Chapter 1

Introduction to
Distributed Database Management Systems

Contents

- Distributed Data Processing
- Concepts of Distributed Data Base Systems
- Data Delivery Alternatives
- Data Delivery Alternatives
- Distributed DBS Promises
- Distributed DBS Issues
- Review of Computer Networks

Distributed Data Processing

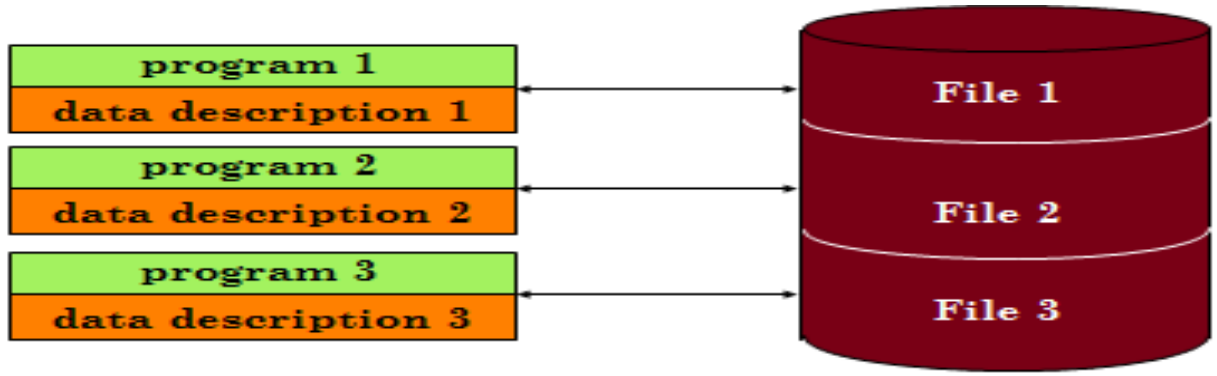
- Distributed data processing (or distributed computing) is a number of autonomous *processing elements* that are interconnected by a computer network and that cooperate in performing their assigned tasks.
  - The "processing element" is a computing device that can execute a program on its own.
  - The term "distributed" can refer to:
    - Processing logic - processing logic or processing elements are distributed over the network
    - Function -Various functions of a computer system could be delegated to various pieces of hardware or software
    - Data - Data used by a number of applications may be distributed to a number of processing sites
    - Control- The control of the execution of various tasks might be distributed instead of being performed by one computer system.

Concepts of Distributed Data Base Systems

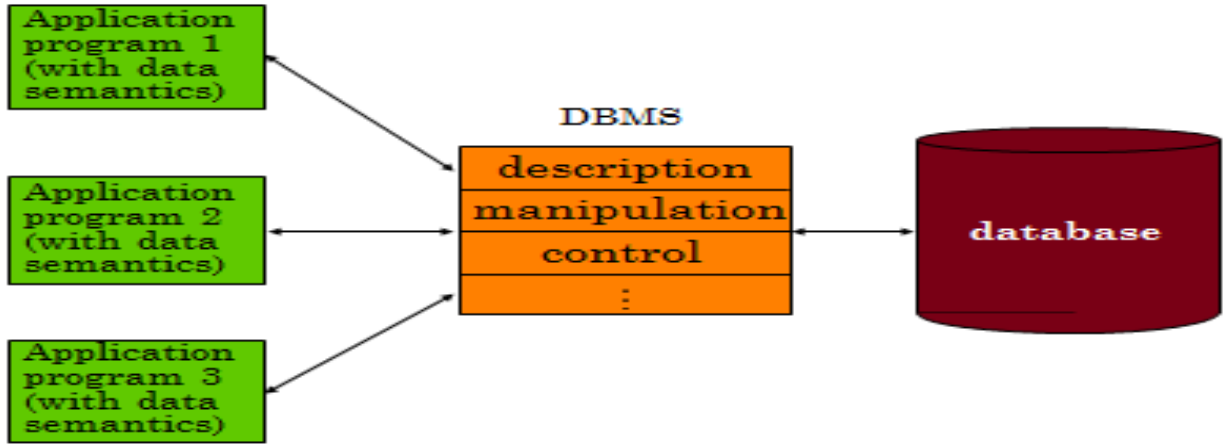
- A distributed database (DDB) is a collection of multiple, *logically interrelated* databases distributed over a *computer network*.
- A distributed database management system (DDBMS) is the software that manages the DDB and provides an access mechanism that makes this distribution transparent to the users.
- Distributed database system (DDBS) = DDB + DDBMS
- DDBS makes distributed processing easier and more efficient
- DDBS technology is the integration of two opposed approaches to data processing: database system and computer network technologies.

Concepts of Distributed Data Base Systems(Cont.…)

- Database systems involves in data processing in which each application defines and maintains its own data:

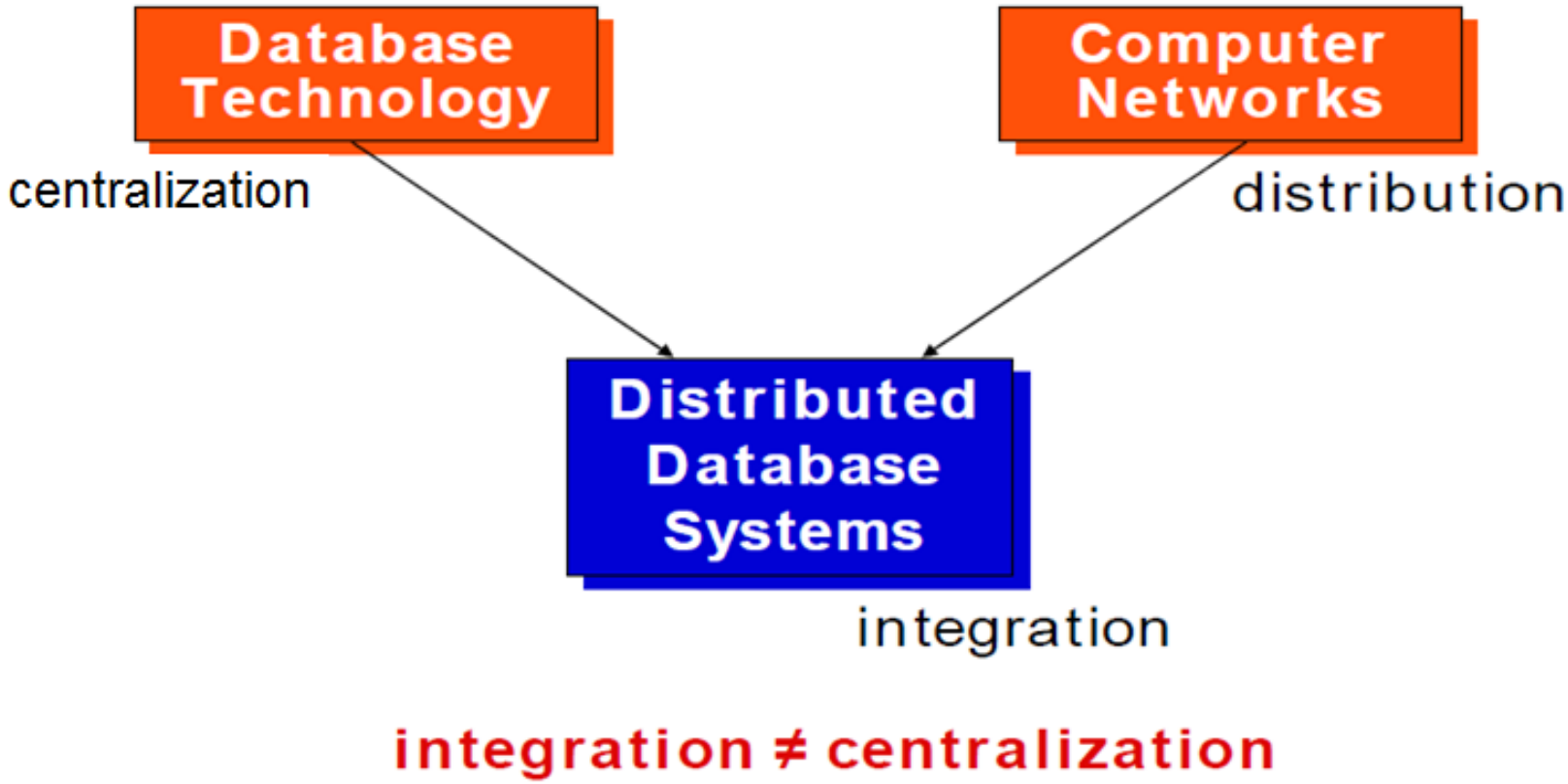


- It maintains its data to one in which the data are administered *centrally*:



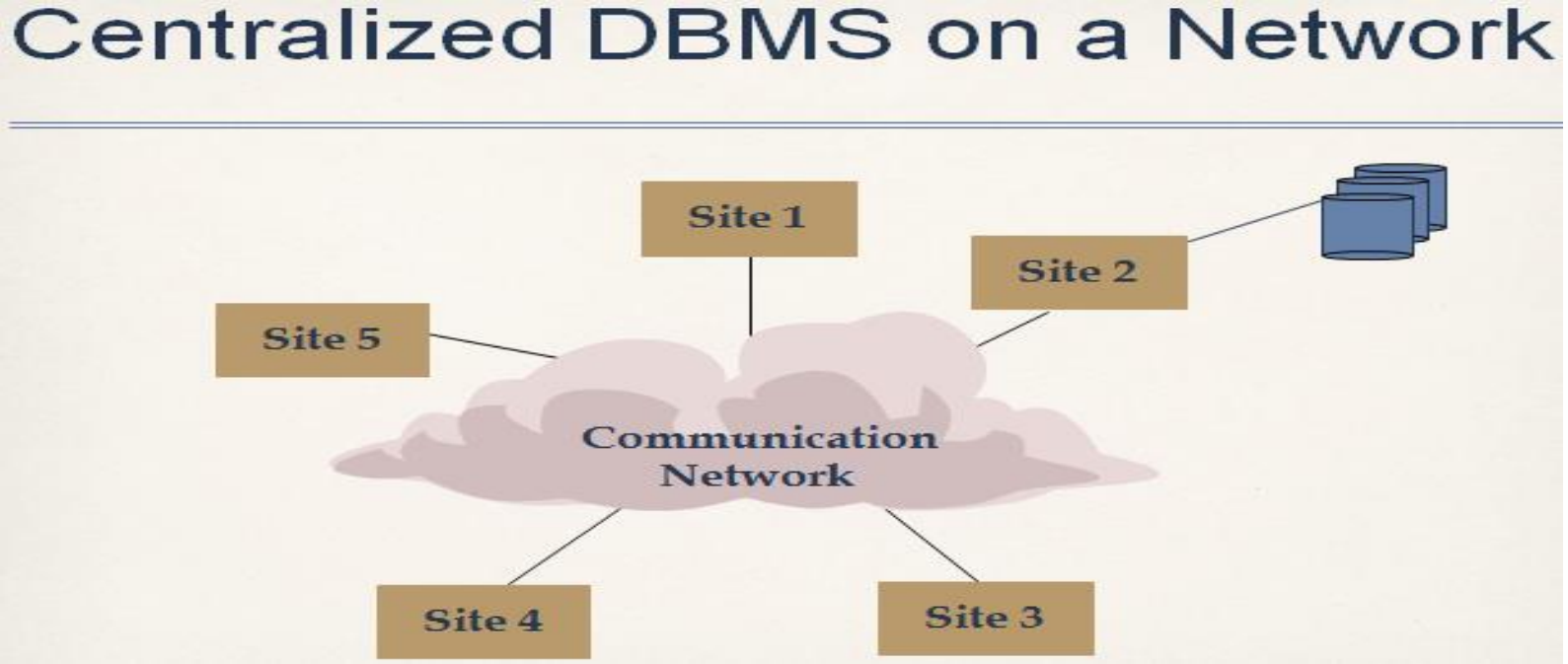
Concepts of Distributed Data Base Systems(Cont.…)

- Computer networks, on the other hand, promotes a mode of work that goes against all centralization efforts.
- The most important objective of the DDBS technology is integration, not centralization:



Concepts of Distributed Data Base Systems (Cont.…)

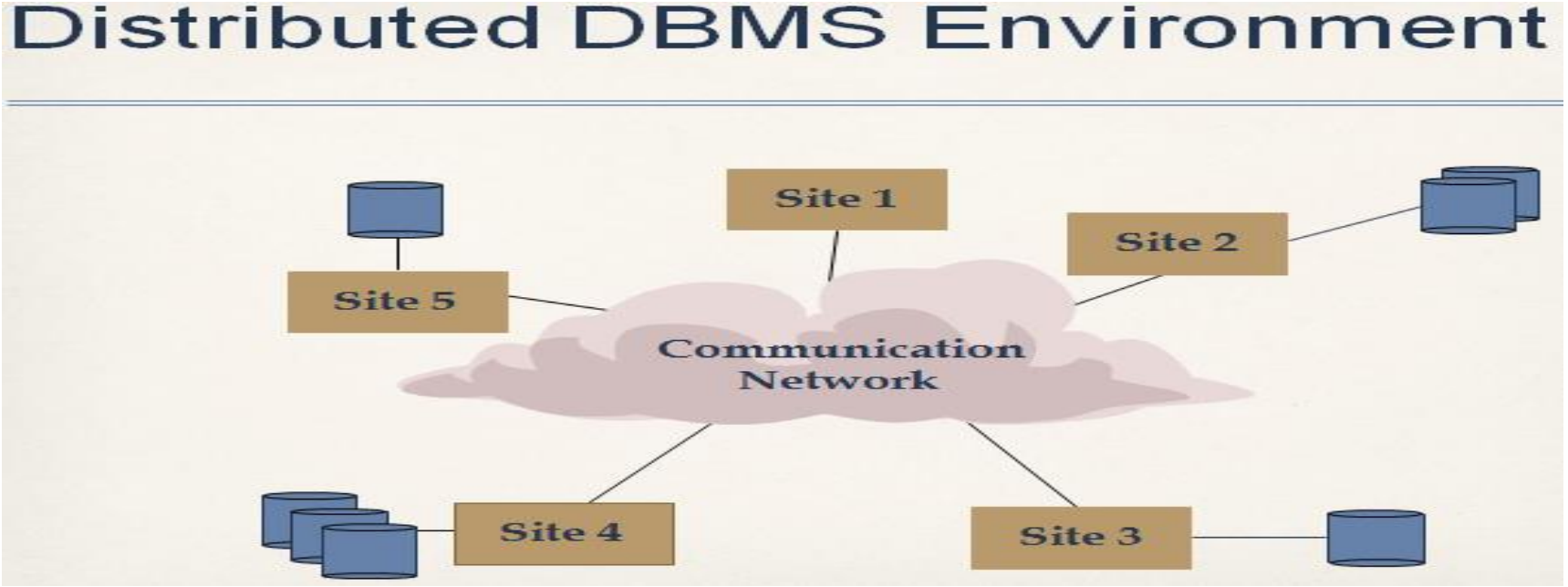
- A DDBS is not a system where, despite the existence of a network, the database resides at only one node of the network:



→ In this case, the database is centrally managed by one computer system (site 2 in the Figure) and all the requests are routed to that site.

Concepts of Distributed Data Base Systems(Cont.…)

- In DDBS, data are distributed among a number of sites:



- The existence of a computer network or a collection of “files” is not sufficient to form a distributed database system

Concepts of Distributed Data Base Systems (Cont.…)

- Implicit Assumptions:
  - Data stored at a number of sites each site *logically* consists of a single processor.
  - Processors at different sites are interconnected by a computer network no multiprocessors
    - parallel database systems
  - Distributed database is a database, not a collection of files data logically related as exhibited in the users' access patterns
    - relational data model
  - D-DBMS is a full-fledged DBMS
    - not remote file system

Data Delivery Alternatives

- In distributed databases, data are "delivered" from the sites where they are stored to where the query is posed.
- Data delivery alternatives can be characterized along three dimensions:
  - delivery modes
    - Pull-only
    - Push-only
    - Hybrid
  - frequency
    - Periodic
    - Conditional
    - Ad-hoc or irregular
  - communication methods
    - Unicast
    - One-to-many

Data Delivery Alternatives (Cont.…)

- **Delivery modes**
  - Pull-only: In this mode of data delivery, the transfer of data from servers to clients is initiated by a client pull. When a client request is received at a server, the server responds by locating the requested information.
    - Conventional DBMSs offer primarily pull-based data delivery.
  - Push-only: In this mode, the transfer of data from servers to clients is initiated by a server push in the absence of any specific request from clients.
    - Difficulty is in deciding which data would be of common interest, and when to send them to clients
  - Hybrid: combines the client-pull and server-push mechanisms.
    - The continuous (or continual) query approach presents one way of hybrid mode-namely, the transfer of information from servers to clients is first initiated by a client pull (by posing the query), and the subsequent transfer of updated information to clients is initiated by a server push.

Data Delivery Alternatives (Cont.…)

- **Frequency**
  - Periodic: In periodic delivery, data are sent from the server to clients at regular intervals. Both pull and push can be performed in periodic fashion.
    - Periodic delivery is carried out on a regular and pre-specified repeating schedule.
  - Conditional: In conditional delivery, data are sent from servers whenever certain conditions installed by clients in their profiles are satisfied.
    - Conditional delivery is mostly used in the hybrid or push-only delivery systems.
  - Ad-hoc: Ad-hoc delivery is irregular and is performed mostly in a pure pull-based system.
    - Data are pulled from servers to clients in an ad-hoc fashion whenever clients request it. In contrast, periodic pull arises when a client uses polling to obtain data from servers based on a regular period (schedule).

Data Delivery Alternatives (Cont.…)

- **Communication Methods:** These methods determine the ways in which servers and clients communicate for delivering information to clients. The alternatives are:
  - →unicast and
  - →one-to-many.
- In unicast, the communication from a server to a client is one-to-one: the server sends data to one client using a particular delivery mode with some frequency.
- In one-to-many, as the name implies, the server sends data to a number of clients.

Distributed DBS Promises

Four fundamental promises/advantages of DDBS are:

1. Transparent management of distributed, fragmented, and replicated data
2. Improved reliability/availability through distributed transactions
3. Improved performance
4. Easier and more economical system expansion

Distributed DBS Promises (Cont.…)

- **Transparency**: Transparency refers to separation of the higher-level semantics of a system from lower-level implementation issues.
  - → It "hides" the implementation details from users.
  - → Transparency provides
    - ✦ Data independence in the distributed environment
    - ✦ Network (distribution) transparency
    - ✦ Replication transparency
    - ✦ Fragmentation transparency
      - ✓ horizontal fragmentation: selection
      - ✓ vertical fragmentation: projection
      - ✓ hybrid

Distributed DBS Promises (Cont.…)

- **Data independence** : Refers to the immunity of user applications to changes in the definition and organization of data, and vice versa.
- Data definition occurs at two levels:
  - Schema Definition : At this level, the logical structure of the data are specified
  - Physical Data Description: At this level, the physical structure of the data are specified.
- Two types of data independence:
  - Logical Data Independence: Refers to the immunity of user applications to changes in the logical structure (i.e., schema) of the database
  - Physical Data Independence: Deals with hiding the details of the storage structure from user applications

Distributed DBS Promises (Cont.…)

- **Network (Distribution) Transparency**: Network Transparency provides the protection to the user from the operational details of the network; possibly even hiding the existence of the network.
  - The user finds no difference between database applications that would run on a centralized database and those that would run on a distributed database.
- Two types of Network Transparency:
  - Location transparency: Refers to the fact that the command used to perform a task is independent of both the location of the data and the system on which an operation is carried out
  - Naming transparency: Means that a unique name is provided for each object in the database.
    - In the absence of naming transparency, users are required to embed the location name (or an identifier) as part of the object name.

Distributed DBS Promises (Cont.…)

- **Replication Transparency**: Replication Transparency distribute data in a replicated fashion across the machines on a network.
- Replication helps performance since diverse and conflicting user requirements can be more easily accommodated:
  - Data that are accessed by one user can be placed on that user's local machine as well as on the machine of another user with the same access requirements.
  - If one of the machines fails, a copy of the data are still available on another machine on the network.
- The decision as to whether to replicate or not, and how many copies of any database object to have, depends to a considerable degree on user applications.
- Note: Replication transparency refers only to the existence of replicas, not to their actual location.

Distributed DBS Promises (Cont.…)

- **Fragmentation Transparency**: This divides each database relation into smaller fragments and treat each fragment as a separate database object (i.e., another relation).
  - ➞ This is done for reasons of performance, availability, and reliability.
- Fragmentation can reduce the negative effects of replication.
  - ➞ Each replica is not the full relation but only a subset of it; thus less space is required and fewer data items need be managed.
- Two types of fragmentation transparency:
  - ➞ Horizontal fragmentation: a relation is partitioned into a set of sub-relations each of which have a subset of the tuples (rows) of the original relation
  - ➞ Vertical fragmentation : each sub-relation is defined on a subset of the attributes (columns) of the original relation.

Distributed DBS Promises (Cont.…)

Example:

- Consider a firm that has offices in Boston, Montreal, New York, Paris and Tokyo. They run projects at each of these sites and maintain a database of their employees, the projects and other related data.
- Assuming that the database is relational, we can store this information in four relations:
  - EMP(ENO, ENAME, TITLE)
  - PROJ(PNO, PNAME, BUDGET)
  - PAY(TITLE, SAL)
  - ASG(ENO, PNO, RESP, DUR)

  (ASG indicates which employees have been assigned to which projects for what duration with what responsibility)

Distributed DBS Promises (Cont.…)

Example (contd…):

EMP

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E2	M. Smith	Syst. Anal.
E3	A. Lee	Mech. Eng.
E4	J. Miller	Programmer
E5	B. Casey	Syst. Anal.
E6	L. Chu	Elect. Eng.
E7	R. Davis	Mech. Eng.
E8	J. Jones	Syst. Anal.

ASG

ENO	PNO	RESP	DUR
E1	P1	Manager	12
E2	P1	Analyst	24
E2	P2	Analyst	6
E3	P3	Consultant	10
E3	P4	Engineer	48
E4	P2	Programmer	18
E5	P2	Manager	24
E6	P4	Manager	48
E7	P3	Engineer	36
E8	P3	Manager	40

PROJ

PNO	PNAME	BUDGET
P1	Instrumentation	150000
P2	Database Develop.	135000
P3	CAD/CAM	250000
P4	Maintenance	310000

PAY

TITLE	SAL
Elect. Eng.	40000
Syst. Anal.	34000
Mech. Eng.	27000
Programmer	24000

Distributed DBS Promises (Cont.…)

Example (contd…):

- If all of this data are stored in a centralized DBMS, and we want to find out the names and employees who worked on a project for more than 12 months, we would specify this using the following SQL query:

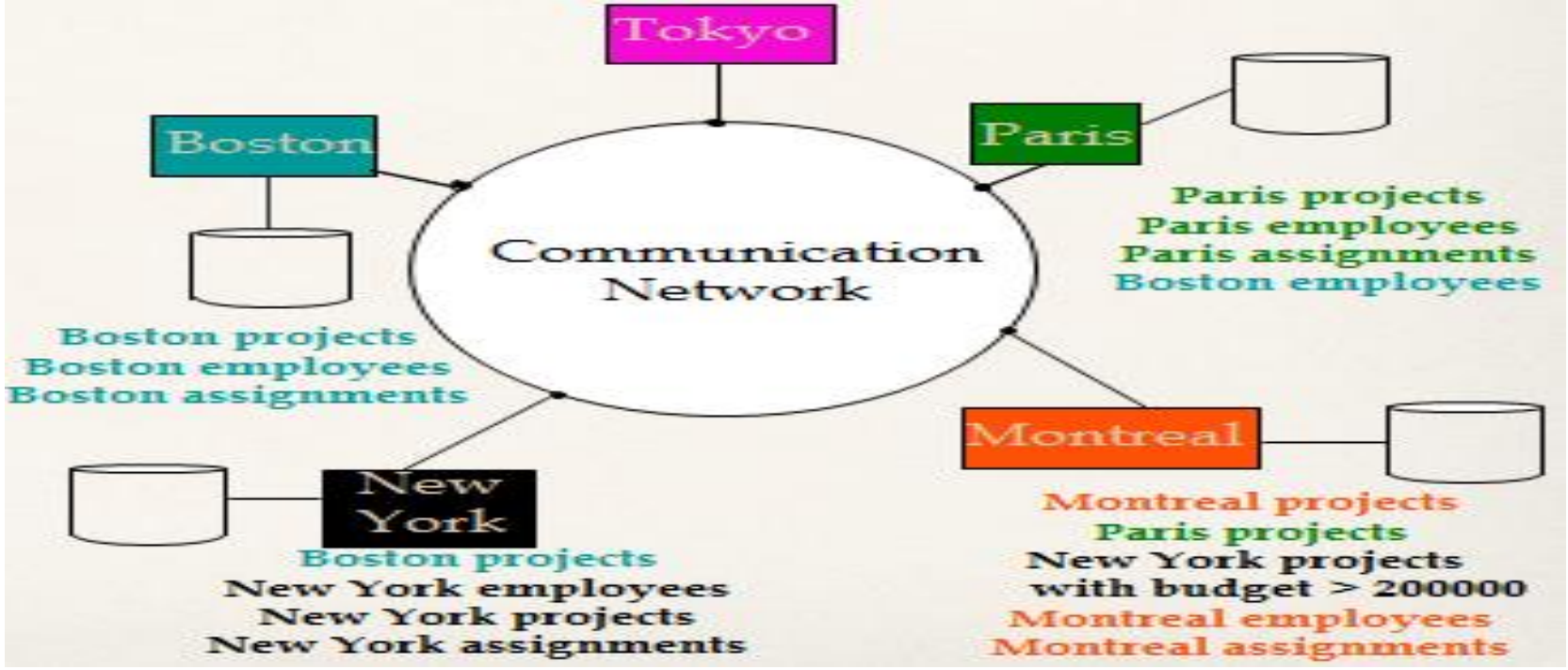
```
SELECT ENAME,SAL
FROM   EMP,ASG,PAY
WHERE  DUR > 12
AND    EMP.ENO = ASG.ENO
AND    PAY.TITLE = EMP.TITLE
```

- But under DDBMS,
  - ➞ data are localized such that data about the employees in Boston office are stored in Boston, those in the Montreal office are stored in Montreal, and so forth.
    - ✦ The same applies to the project and salary information.

Distributed DBS Promises (Cont.…)

Example (contd…):

- Under DDBS,
  - → we partition each of the relations and store each partition at a different site. This is *fragmentation*.
  - → We can duplicate some of this data at other sites for performance and reliability reasons. This is *replication*.
  - → The result is a distributed database which is fragmented and replicated



Distributed DBS Promises (Cont.…)

Reliability through Distributed Transaction

- A transaction is a basic unit of consistent and reliable computing, consisting of a sequence of database operations executed as an atomic action.
- It transforms a consistent database state to another consistent database state even when a number of such transactions are executed concurrently (sometimes called *concurrency transparency*), and even when failures occur (also called *failure atomicity*).
- A DBMS that provides full transaction support guarantees that concurrent execution of user transactions will not violate database consistency in the face of system failures as long as each transaction is correct, i.e., obeys the integrity rules specified on the database.

Distributed DBS Promises (Cont.…)

Reliability through Distributed Transaction

- Distributed transactions execute at a number of sites at which they access the local database.
- With full support for distributed transactions, user applications can access a single logical image of the database and rely on the distributed DBMS to ensure that their requests will be executed correctly no matter what happens in the system.
  - → User applications do not need to be concerned with coordinating their accesses to individual local databases nor do they need to worry about the possibility of site or communication link failures during the execution of their transactions.

Distributed DBS Promises (Cont.…)

Improved Performance

- Two points of improved performance:
  - a distributed DBMS fragments the conceptual database, enabling data to be stored in close proximity to its points of use (also called *data localization*).
  - the inherent parallelism of distributed systems may be exploited for
    - **inter-query parallelism-** results from the ability to execute multiple queries at the same time
    - **intra-query parallelism-** breaks up a single query into a number of subqueries each of which is executed at a different site, accessing a different part of the distributed database.

Distributed DBS Promises (Cont.…)

Easier System Expansion

- In a DDBS, it is much easier to accommodate increasing database sizes.
- Expansion can be handled by adding processing and storage power to the network.
  - ➞ One aspect of easier system expansion is economics.
    - ✦ It costs much less to put together a system of "smaller" computers with the equivalent power of a single big machine.

Distributed DBS Issues

- The issues that arise in building a distributed DBMS are:
  - Distributed Database Design
  - Distributed Directory Management
  - Distributed Query Processing
  - Distributed Concurrency Control
  - Distributed Deadlock Management
  - Reliability of Distributed DBMS
  - Replication
  - Additional Issues

Distributed DBS Issues (Cont.…)

Issues in Distributed Database Design: How the database and the applications that run against it should be placed across the sites?

- Two basic alternatives to placing data:
  - partitioned (or non-replicated) scheme: database is divided into a number of disjoint partitions each of which is placed at a different site.
  - Replicated Scheme: Replicated designs can be either:
    - fully replicated (also called fully duplicated) where the entire database is stored at each site, or
    - partially replicated (or partially duplicated) where each partition of the database is stored at more than one site, but not at all the sites.

Distributed DBS Issues (Cont.…)

Issues in Distributed Database Design:

- The two fundamental design issues are:
  - Fragmentation: the separation of the database into partitions called fragments, and
  - Distribution: the optimum distribution of fragments.
- Research on Distributed Database Design Issue: Involves mathematical programming in order to minimize the combined cost of:
  - storing the database,
  - processing transactions against it, and
  - message communication among sites.

Distributed DBS Issues (Cont.…)

Issues in Distributed Directory Management : How the directory should be placed across the sites?

- A directory contains information (such as descriptions and locations) about data items in the database.

- A directory may be
  - global to the entire DDBS or
  - local to each site
- A directory can be
  - centralized at one site or
  - distributed over several sites

Distributed DBS Issues (Cont.…)

Issues in Distributed Query Processing: How to decide on a strategy for executing each query over the network in the most cost-effective way?

- ➞ Query processing deals with designing algorithms that analyze queries and convert them into a series of data manipulation operations.

- The factors to be considered are:
  - ➞ the distribution of data,
  - ➞ communication costs, and
  - ➞ lack of sufficient locally-available information
- The objective is to optimize the performance of executing the transaction, subject to the above-mentioned constraints.

Distributed DBS Issues (Cont.…)

Issues in Distributed Concurrency Control:

Issue I: How to synchronize concurrent accesses to the distributed database, such that the integrity of the database is maintained?

Issue II: How to maintain the consistency of multiple copies of the data base?

- Two fundamental primitives are:
  - → locking, which is based on the mutual exclusion of accesses to data items, and
  - → timestamping, where the transaction executions are ordered based on timestamps.

Distributed DBS Issues (Cont.…)

Issues in Distributed Deadlock Management:

- How to manage the deadlocks that arise due to the competition among users for access to the data, if the synchronization mechanism is based on locking?

- The well-known alternatives that apply to DDBS are:
  - prevention,
  - avoidance, and
  - detection/recovery

Distributed DBS Issues (Cont.…)

Issues in Reliability in DDBMS:

- How to maintain consistency and up-to-date in the databases at the operational sites, when a failure occurs and various sites become either inoperable or inaccessible?
- How to recover and bring the databases at the failed sites up-to-date when the computer system or network recovers from the failure?

Issues in Replication:

- How to ensure the consistency of the replicas (i.e., copies of the same data item have the same value), if the distributed database is (partially or fully) replicated ?

Distributed DBS Issues (Cont.…)

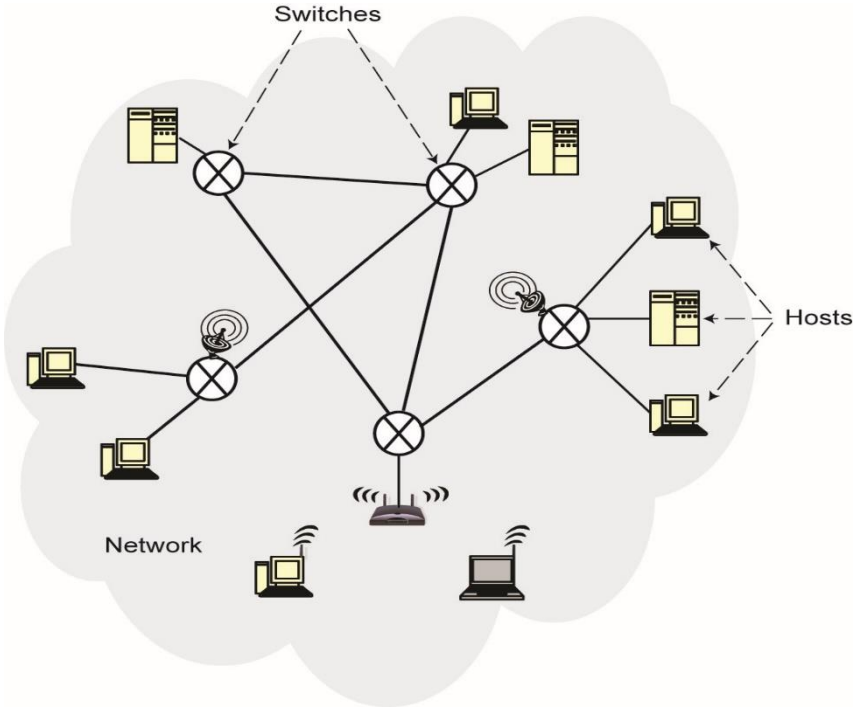
Additional Issues in DDBS:

- Database design issues in multidatabase systems (i.e., database integration)
- Issues relating to peer-to-peer data management
- Issues that arise in web data management
- Data management issues in a parallel system

Review of Computer Networks

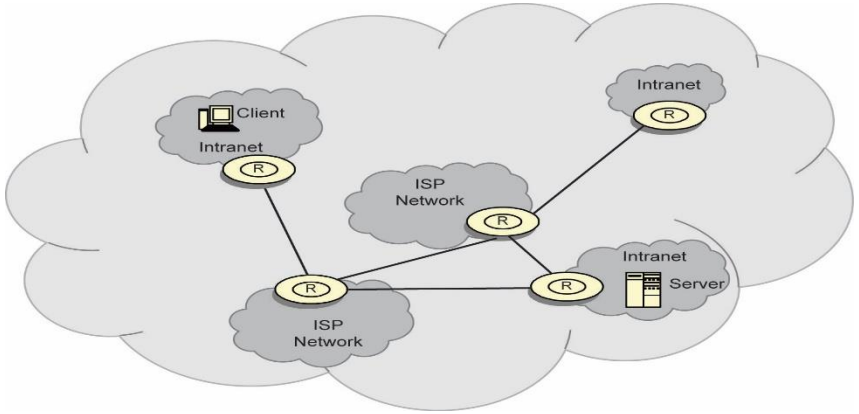
Computer Networks:

- An interconnected collection of autonomous computers that are capable of exchanging information among themselves.
- Components
  - Hosts (nodes, end systems)
  - Switches
  - Communication link



Internet:

- Network of networks



Review of Computer Networks (Cont.…)

Types of Networks:

- According to scale (geographic distribution)
  - Wide are network (WAN)
    - Distance between any two nodes > 20 km and can go as high as thousands of kms
    - Long delays due to distance traveled
    - Heterogeneity of transmission media
    - Speeds of 150 Mbps to 10 Gbps (OC192 on the backbone)
  - Local area network (LAN)
    - Limited in geographic scope (usually < 2km)
    - Speeds 10-1000 Mbps
    - Short delays and low noise
  - Metropolitan area network (MAN)
    - In between LAN and WAN

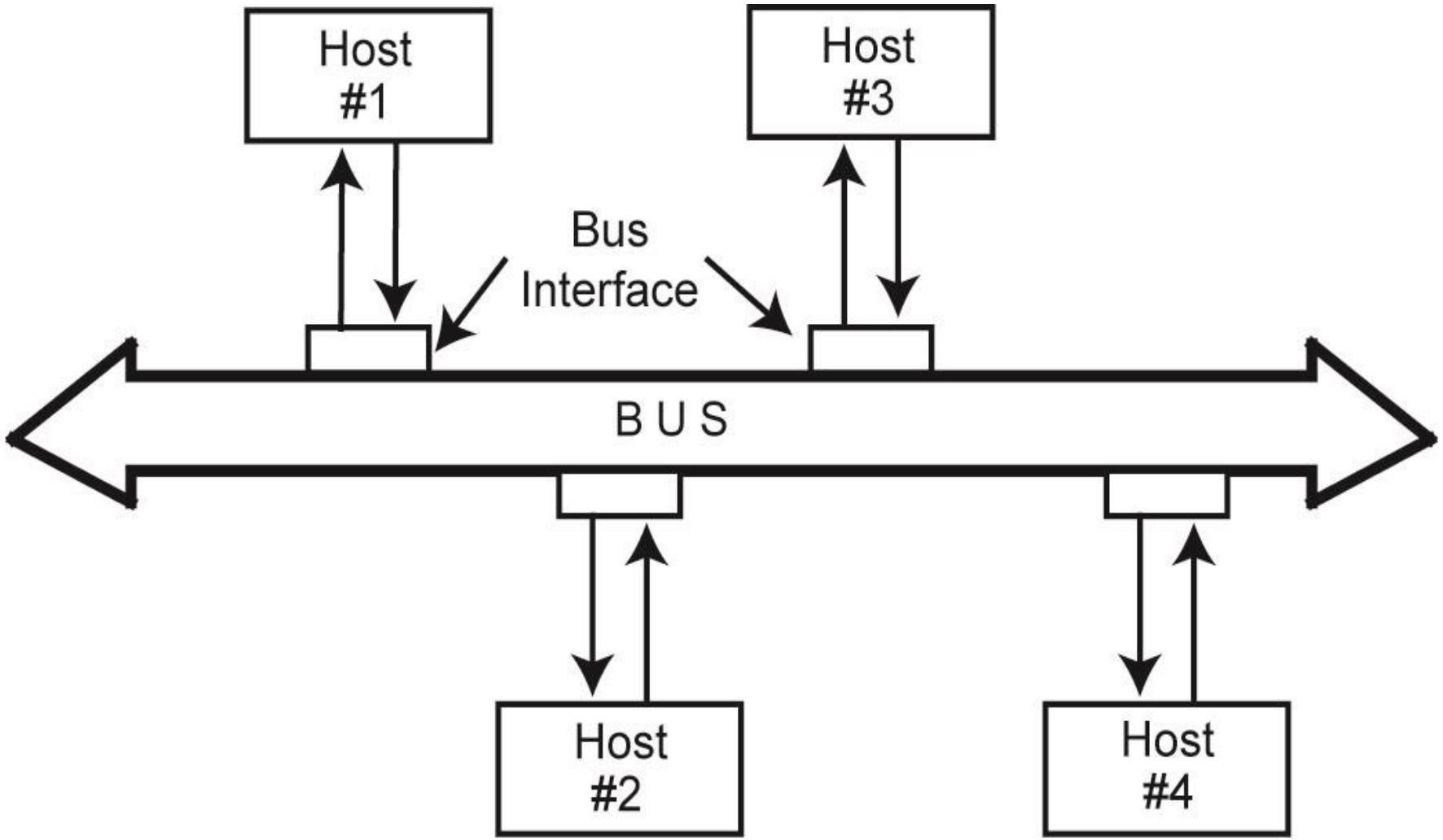
Review of Computer Networks (Cont.…)

Types of Networks (cont'd):

- Topology
  - Irregular
    - No regularity in the interconnection – e.g., Internet
  - Bus
    - Typical in LANs – Ethernet
    - Using Carrier Sense Medium Access with Collision Detection (CSMA/CD)
      - Listen before and while you transmit
  - Star
  - Ring
  - Mesh

Review of Computer Networks (Cont.…)

Bus Network:



Review of Computer Networks (Cont.…)

Communication Schemes:

- Point-to-point (unicast)
  - One or more (direct or indirect) links between each pair of nodes
  - Communication always between two nodes
  - Receiver and sender are identified by their addresses included in the message header
  - Message may follow one of many links between the sender and receiver using switching or routing
- Broadcast (multi-point)
  - Messages are transmitted over a shared channel and received by all the nodes
  - Each node checks the address and if it not the intended recipient, ignores
  - Multi-cast: special case
    - Message is sent to a subset of the nodes

Review of Computer Networks (Cont.…)

Communication Alternatives:

- Twisted pair
- Coaxial
- Fiber optic cable
- Satellite
- Microwave
- Wireless

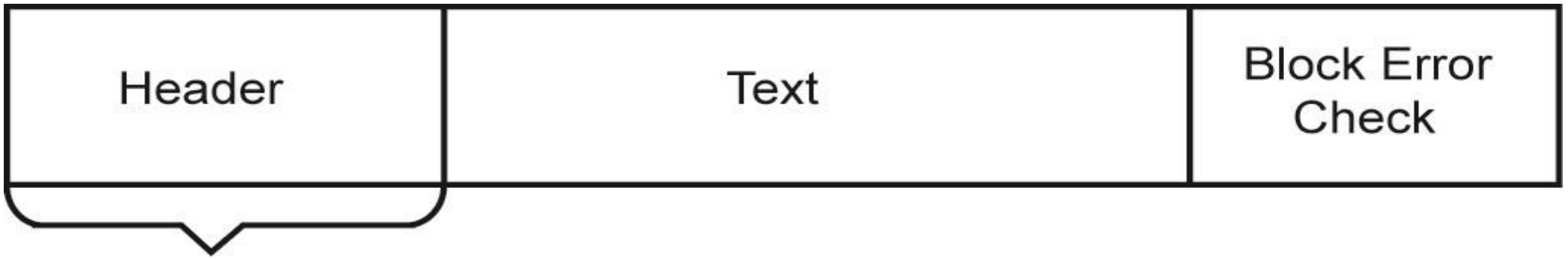
Review of Computer Networks (Cont.…)

Data Communication:

- Hosts are connected by links, each of which can carry one or more channels
- Link: physical entity; channel: logical entity
- Digital signal versus analog signal
- Capacity – bandwidth
  - The amount of information that can be transmitted over the channel in a given time unit
- Alternative messaging schemes
  - Packet switching
    - Messages are divided into fixed size packets, each of which is routed from the source to the destination
  - Circuit switching
    - A dedicated channel is established between the sender and receiver for the duration of the session

Review of Computer Networks (Cont.…)

Packet Formats:



- Source address
- Destination address
- Message number
- Packet number
- Acknowledgment
- Control information

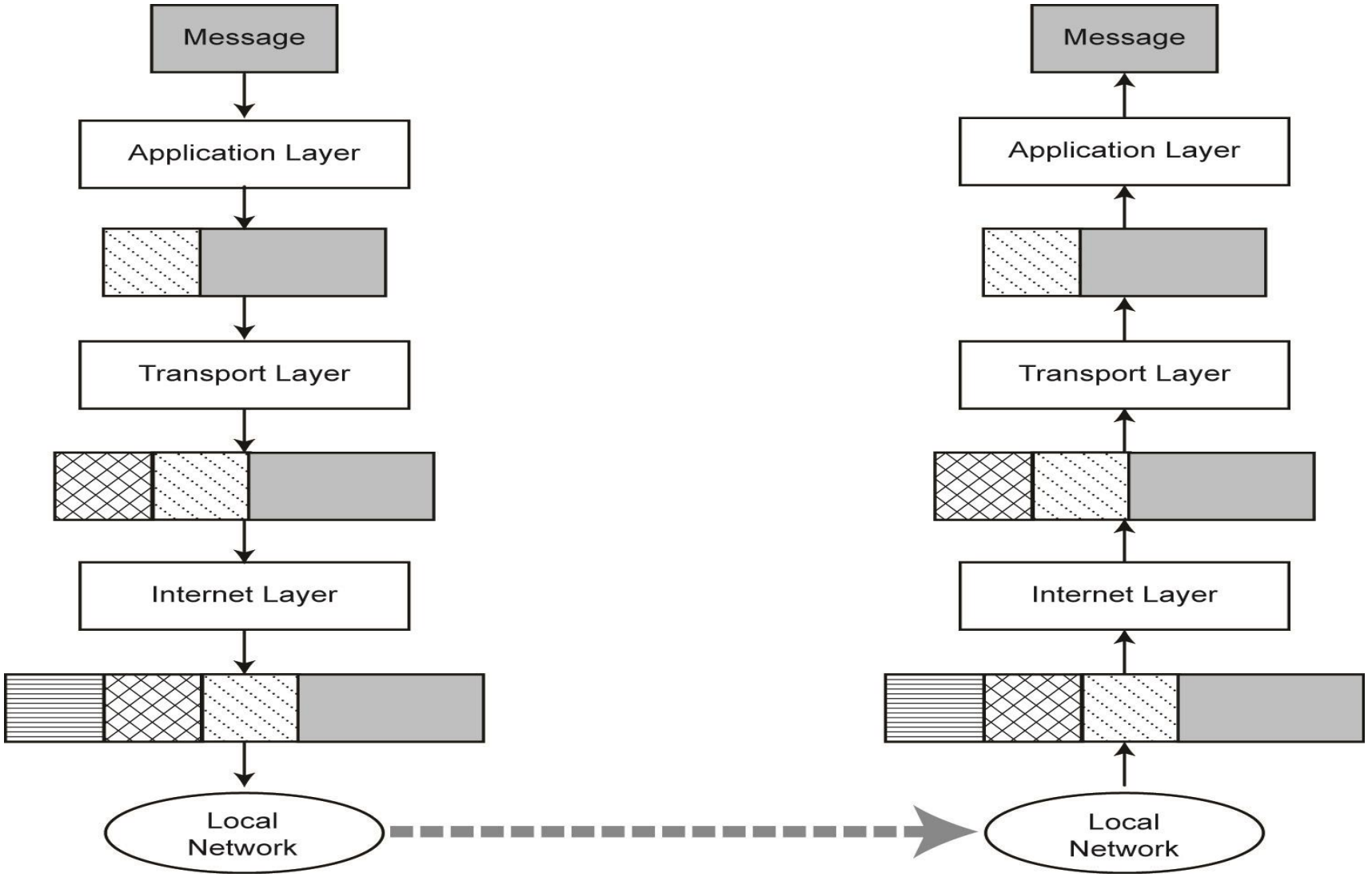
Review of Computer Networks (Cont.…)

Communication Protocols:

- Software that ensures error-free, reliable and efficient communication between hosts
- Layered architecture – hence protocol stack or protocol suite
- TCP/IP is the best-known one
  - ➞ Used in the Internet

Review of Computer Networks (Cont.…)

Message Transmission using TCP/IP:



Review of Computer Networks (Cont.…)

TCP/IP Protocol: