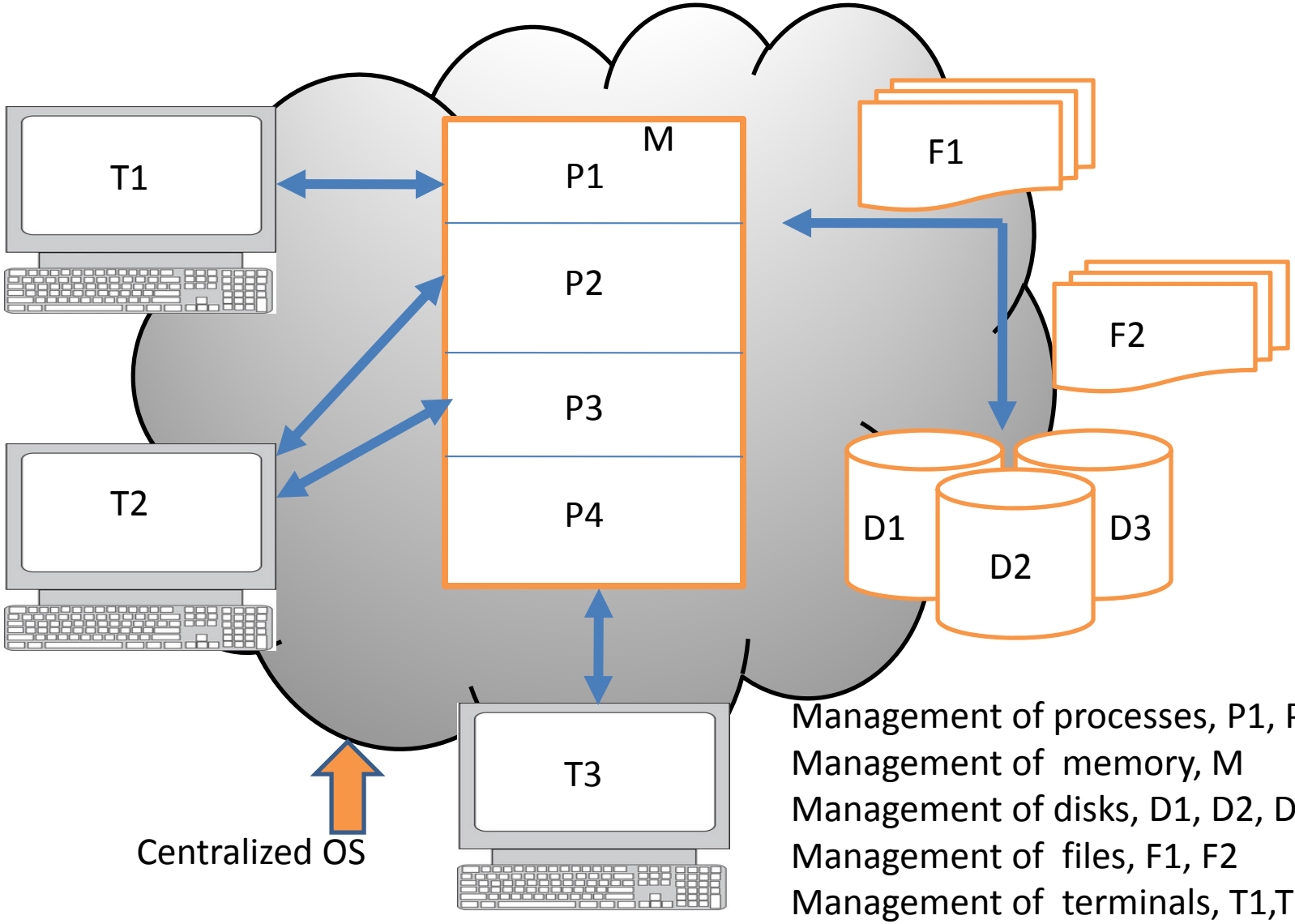
# Generic Components (H/W) of a DCS

- (2) Networking
  - Interconnected heterogeneous computer system which can exchange information.
  - LAN: connects computers at single location and normally owned by single organization.
  - INTERNET: a world-wide collection of thousands of interconnected computers and no one own it.
- DCS is networked computers providing single computer view to the user.

# Evolution of Modern OS

- First generation OS
  - System :
    - Centralized OS
  - Characteristics :
    - Process Management
    - Memory Management
    - I/O Management
    - File Management.
  - Goals :
    - Resource Management

# Centralized OS

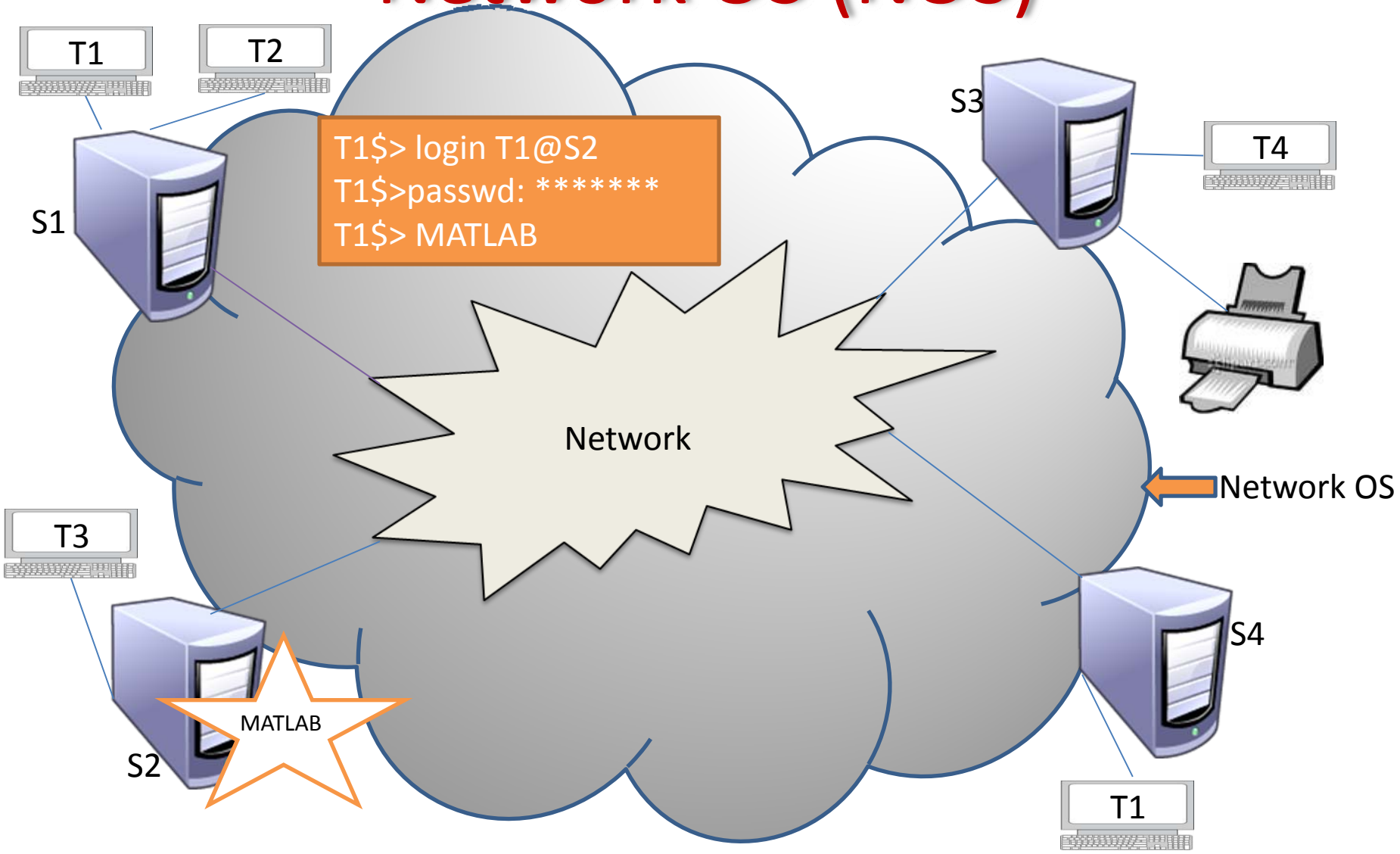



Management of processes, P1, P2, P3, P4
Management of memory, M
Management of disks, D1, D2, D3
Management of files, F1, F2
Management of terminals, T1,T2,T3

# Evolution of Modern OS

- Second generation OS
  - System :
    - Network OS
  - Characteristics :
    - Remote access
    - Information exchange
    - Network browsing
  - Goals :
    - Interoperability – sharing of resources between the systems.

# Network OS (NOS)

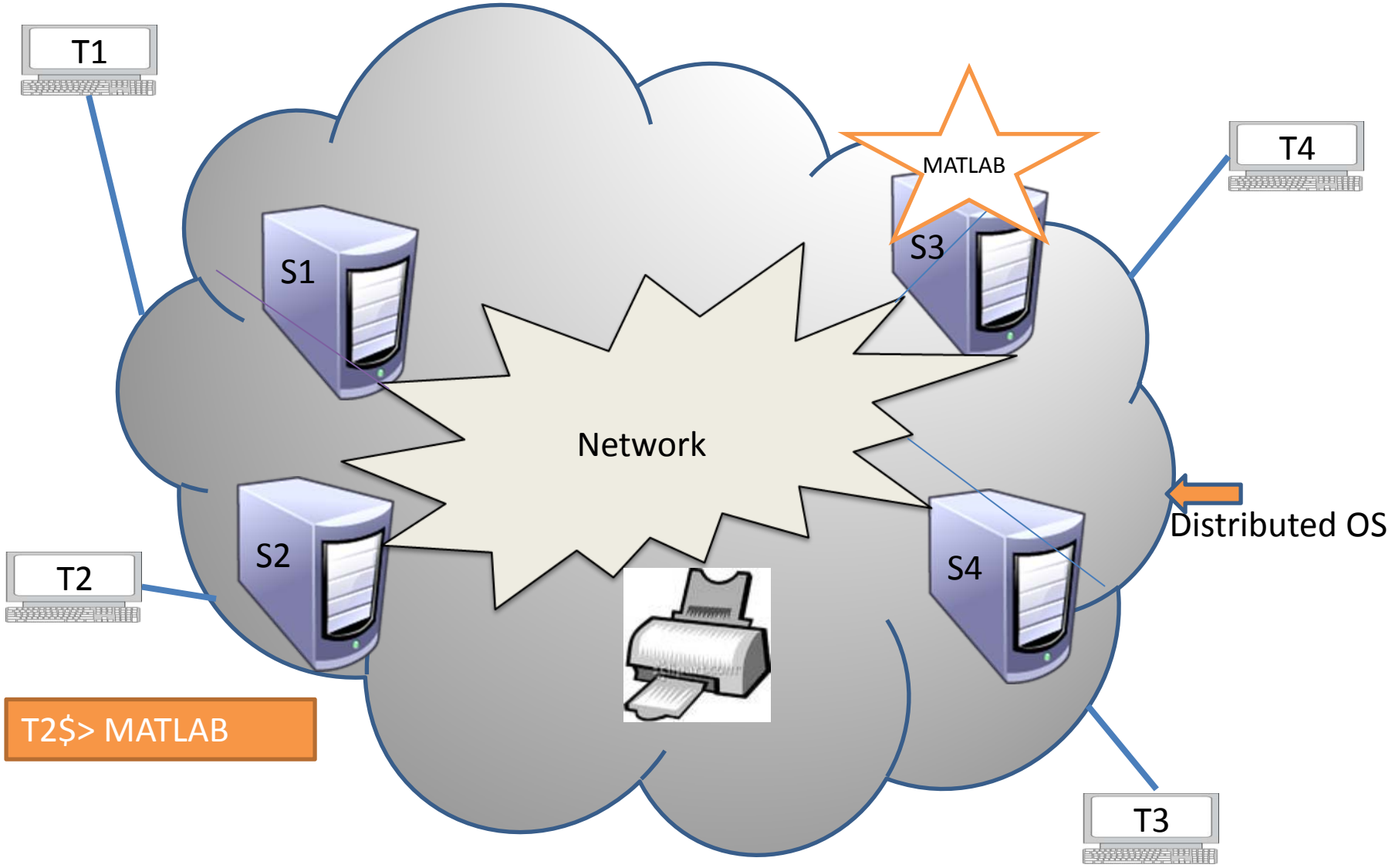



Accessing of remote servers, remote high speed printers over network

# Evolution of Modern OS

- Third generation OS
  - System :
    - Distributed OS
  - Characteristics :
    - Global view of computational power, file systems, name space,…etc.
  - Goals :
    - Single computer view of multiple heterogeneous computer systems

# Distributed OS (DOS)




Single computer view of multiple heterogeneous computer systems

# Differences between NOS and DOS

- NOS
  - User need to login explicitly to access the remote resources
  - User need to know whereabouts of the resources
  - No implicit sharing of loads
  - Full local autonomy at each system
- DOS
  - No explicit login to the system to access any resources
  - Gives the illusion of having all the resources locally available
  - Sharing of loads between nodes (load balancing)
  - Availability & Reliability is more

# Evolution of Modern OS

- Fourth generation OS
  - System :
    - Cooperative autonomous system - CAS
  - Characteristic :
    - Cooperative distributed applications
  - Goals :
    - Cooperative work

# Differences between DOS and CAS

- DOS
  - Top down design approach
  - Top level services are specified. It is then decomposed and implemented as low level services.
  - No legacy system is involved.
- CAS
  - Design approach is bottom up
  - Existing services are integrated in possible way to form new services.
  - Objective here is to develop the distributed system involving legacy system

# Distributed Computing System

- Multi processor system
- Processors are geographically apart
- Communication is by message passing
- Coordination among the processors to perform the specific task.

# Advantages of DCS over Centralized System

- Speed: 100 CPUs will have 100 times the speed of single processor ideally.
- Inherent distribution of applications: Some applications are inherently distributed.
  - For example, a bank may have many branches with the computing systems distributed across the geographical region to cater the local need. However, they need to coordinate for global need like *accounting across the banks.*
- Robustness: It is viewed with following metrics:
  - Availability: Probability that system is up at any given time interval.
  - Reliability: Probability that system is up at any point of time.
  - Both Availability and Reliability are high in DCS
- Incremental growth: Incrementally adding of resources will be gradually visualized performance wise.

# Disadvantages of DCS over Centralized System

- Software:
  - Would be complex
- Network problem:
  - Network saturation
  - Malfunctioning of network
- Security:
  - Possibility of security violation since the private data are visible to others over the network.

# Design Goals of DCS

- Efficiency:
  - Bottleneck: Communication delay, load balancing, number of messages for distributed algorithms
- Flexibility:
  - This refers to the friendliness of the system. From the systems point of view, it includes modularity, scalability, portability and interoperability.
- Consistency:
  - Computation should give correct results irrespective of time at which the computation is done for the same input conditions.
- Robustness:
  - The system should perform effectively and consistently in spite of failures and attempted security violations.

# Design Goals of DCS

- Scalability:
  - This refers to the issue that the growth of the system should not result in degraded performance or unavailability.
  - Scalable resources are – bandwidth, storage, processing power…etc.
- Compatibility: Three levels of compatibility are possible
  - Binary level: same binary code can be executed at all nodes.
  - Execution level: same source code can be compiled and linked, and executed on any system
  - Protocol level: It achieves interoperability by acquiring system components to support common set of protocol.

# Transparencies in DCS

- Access transparency:
  - Local and remote objects should be accessed in a uniform way. User should not find any difference in accessing local and remote objects.
- Location transparency:
  - Objects are referred by logical names which hide the physical location of the objects. For example, a network printer connected to a server S1 on LAN L1 is referred by the name associated with that printer and the exact location of the printer is invisible to the user.
- Migration transparency:
  - Movement of object (migration) from one system to other is invisible to user. Load balancing is one among many reason for migration of objects.
- Concurrency transparency:
  - Sharing of objects without interference. The same data may be referred by different applications. Performing of concurrency control to ensure the consistency of data should be invisible to the user.

# Transparencies in DCS

- Replication transparency:
  - Replica of files and data are transparent to the user. One of the reason for replication is to ensure ‘robustness’ design goal.
- Parallelism transparency:
  - A user task may be scheduled for execution at different node/system which is invisible to the user. The reason for scheduling different node is to ensure ‘efficiency’ design goal.
- Failure transparency:
  - Graceful degradation in the performance when ever there are system failures.
- Size transparency:
  - Addition/deletion of resources should not effect the execution of task.
- Revision transparency:
  - Software revisions in the system are not visible to the user.

# Relationship between Design Goals and Transparencies

| Design Goals | Transparencies |
| --- | --- |
| Efficiency | Concurrency<br>Parallelism |
| Flexibility | Access<br>Location<br>Migration<br>Size<br>Revision |
| Consistency | Access<br>Replication<br>Performance |
| Robustness | Failure<br>Replication<br>Size<br>Revision |

# Fundamental Issues in DCS

- Global state detection
  - It is impossible to get the global up-to-date information about DCS due to
    - Lack of global memory
    - Lack of global clock
    - Unpredictable message delays
  - Global state of DCS is important for the consistency check
- Ordering of events
  - Due to lack of global clock, ordering of events across the systems in DCS is an issue
- Naming
  - A server needs to be identified in order to use its service.
  - A server can be identified using name, physical address or its service.
  - To handle the transparency issues need to map the physical name to the logical name.

# Fundamental Issues in DCS

- Inter-process communication
  - Message passing communication
  - Request – reply communication
  - Transaction communication
  - Group communication
- Inter-process synchronization
  - Distributed mutual exclusion
  - Leader election
  - Agreement problems
- Resource management
  - Data migration
    - Distributed file system
    - Distributed shared memory
  - Computation migration
    - Distributed scheduling
    - Remote Procedure Call

# Fundamental Issues in DCS

- Fault tolerance
  - Atomic execution of transactions
  - Replication management
  - Check pointing and recovery
- Security
  - Secrecy of data (Cryptography): Unauthorized access prevention
  - Authentication: Who are you?
  - Verification of sender (Digital signature): Message sent by X has really been sent by X.

# Summary

- Definition, Basics of Distributed Computing is discussed.
- Differences between Distributed OS and Network OS is outlined.
- Requirements in terms of design goals, transparencies and fundamental issues in a Distributed Computing System is discussed.

# References


- Advanced Operating Systems
  - M Singhal and N G Shivarathri, McGraw Hill, International
- Distributed Operating Systems and Algorithms
  - Randy Chaw and T Johnson, Addison Wesley
- Distributed Operating Systems
  - A S Tanenbaum, Prentice Hall
- Distributed Algorithms
  - Nancy A Lynch, Morgan Kaufman
- Distributed Systems Concepts and Design
  - G Coulouris and J Dollimore, Addison Wesley