Outline

- Introduction
- Background
  - ➡ Relational database systems
  - ➡ Computer networks
- Distributed Database Design
- Database Integration
- Semantic Data Control
- Distributed Query Processing
- Multidatabase Query Processing
- Distributed Transaction Management
- Data Replication
- Parallel Database Systems
- Distributed Object DBMS
- Peer-to-Peer Data Management
- Web Data Management
- Current Issues

Relational Model

- Relation
  - ➡ A relation R with attributes $A = \{A_1, A_2, \ldots, A_n\}$ defined over n domains $D = \{D_1, D_2, \ldots, D_n\}$ (not necessarily distinct) with values $\{Dom_1, Dom_2, \ldots, Dom_n\}$ is a finite, time varying set of n-tuples $\langle d_1, d_2, \ldots, d_n\rangle$ such that $d_1 \in Dom_1$, $d_2 \in Dom_2$, ..., $d_n \in Dom_n$, and $A_1 \subseteq D_1$, $A_2 \subseteq D_2$, ..., $A_n \subseteq D_n$.
  - ➡ Notation: $R(A_1, A_2, \ldots, A_n)$ or $R(A_1{:}\ D_1, A_2{:}\ D_2, \ldots, A_n{:}\ D_n)$
  - ➡ Alternatively, given R as defined above, an instance of it at a given time is a set of n-tuples:

    $\{\langle A_1{:}\ d_1, A_2{:}\ d_2, \ldots, A_n{:}\ d_n\rangle \mid d_1 \in Dom_1, d_2 \in Dom_2, \ldots, d_n \in Dom_n\}$
- Tabular structure of data where
  - ➡ R is the table heading
  - ➡ Attributes are table columns
  - ➡ Each tuple is a row

Relation Schemes and Instances

- Relational scheme
  - ➡ A relation scheme is the definition; i.e., a set of attributes
  - ➡ A relational database scheme is a set of relation schemes:
    - ✦ i.e., a set of sets of attributes
- Relation instance (simply *relation*)
  - ➡ An relation is an instance of a relation scheme
  - ➡ a relation **r** over a relation scheme $R = \{A_1, ..., A_n\}$ is a subset of the Cartesian product of the domains of all attributes, i.e.,

$$\mathbf{r} \subseteq Dom_1 \times Dom_2 \times \ldots \times Dom_n$$

Domains

- A domain is a *type* in the programming language sense
  - ➡ Name: String
  - ➡ Salary: Real
- Domain values is a set of acceptable values for a variable of a given type.
  - ➡ Name: CdnNames = {…},
  - ➡ Salary: ProfSalary = {45,000 - 150,000}
  - ➡ Simple/Composite domains
    - ✦ Address = Street name+street number+city+province+ postal code
- Domain compatibility
  - ➡ Binary operations (e.g., comparison to one another, addition, etc.) can be performed on them.
- Full support for domains is not provided in many current relational DBMSs

Relation Schemes

EMP

<u>ENO</u>	ENAME	TITLE	SAL	<u>PNO</u>	RESP	DUR

PROJ

<u>PNO</u>	PNAME	BUDGET

EMP(<u>ENO</u>, ENAME, TITLE, SAL, <u>PNO</u>, RESP, DUR)

PROJ (<u>PNO</u>, PNAME, BUDGET)

- Underlined attributes are relation keys (tuple identifiers).
- Tabular form

Example Relation Instances

EMP

ENO	ENAME	TITLE	SAL	PNO	RESP	DUR
E1	J. Doe	Elect. Eng.	40000	P1	Manager	12
E2	M. Smith	Analyst	34000	P1	Analyst	24
E2	M. Smith	Analyst	34000	P2	Analyst	6
E3	A. Lee	Mech. Eng.	27000	P3	Consultant	10
E3	A. Lee	Mech. Eng.	27000	P4	Engineer	48
E4	J. Miller	Programmer	24000	P2	Programmer	18
E5	B. Casey	Syst. Anal.	34000	P2	Manager	24
E6	L. Chu	Elect. Eng.	40000	P4	Manager	48
E7	R. Davis	Mech. Eng.	27000	P3	Engineer	36
E8	J. Jones	Syst. Anal.	34000	P3	Manager	40

PROJ

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

Repetition Anomaly

- The NAME,TITLE, SAL attribute values are repeated for each project that the employee is involved in.
  - ➡ Waste of space
  - ➡ Complicates updates

EMP

ENO	ENAME	TITLE	SAL	PNO	RESP	DUR
E1	J. Doe	Elect. Eng.	40000	P1	Manager	12
E2	M. Smith	Analyst	34000	P1	Analyst	24
E2	M. Smith	Analyst	34000	P2	Analyst	6
E3	A. Lee	Mech. Eng.	27000	P3	Consultant	10
E3	A. Lee	Mech. Eng.	27000	P4	Engineer	48
E4	J. Miller	Programmer	24000	P2	Programmer	18
E5	B. Casey	Syst. Anal.	34000	P2	Manager	24
E6	L. Chu	Elect. Eng.	40000	P4	Manager	48
E7	R. Davis	Mech. Eng.	27000	P3	Engineer	36
E8	J. Jones	Syst. Anal.	34000	P3	Manager	40

Update Anomaly

- If any attribute of project (say SAL of an employee) is updated, multiple tuples have to be updated to reflect the change.

EMP

ENO	ENAME	TITLE	SAL	PNO	RESP	DUR
E1	J. Doe	Elect. Eng.	40000	P1	Manager	12
E2	M. Smith	Analyst	34000	P1	Analyst	24
E2	M. Smith	Analyst	34000	P2	Analyst	6
E3	A. Lee	Mech. Eng.	27000	P3	Consultant	10
E3	A. Lee	Mech. Eng.	27000	P4	Engineer	48
E4	J. Miller	Programmer	24000	P2	Programmer	18
E5	B. Casey	Syst. Anal.	34000	P2	Manager	24
E6	L. Chu	Elect. Eng.	40000	P4	Manager	48
E7	R. Davis	Mech. Eng.	27000	P3	Engineer	36
E8	J. Jones	Syst. Anal.	34000	P3	Manager	40

Insertion Anomaly

- It may not be possible to store information about a new project until an employee is assigned to it.

EMP

ENO	ENAME	TITLE	SAL	PNO	RESP	DUR
E1	J. Doe	Elect. Eng.	40000	P1	Manager	12
E2	M. Smith	Analyst	34000	P1	Analyst	24
E2	M. Smith	Analyst	34000	P2	Analyst	6
E3	A. Lee	Mech. Eng.	27000	P3	Consultant	10
E3	A. Lee	Mech. Eng.	27000	P4	Engineer	48
E4	J. Miller	Programmer	24000	P2	Programmer	18
E5	B. Casey	Syst. Anal.	34000	P2	Manager	24
E6	L. Chu	Elect. Eng.	40000	P4	Manager	48
E7	R. Davis	Mech. Eng.	27000	P3	Engineer	36
E8	J. Jones	Syst. Anal.	34000	P3	Manager	40

Deletion Anomaly

- If an engineer, who is the only employee on a project, leaves the company, his personal information cannot be deleted, or the information about that project is lost.
- May have to delete many tuples.

EMP

ENO	ENAME	TITLE	SAL	PNO	RESP	DUR
E1	J. Doe	Elect. Eng.	40000	P1	Manager	12
E2	M. Smith	Analyst	34000	P1	Analyst	24
E2	M. Smith	Analyst	34000	P2	Analyst	6
E3	A. Lee	Mech. Eng.	27000	P3	Consultant	10
E3	A. Lee	Mech. Eng.	27000	P4	Engineer	48
E4	J. Miller	Programmer	24000	P2	Programmer	18
E5	B. Casey	Syst. Anal.	34000	P2	Manager	24
E6	L. Chu	Elect. Eng.	40000	P4	Manager	48
E7	R. Davis	Mech. Eng.	27000	P3	Engineer	36
E8	J. Jones	Syst. Anal.	34000	P3	Manager	40

What to do?

- Take each relation individually and "improve" it in terms of the desired characteristics
  - Normal forms
    - Atomic values (1NF)
    - Can be defined according to keys and dependencies.
    - Functional Dependencies (2NF, 3NF, BCNF)
    - Multivalued dependencies (4NF)
  - Normalization
    - Normalization is a process of concept separation which applies a top-down methodology for producing a schema by *subsequent refinements and decompositions*.
    - Do not combine unrelated sets of facts in one table; each relation should contain an independent set of facts.
    - Universal relation assumption
    - 1NF to 3NF; 1NF to BCNF

Normalization Issues

- How do we decompose a schema into a desirable normal form?
- What criteria should the decomposed schemas follow in order to preserve the semantics of the original schema?
  - ➡ Reconstructability: recover the original relation ⇒ no spurious joins
  - ➡ Lossless decomposition: no information loss
  - ➡ Dependency preservation: the constraints (i.e., dependencies) that hold on the original relation should be enforceable by means of the constraints (i.e., dependencies) defined on the decomposed relations.
- What happens to queries?
  - ➡ Processing time may increase due to joins
  - ➡ Denormalization

Functional Dependence

- Given relation R defined over $U = \{A_1, A_2, ..., A_n\}$ where $X \subseteq U$, $Y \subseteq U$. If, for all pairs of tuples t_1 and t_2 in any legal instance of relation scheme R,

  $$t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y],$$

  then the functional dependency $X \rightarrow Y$ holds in R.
- Example
  - In relation EMP
    - (ENO, PNO) → (ENAME, TITLE, SAL, DUR, RESP)
  - In relation PROJ
    - PNO → (PNAME, BUDGET)

Normal Forms Based on FDs

1NF eliminates the relations within relations or relations as attributes of tuples.

First Normal Form (1NF)

eliminate the partial functional dependencies of non-prime attributes to key attributes

Lossless & Dependency preserving

Second Normal Form (2NF)

eliminate the transitive functional dependencies of non-prime attributes to key attributes

Third Normal Form (3NF)

Lossless

eliminate the partial and transitive functional dependencies of prime (key) attributes to key.

Boyce-Codd Normal Form (BCNF)

Normalized Relations – Example

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

Relational Algebra

Specify how to obtain the result using a set of operators

Form

$$\langle Operator\rangle_{\langle parameters\rangle}\langle Operands\rangle \rightarrow \langle Result\rangle$$

$$\downarrow \qquad\qquad \downarrow$$

Relation (s) Relation

Relational Algebra Operators

- Fundamental
  - Selection
  - Projection
  - Union
  - Set difference
  - Cartesian product
- Additional
  - Intersection
  - θ-join
  - Natural join
  - Semijoin
  - Division
- Union compatibility
  - Same degree
  - Corresponding attributes defined over the same domain

Selection

- Produces a horizontal subset of the operand relation
- General form

$$\sigma_F(R)=\{t \mid t\in R \text{ and } F(t) \text{ is true}\}$$

where

- ➡ R is a relation, t is a tuple variable
- ➡ F is a formula consisting of
  - ✦ operands that are constants or attributes
  - ✦ arithmetic comparison operators

    $<, >, =, \neq, \leq, \geq$

  - ✦ logical operators

    $\wedge, \vee, \neg$

Selection Example

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.

$\sigma_{\text{TITLE='Elect. Eng.'}}(\text{EMP})$

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng
E6	L. Chu	Elect. Eng.

Projection

- Produces a vertical slice of a relation
- General form

$$\Pi_{A_1,\ldots,A_n}(R)=\{t[A_1,\ldots, A_n] \mid t\in R\}$$

where

  - ➡ R is a relation, t is a tuple variable
  - ➡ $\{A_1,\ldots, A_n\}$ is a subset of the attributes of R over which the projection will be performed

- Note: projection can generate duplicate tuples. Commercial systems (and SQL) allow this and provide
  - ➡ Projection with duplicate elimination
  - ➡ Projection without duplicate elimination

Projection Example

PROJ

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

$\Pi_{\text{PNO,BUDGET}}(\text{PROJ})$

PNO	BUDGET
P1	150000
P2	135000
P3	250000
P4	310000

Union

- Similar to set union
- General form

$$R \cup S = \{t \mid t \in R \text{ or } t \in S\}$$

where R, S are relations, t is a tuple variable

  - ➡ Result contains tuples that are in R or in S, but not both (duplicates removed)
  - ➡ R, S should be union-compatible

Set Difference

- General Form

$$R - S = \{t \mid t \in R \text{ and } t \notin S\}$$

where R and S are relations, t is a tuple variable

- ➡ Result contains all tuples that are in R, but not in S.
- ➡ $R - S \neq S - R$
- ➡ R, S union-compatible

Cartesian (Cross) Product

- Given relations
  - R of degree k_1, cardinality n_1
  - S of degree k_2, cardinality n_2
- Cartesian (cross) product:

$$R \times S = \{t\,[A_1,\ldots,A_{k_1}, A_{k_1+1},\ldots,A_{k_1+k_2}] \mid t[A_1,\ldots,A_{k_1}] \in R \text{ and } t[A_{k_1+1},\ldots,A_{k_1+k_2}] \in S\}$$

The result of $R \cdot S$ is a relation of degree $(k_1 + k_2)$ and consists of all $(n_1 * n_2)$-tuples where each tuple is a concatenation of one tuple of R with one tuple of S.

Cartesian Product Example

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.

PAY

TITLE	SALARY
Elect. Eng.	55000
Syst. Anal.	70000
Mech. Eng.	45000
Programmer	60000

EMP × PAY

ENO	ENAME	EMP.TITLE	PAY.TITLE	SALARY
E1	J. Doe	Elect. Eng.	Elect. Eng.	55000
E1	J. Doe	Elect. Eng.	Syst. Anal.	70000
E1	J. Doe	Elect. Eng.	Mech. Eng.	45000
E1	J. Doe	Elect. Eng.	Programmer	60000
E2	M. Smith	Syst. Anal.	Elect. Eng.	55000
E2	M. Smith	Syst. Anal.	Syst. Anal.	70000
E2	M. Smith	Syst. Anal.	Mech. Eng.	45000
E2	M. Smith	Syst. Anal.	Programmer	60000
E3	A. Lee	Mech. Eng.	Elect. Eng.	55000
E3	A. Lee	Mech. Eng.	Syst. Anal.	70000
E3	A. Lee	Mech. Eng.	Mech. Eng.	45000
E3	A. Lee	Mech. Eng.	Programmer	60000
E8	J. Jones	Syst. Anal.	Elect. Eng.	55000
E8	J. Jones	Syst. Anal.	Syst. Anal.	70000
E8	J. Jones	Syst. Anal.	Mech. Eng.	45000
E8	J. Jones	Syst. Anal.	Programmer	60000

Intersection

- Typical set intersection

$$R \cap S = \{t \mid t \in R \text{ and } t \in S\}$$

$$= R - (R - S)$$

- *R*, *S* union-compatible

θ-Join

- General form

  $R \bowtie_{F(R.A_i, S.B_j)} S = \{t[A_1,\ldots,A_n, B_1,\ldots,B_m] \mid$

  $t[A_1,\ldots,A_n] \in R$ and $t[B_1,\ldots,B_m] \in S$

  and $F(R.A_i, S.B_j)$ is true$\}$

  where

  - ➡ R, S are relations, t is a tuple variable
  - ➡ $F(R.A_i, S.B_j)$is a formula defined as that of selection.

- A derivative of Cartesian product
  - ➡ $R \bowtie_F S = \sigma_F(R \times S)$

Join Example

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.
E9	A. Hsu	Programmer
E10	T. Wong	Syst. Anal.

(a)

EMP $\bowtie_{\text{EMP.ENO=ASG.ENO}}$ ASG

ENO	ENAME	TITLE	PNO	RESP	DUR
E1	J. Doe	Elect. Eng.	P1	Manager	12
E2	M. Smith	Syst. Anal.	P1	Analyst	12
E2	M. Smith	Syst. Anal.	P2	Analyst	12
E3	A. Lee	Mech. Eng.	P3	Consultant	12
E3	A. Lee	Mech. Eng.	P4	Engineer	12
E4	J. Miller	Programmer	P2	Programmer	12
E5	J. Miller	Syst. Anal.	P2	Manager	12
E6	L. Chu	Elect. Eng.	P4	Manager	12
E7	R. Davis	Mech. Eng.	P3	Engineer	12
E8	J. Jones	Syst. Anal.	P3	Manager	12

(b)

Types of Join

- Equi-join
  - ➡ The formula F only contains equality
  - ➡ $R \bowtie_{R.A=S.B} S$
- Natural join
  - ➡ Equi-join of two relations R and S over an attribute (or attributes) common to both R and S and projecting out one copy of those attributes
  - ➡ $R \bowtie S = \Pi_{R \cup S}\sigma_F(R \times S)$

Natural Join Example

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.

PAY

TITLE	SALARY
Elect. Eng.	55000
Syst. Anal.	70000
Mech. Eng.	45000
Programmer	60000

EMP $\bowtie$ PAY

ENO	ENAME	TITLE	SALARY
E1	J. Doe	Elect. Eng.	55000
E2	M. Smith	Analyst	70000
E3	A. Lee	Mech. Eng.	45000
E4	J. Miller	Programmer	60000
E5	B. Casey	Syst. Anal.	70000
E6	L. Chu	Elect. Eng.	55000
E7	R. Davis	Mech. Eng.	45000
E8	J. Jones	Syst. Anal.	70000

Join is over the common attribute TITLE

Types of Join

- Outer-Join
  - ➡ Ensures that tuples from one or both relations that do not satisfy the join condition still appear in the final result with other relation's attribute values set to NULL
  - ➡ Left outer join $⟕$
  - ➡ Right outer join $⟖$
  - ➡ Full outer join $⟗$

Outer Join Example

- Left outer join

$EMP \mathbin{⟕}_{ENO} ASG$

ENO	ENAME	TITLE	PNO	RESP	DUR
E1	J. Doe	Elect. Eng.	P1	Manager	12
E2	M. Smith	Syst. Anal.	P1	Analyst	12
E2	M. Smith	Syst. Anal.	P2	Analyst	12
E3	A. Lee	Mech. Eng.	P3	Consultant	12
E3	A. Lee	Mech. Eng.	P4	Engineer	12
E4	J. Miller	Programmer	P2	Programmer	12
E5	J. Miller	Syst. Anal.	P2	Manager	12
E6	L. Chu	Elect. Eng.	P4	Manager	12
E7	R. Davis	Mech. Eng.	P3	Engineer	12
E8	J. Jones	Syst. Anal.	P3	Manager	12
E9	A. Hsu	Programmer	Null	Null	Null
E10	T. Wong	Syst. Anal.	Null	Null	Null

Semijoin

- Derivation

  $R \ltimes_F S = \Pi_A(R \bowtie_F S) = \Pi_A(R) \bowtie \Pi_{A \cap B}(S) = R \bowtie_F \Pi_{A \cap B}(S)$

  where

  - R, S are relations
  - A is a set of attributes

Semijoin Example

$EMP \ltimes_{EMP.TITLE=PAY.TITLE} PAY$

ENO	ENAME	TITLE
E1	J. Doe	Elect. Eng.
E2	M. Smith	Analyst
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.

Division (Quotient)

- Given relations
  - ➡ R of degree k_1 ($R = \{A_1,\ldots,A_{k_1}\}$)
  - ➡ S of degree k_2 ($S = \{B_1,\ldots,B_{k_2}\}$)

  Let $A = \{A_1,\ldots,A_{k_1}\}$ [i.e., $R(A)$] and $B = \{B_1,\ldots,B_{k_2}\}$ [i.e., $S(B)$] and $B \subseteq A$.

  Then, $T = R \div S$ gives T of degree k_1-k_2 [i.e., $T(Y)$ where $Y = A$-B] such that for a tuple t to appear in T, the values in t must appear in R in combination with *every tuple* in S.

- Derivation

  $R \div S = \Pi_Y(R) - \Pi_Y((\Pi_Y(R) \times S) - R)$

Division Example

ASG'

ENO	PNO	PNAME	BUDGET
E1	P1	Instrumentation	150000
E2	P1	Instrumentation	150000
E2	P2	Database Develop.	135000
E3	P3	CAD/CAM	250000
E3	P4	Maintenance	310000
E4	P2	Database Develop.	135000
E5	P2	Database Develop.	135000
E6	P4	Maintenance	310000
E7	P3	CAD/CAM	250000
E8	P3	CAD/CAM	250000

PROJ'

PNO	PNAME	BUDGET
P3	CAD/CAM	250000
P4	Maintenance	310000

(ASG' ÷ PROJ')

ENO
E3

Relational Calculus

- Specify the properties that the result should hold
- Tuple relational calculus
- Domain relational calculus

Tuple Relational Calculus

- Query of the form $\{t \mid F\{t\}\}$ where
  - ➡ t is a tuple variable
  - ➡ F is a well-formed formula
- Atomic formula
  - ➡ Tuple-variable membership expressions
    - ✦ $R.t$ or $R(t)$: tuple t belongs to relation R
  - ➡ Conditions
    - ✦ $s[A]\ \theta\ t[B]$; s and t are tuple variables, A and B are components of s and t, respectively, $\theta \in \{<, >, =, \neq, \leq, \geq\}$; e.g., s[SAL] > t[SAL]
    - ✦ $s[A]\ \theta\ c$; s, A, and θ as defined above, c is a constant; e.g., s[ENAME] = 'Smith'
- SQL is an example of tuple relational calculus (at least in its simple form)

Domain Relational Calculus

- Query of the form $x_1, x_2, \ldots, x_n \mid F(x_1, x_2, \ldots, x_n)$ where
  - ➡ F is a well-formed formula in which $x_1, x_2, \ldots, x_n$ are the free variables
- QBE is an example

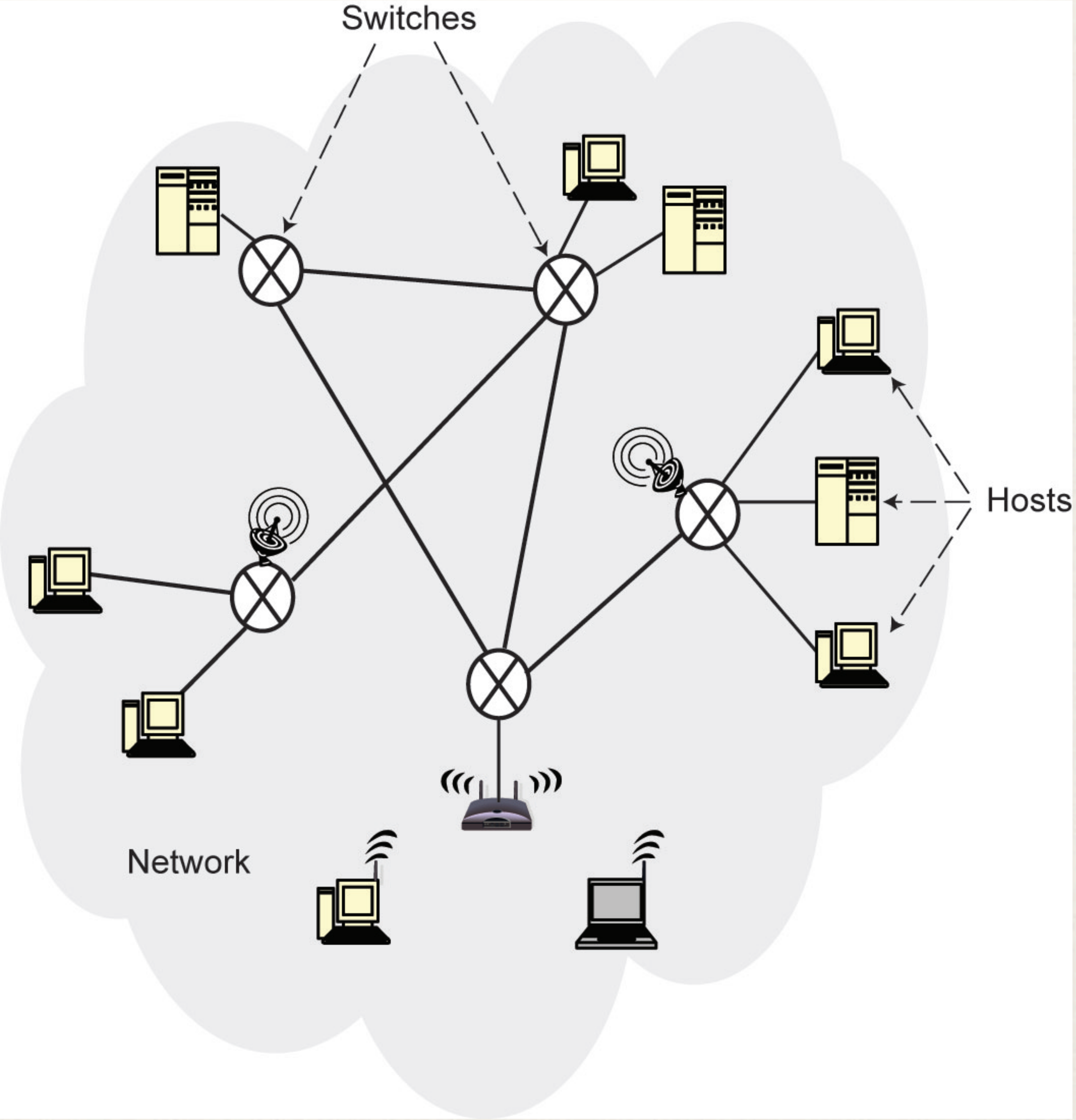
EMP	ENO	ENAME	TITLE
	E2	P.	

ASG	ENO	PNO	RESP	DUR
	E2	P3		

PROJ	PNO	PNAME	BUDGET
	P3	CAD/CAM	

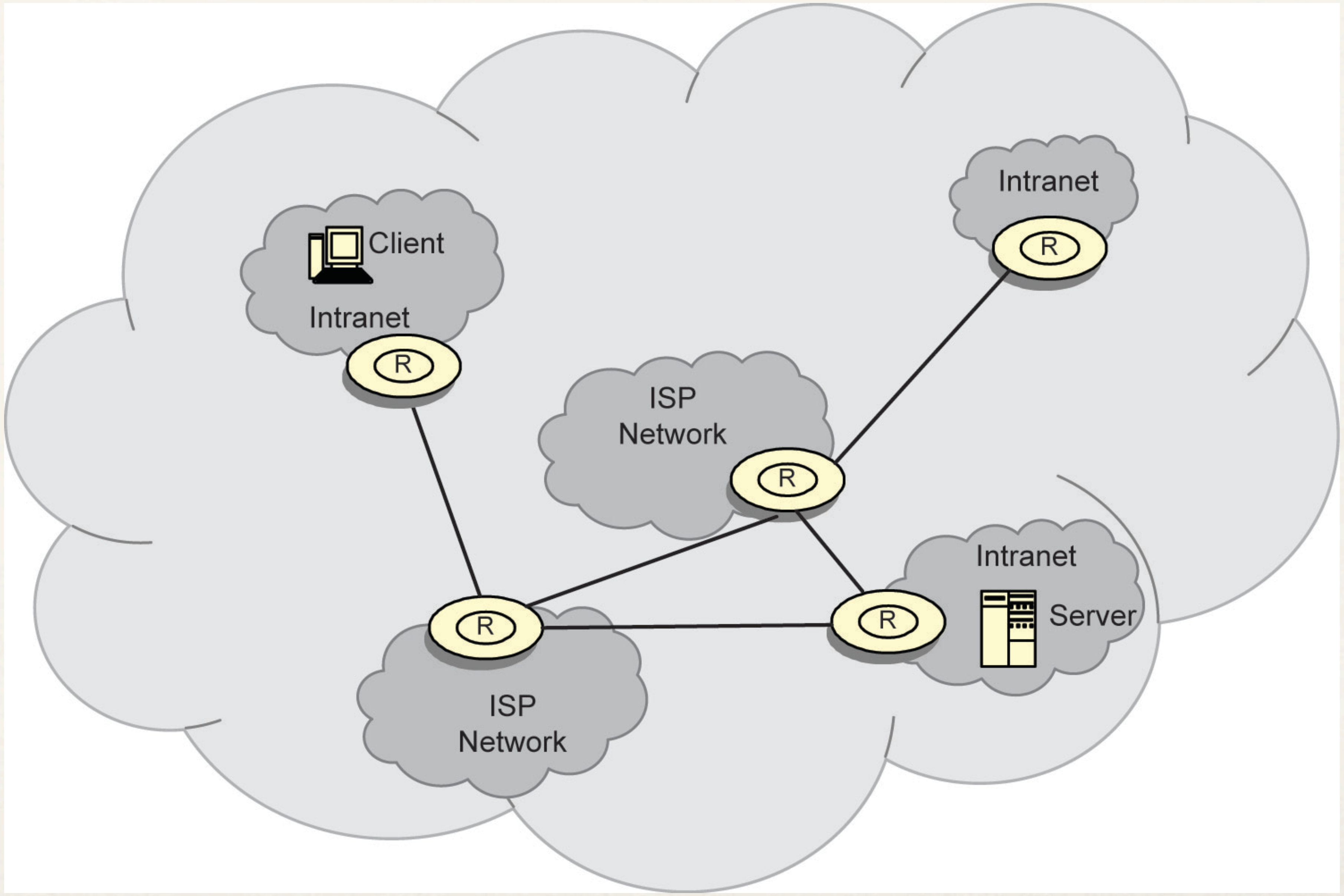
Computer Network

- 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



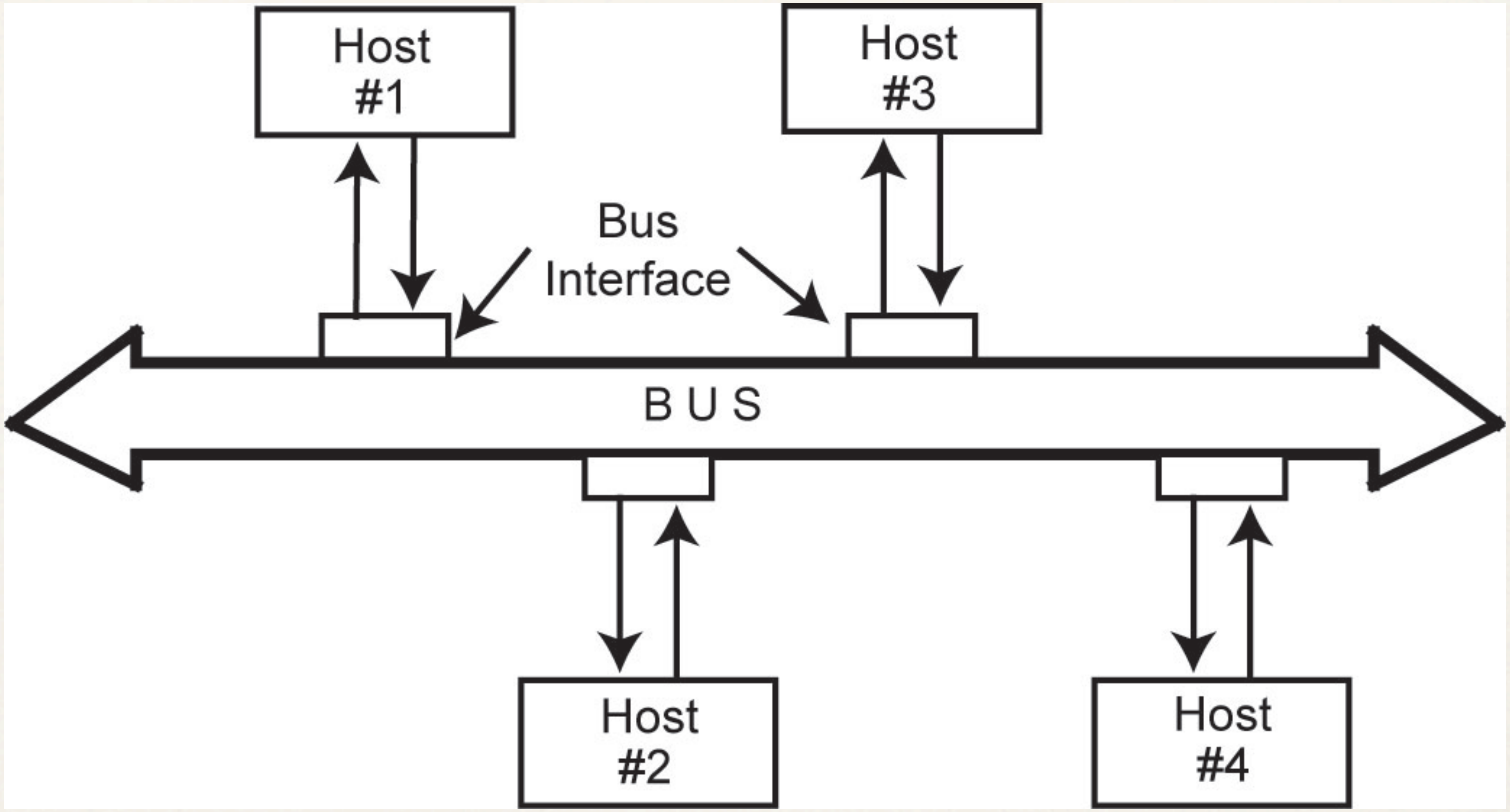
Types of Networks

- According to scale (geographic distribution)
  - ➡ Wide are network (WAN)
    - ✦ Distance between any two nodes > 20km and can go as high as thousands of kms
    - ✦ Long delays due to distance traveled
    - ✦ Heterogeneity of transmission media
    - ✦ Speeds of 150Mbps to 10Gbps (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

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

Bus network



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

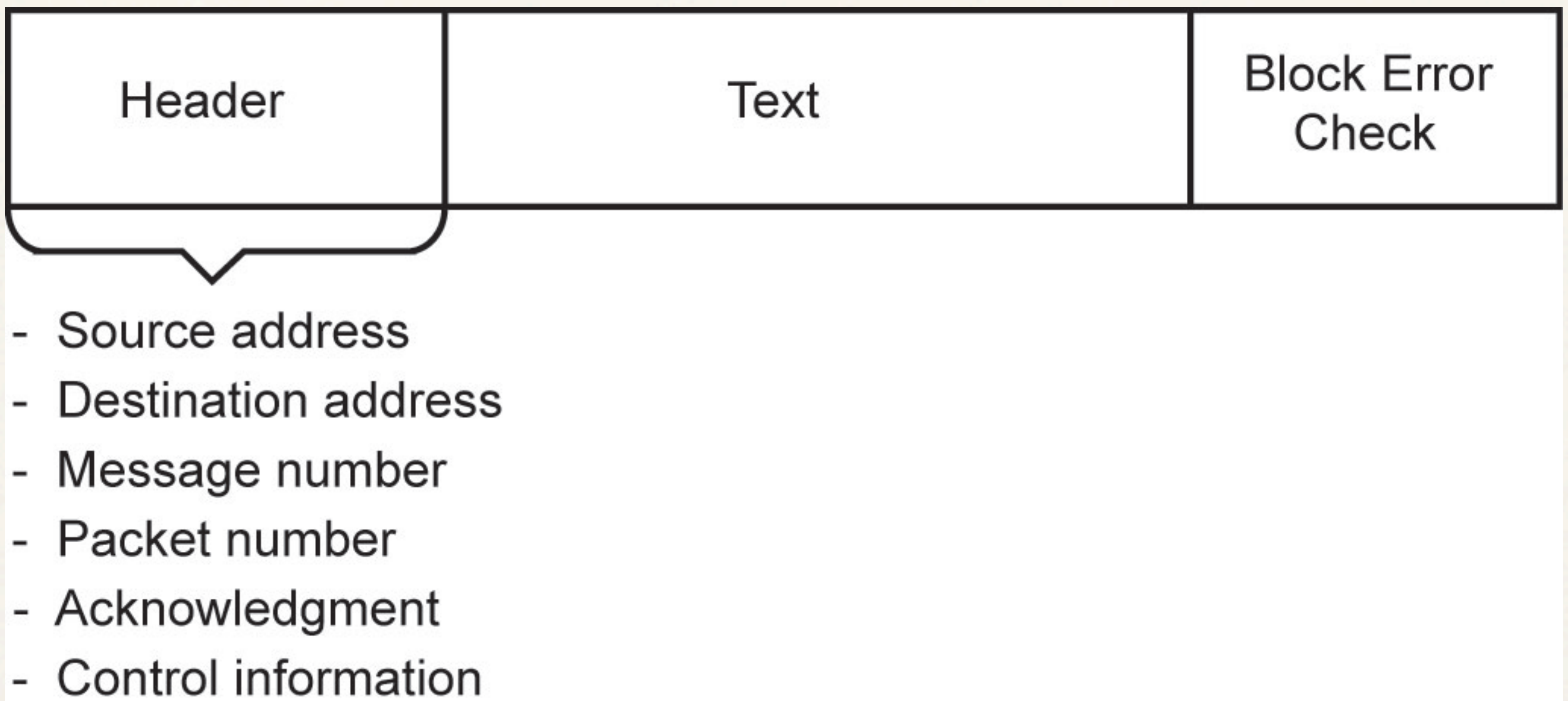
Communication Alternatives

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

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 trnsmitted 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

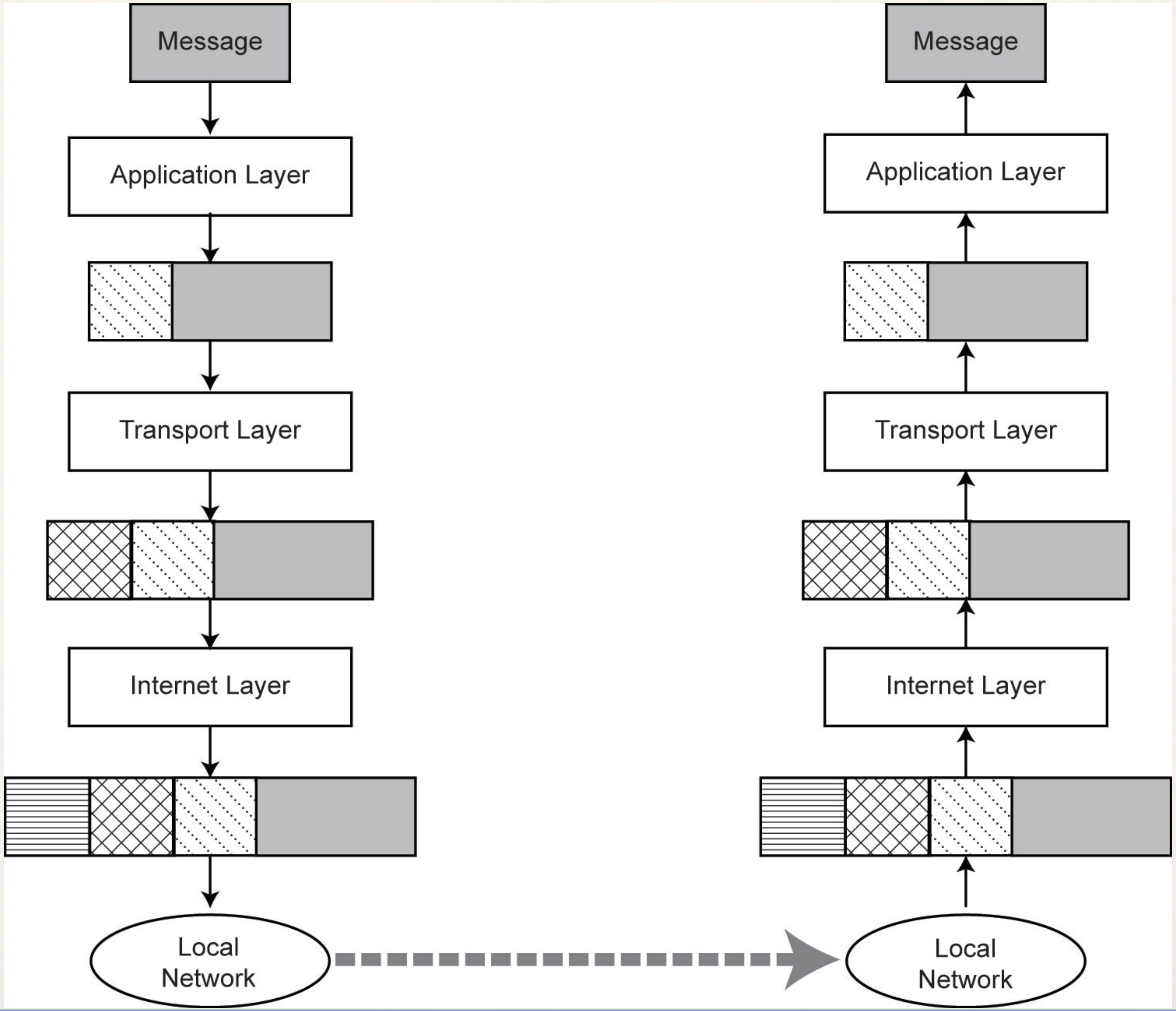
Packet Format



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

Message Transmission using TCP/IP



TCP/IP Protocol