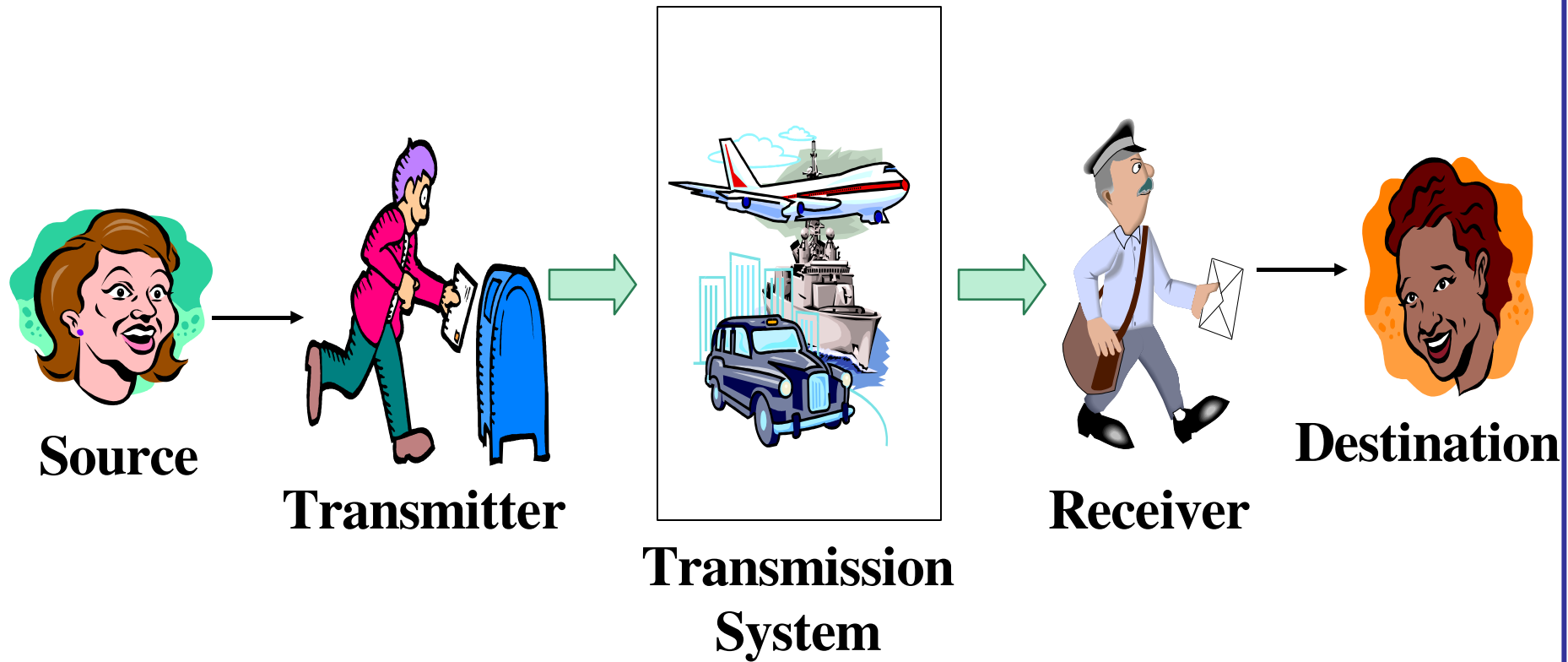


Introduction to Basics of Communication Protocol

Prof Pallapa. Venkataram
Department of Electrical Communication Engineering
Indian Institute of Science
Bangalore – 560012, India



Physical Communication Model



Communication

It is any transmission, emission, or reception of signs, signals, writings, images and sounds or intelligence of any nature by wire, radio, optical or other electromagnetic system

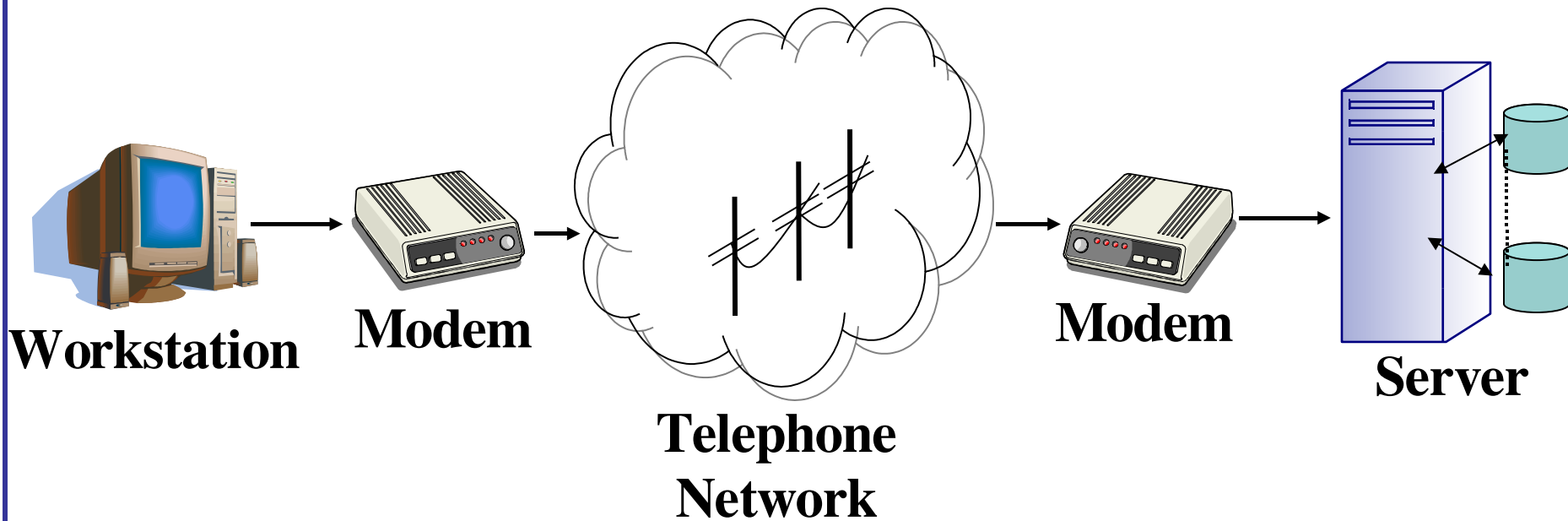
Communication model

A communication model comprises of following subsystems:

- Source – Generates the data to be transmitted
- Transmitter – Converts the data into transmittable signals
- Transmission system – Carries the data.
- Receiver – Converts received signals into data.
- Destination – Takes incoming data.



Electronics Communication Model

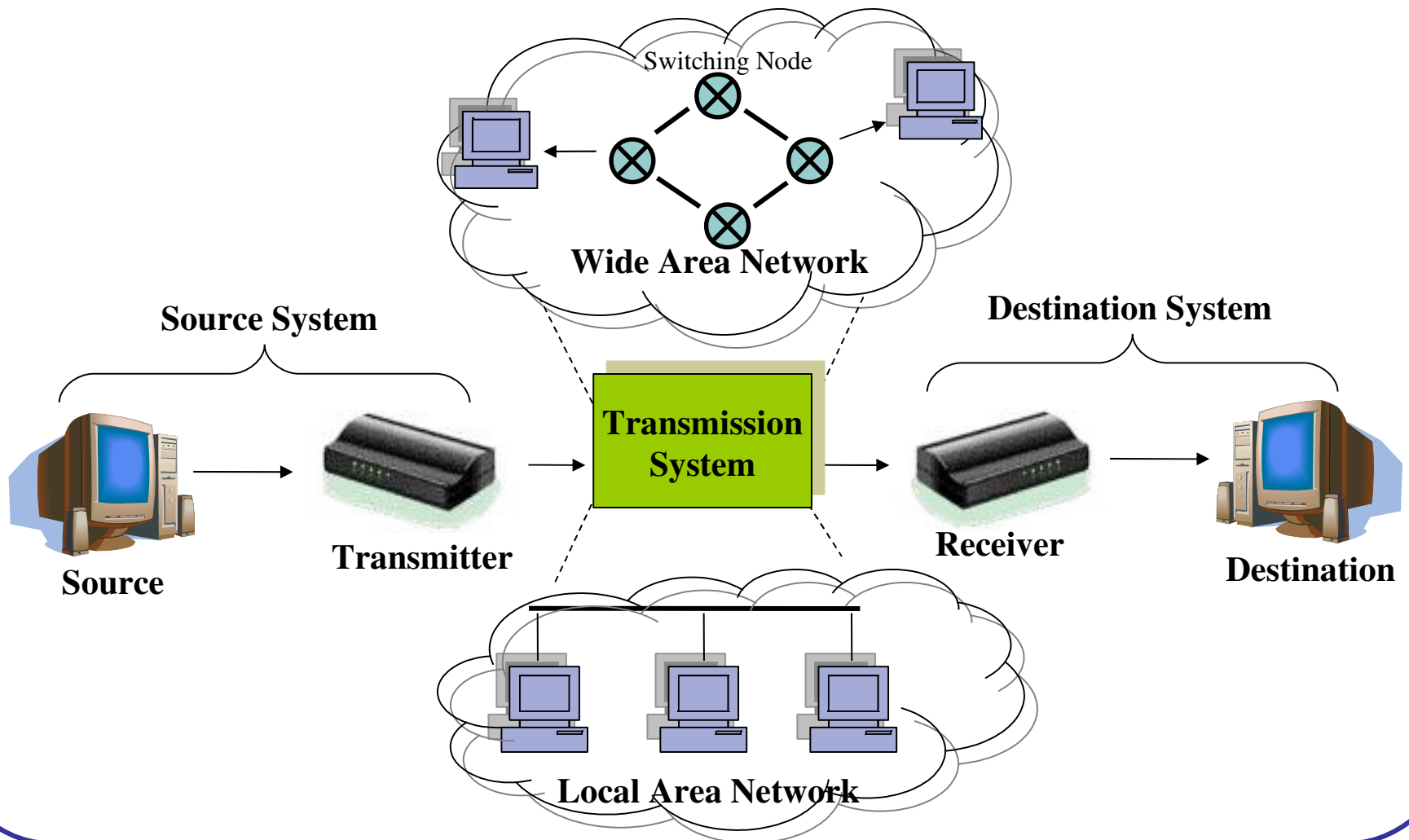


The key communication tasks are:

- Transmission system utilization
- Interfacing
- Signal generation
- Synchronization
- Exchange management
- Error detection and correction
- Addressing and routing
- Security and network management



Network Model



Communication Software

It is a set of computer programs which are used for generation, transmission and reception of information between/among the set of computers connected by wired or wireless media.

Different Kinds of Programs for Communication are:

Dialing: Dialing software tells the computer, how to place a call on the phone line connected to it.

File Transfer: Transfer of files, from one to the other
(Downloading / Uploading of files)

Terminal Emulation: Programs running on a network that connects to a computer bulletin board system

Data Encryption: Program to encode data



Communication Subsystems

- The design and implementation of large scale communication systems requires the interconnection and integration of diverse subsystems.
- The main purpose of the communication environment is to keep all the subsystems informed of any changes in the design parameters.
 1. Change reported from one of the subsystems.
 2. Request for data from one subsystem to another



Communication between two system

User A



User B



Hi, I want to chat with you

Ok. I am ready

Dialogue between A & B

Hi, Let us finish chatting

Ok. Let us disconnect

Protocol

- It is a set of rules governing the format and meaning of frames, packets, or messages that are exchanged by peer entities within a layer.
- Protocol are used for communications between entities in a systems.
- Entities use protocols in order to implement their service definitions.

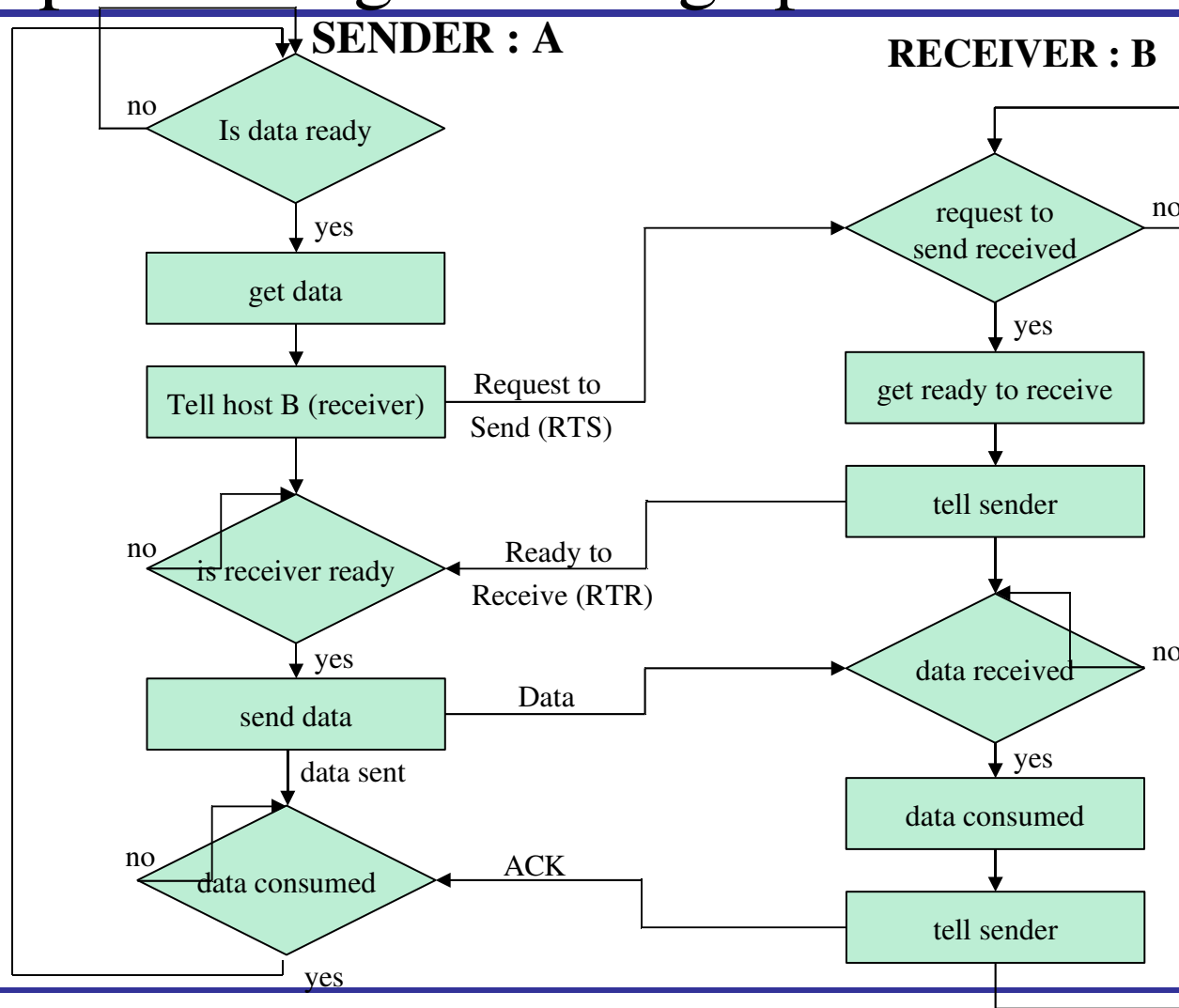
The key elements of a protocol are:

Syntax : Include Time data formats and signal levels

Semantics: Includes control information and error handling.



Simple message exchange protocol flowchart



Representation of Protocol

The process of a protocol like, sender and receiver protocol can be formally specified using

2. Finite State Machine (FSM/PETRINET)
3. State transition table

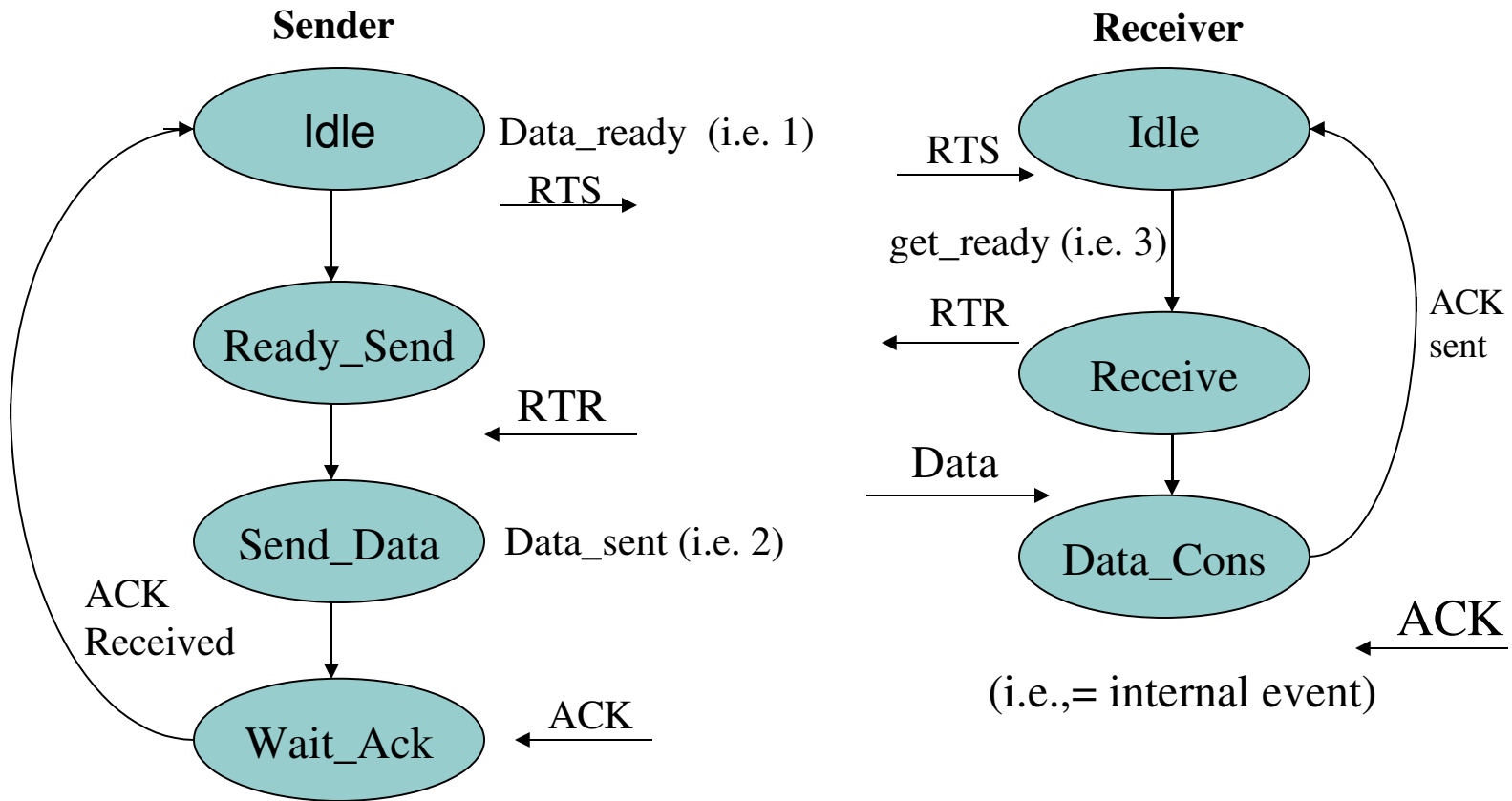
Finite State Machine (FSM)

- A : Finite state machine M is a quintuple, $M=(I; O; S; N; A)$
- I : Finite and nonempty sets of input symbols
- O : Finite and nonempty sets of output symbols
- S : Finite and nonempty sets of states

Figure gives FSM of simple message exchange protocol



FSM of simple message exchange Protocol



State Transition Table

FSM state and transitions can be represented in the form of table called as state transition tables.

Input State	i.e. 1 (data_ready)	RTR	i.e. 2 (data_ready)	ACK
S1	S2/RTS	S1/-	S1/-	S1/-
S2	S2/-	S3/-	S2/-	S2/-
S3	S3/-	S3/-	S4/data	S3/-
S4	S4/-	S4/-	S4/-	S1/-

Communication protocol development methods

There are two ways of design and development of Communication protocols

1. Informal
2. Formal Methods

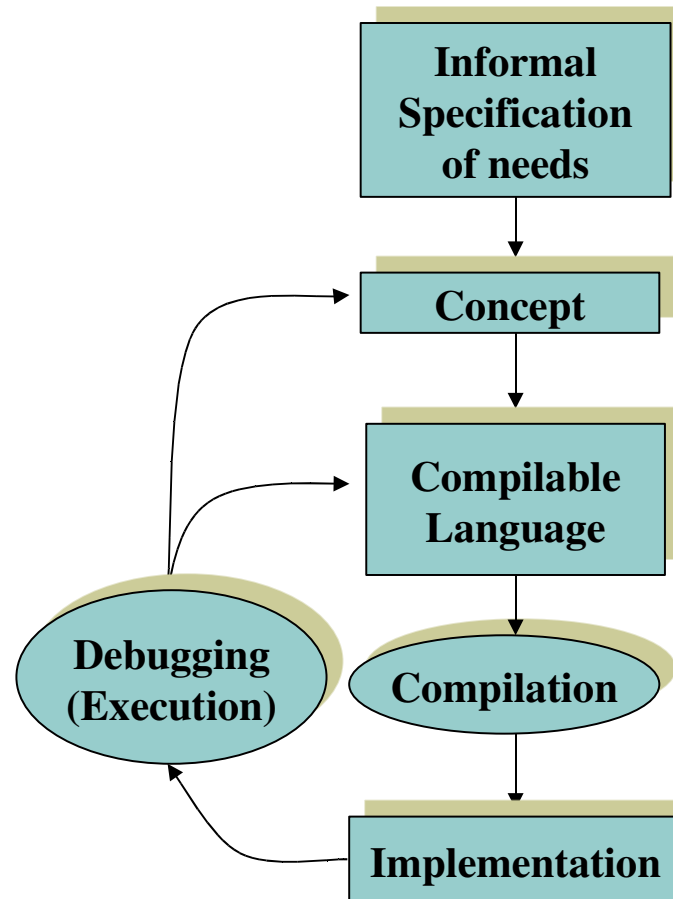


Informal methods

- Informal way of specification of the protocols include textual description.
- Not suitable for large complex protocols.
- Often leads unclear and ambiguous specifications.
- The complexity of protocols make them very hard to analyze in an informal way.

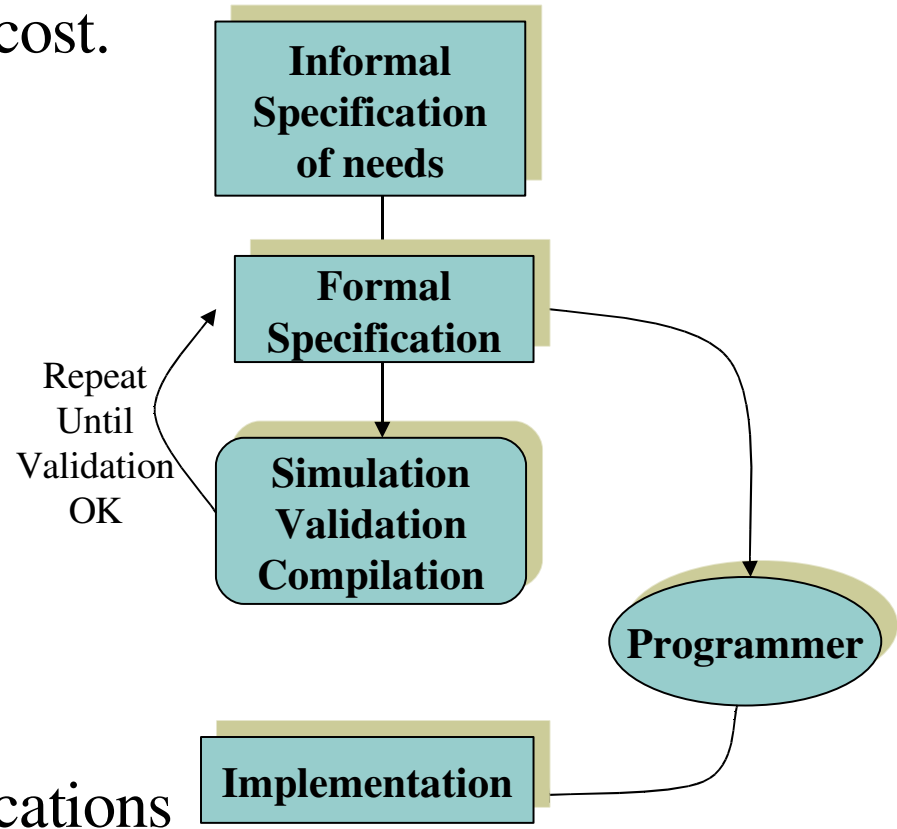


Informal Specifications



Formal methods

- Use of formal specification languages.
- These protocols must be of high quality with higher performance with low cost.



Formal Specifications

Petrinet model

A petrinet is a four tuple $(P; T; IN; OUT; M0)$ where,
 $P = \{p_1; p_2; \dots; p_n\}$ is a set of places.

$T = \{t_1; t_2; \dots; t_n\}$ is a set of transition.

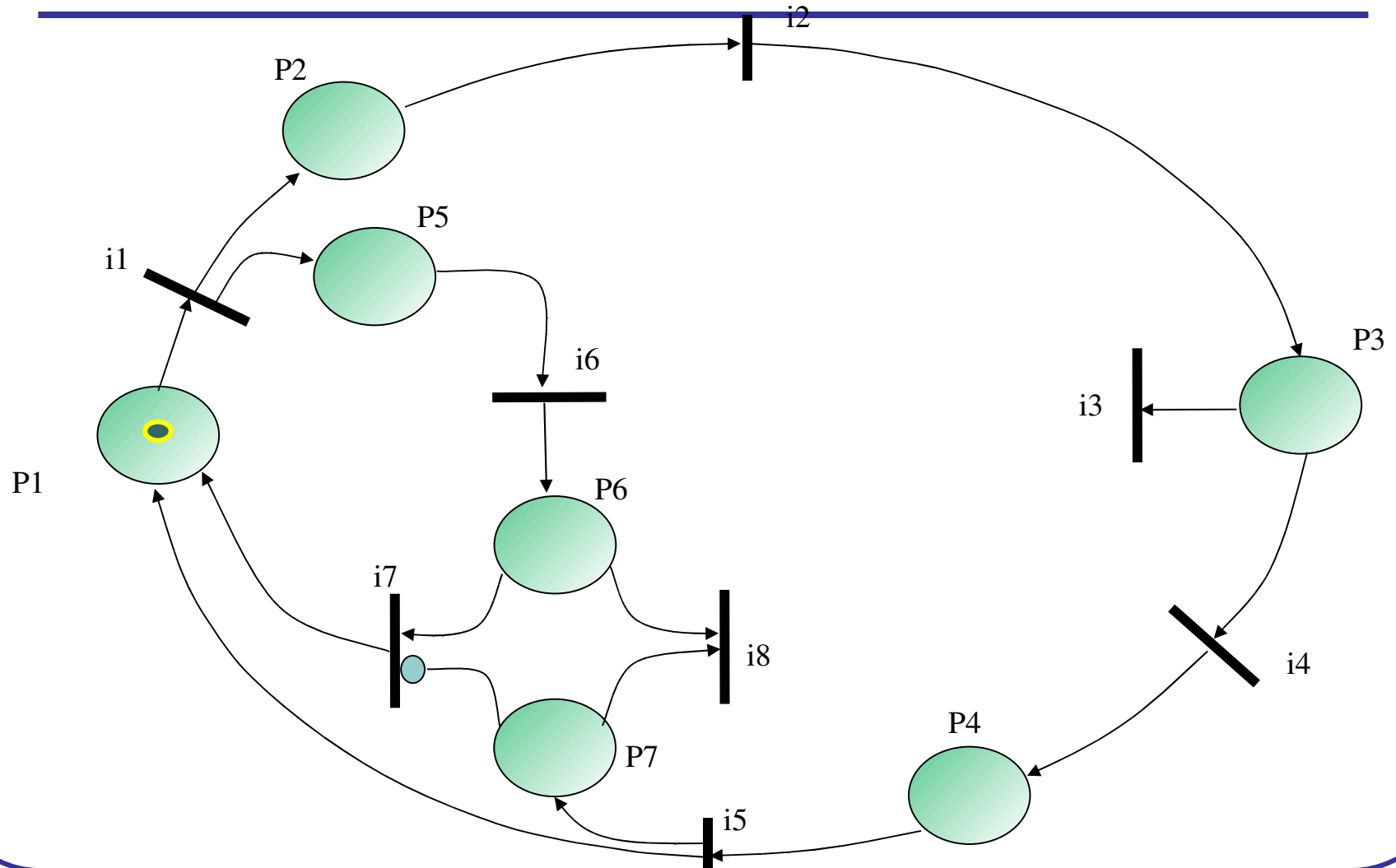
$M0$ = Initial marking (token at a place).

$IN : (P \times T) \rightarrow \mathbb{N}$ is an input function that defines directed arcs
from places to transition

$OUT : (T \times P) \rightarrow \mathbb{N}$ is an output function that defines directed arcs
From transition to places.



A Petrinet model of a simple protocol with a timeout mechanism



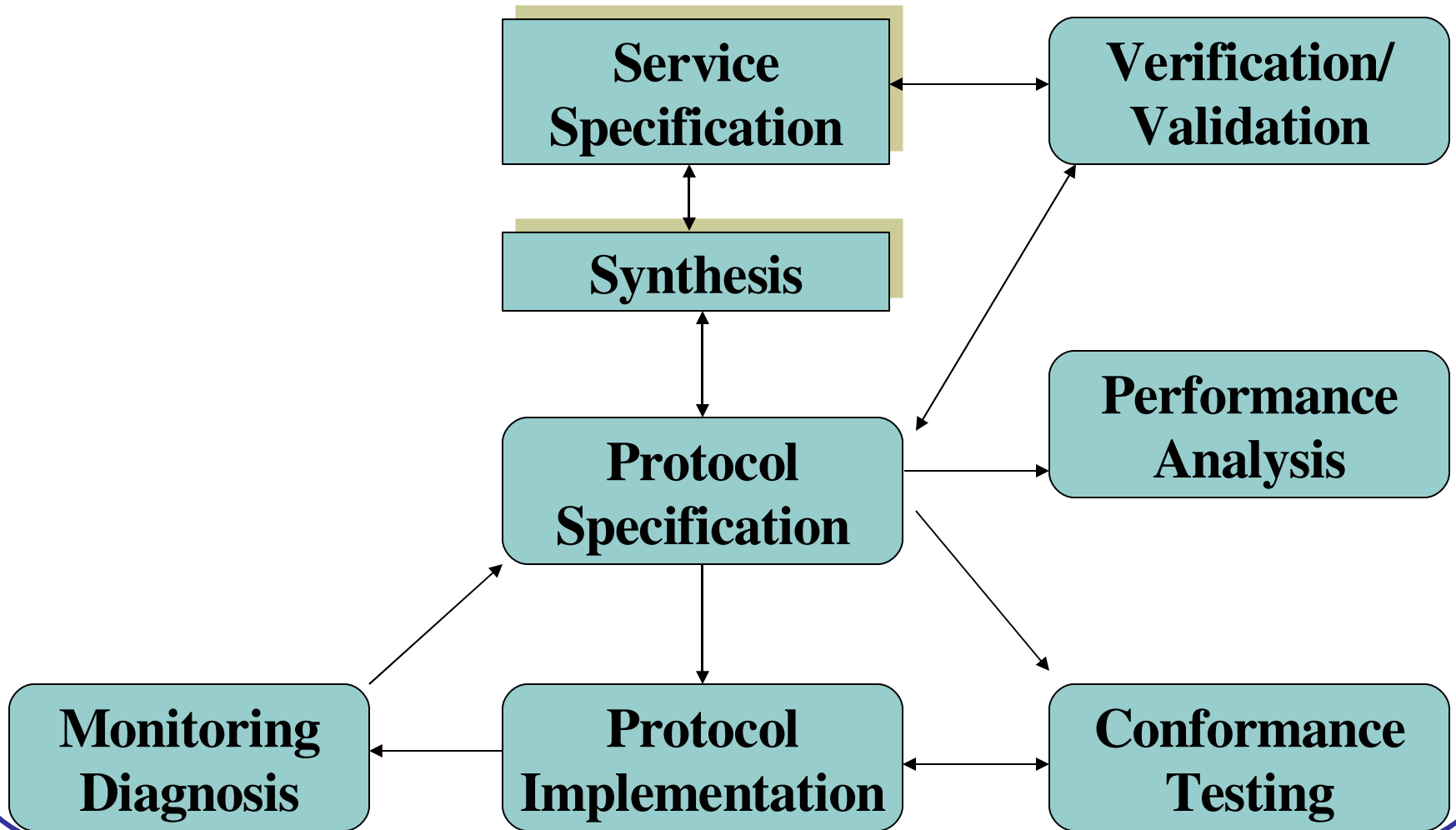
Protocol Engineering Process

Need for protocol engineering

- Many protocols if not properly designed and implemented as per the requirements, leads to improper behavior of network and system may jam the networks.
- Hence it is required to, engineer the protocols for their correctness reliability, optimized performance, reusability, and code optimization
- Deals with application of formal techniques and software engineering methodologies to protocol design and implementation



Phases of Protocol Engineering



Functions:

- **Service specification :** The service required for performing certain tasks in any given environment.
- **Synthesis :** It is an automated tool to generate the formal specification of a protocol.
- **Protocol specification :** It looks into syntax and semantics of the protocols specifications.
- **Protocol verification / validation :** User makes about the structure of possible dialogs between processes of protocol.
- **Performance analysis :** It analyses the protocol performance in terms of message complexity, time complexity, space complexity, scalability.
- **Conference testing :** it tests whether the protocol conforms to the specifications laid down in protocol specification phase by generating exhaustive set of test sequences.
- **Protocol implementation :** It deals with the real coding of the protocol using software engineering aspects.
- **Monitoring/diagnosis :** It monitors the working of implemented protocol and checks for the errors.

