CS 348: Computer Networks

- Layering; 26th July 2012

Instructor: Sridhar Iyer
IIT Bombay

Clicker Question

1. Which of the following best describes the concept of "protocol"?

   A) A protocol is a connection for peer layers of a network to communicate with each other.

   B) A protocol is a set of rules which govern the syntax and semantics of communication between corresponding layers in a network.

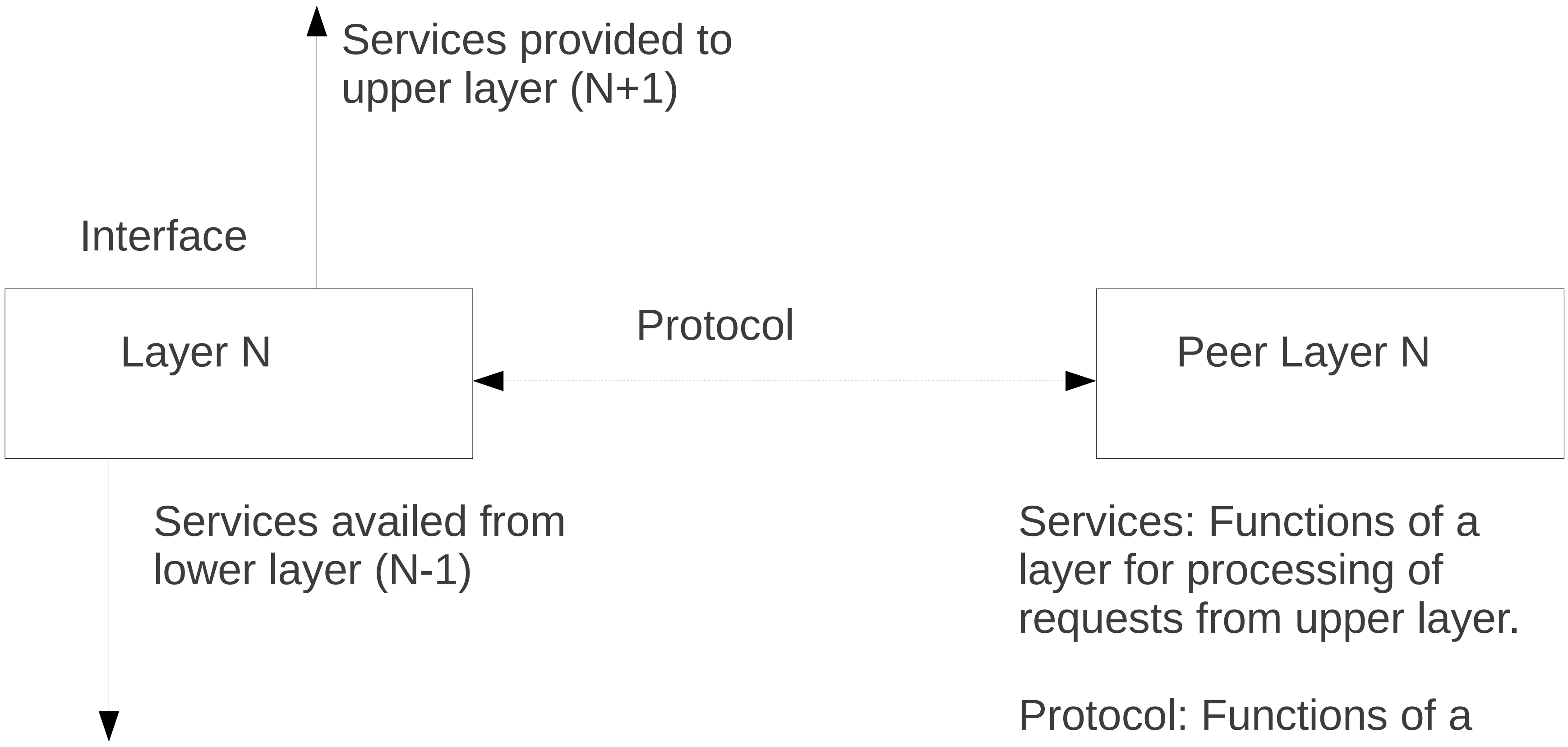
   C) A protocol defines standards that determine the format and interpretation of data in a network.

   D) A protocol is a set of services that one layer uses from layers below it.

- Choose one of the options below.

1. All of the above	2. A), B) and C)
3. B) and C)	4. D) only

Layers, Interfaces and Protocols



Services provided to upper layer (N+1)

Interface

Layer N

Protocol

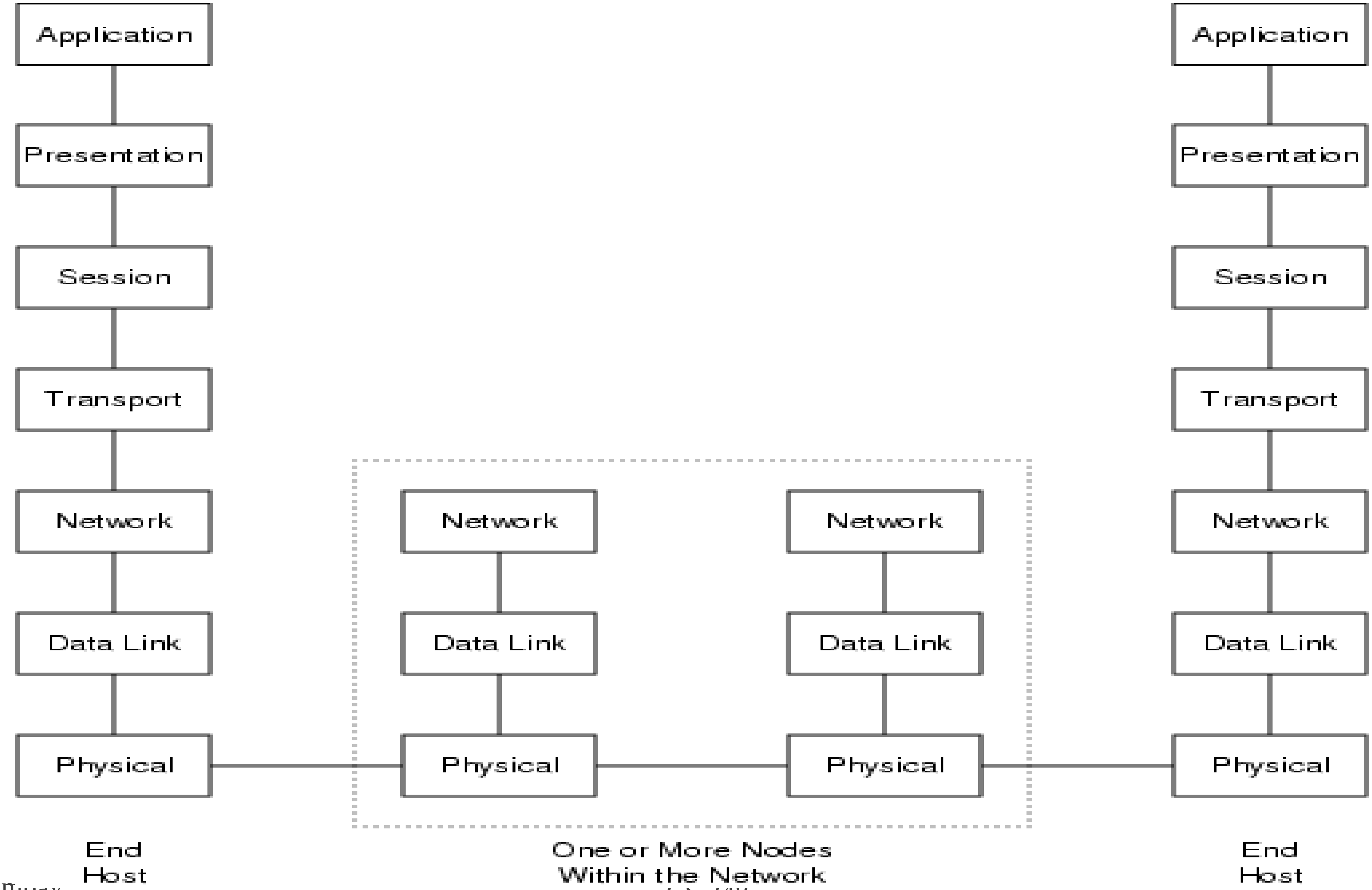
Peer Layer N

Services availed from lower layer (N-1)

Services: Functions of a layer for processing of requests from upper layer.

Protocol: Functions of a layer for understanding exchanges with peer-layer.

Technical slide on layering



Today's class

- Importance of layering in the networking context: Questions below to help you think about it.
  - What is the purpose of such a layering mechanism?
  - What will happen if we do not have it?
  - What are the pros and cons?
  - In what situations should we use layering?
  - In what situations should we not use layering?
  - Any other questions..?

Activity: Debate

- Class divided into 2 sets of groups A and B.
- Groups in set A should come up with answers to:
  - What are the advantages of using layering?
  - When should we use layering? Give examples.
- Groups in set B should come up with answers to:
  - What are the disadvantages of using layering?
  - When should we not use layering? Give examples.
- 5 minutes to discuss within groups, then debate!

Key Points regarding layering

- For using layering:
  - Pros: Abstraction, Managing complexity is easier.
  - Cons: Performance overhead (time taken and additional header size).
- For not using layering:
  - Pros: Suitable for 1-time activity on a 'small' scale.
  - Cons: When there is re-use or multiple choices.

Activity: Think-Pair-Share

- Question:
  - Can you think of examples of other areas where you have seen some concept similar to layering being applied?

- Think - Think individually for 2 minutes.
- Pair - Discuss with your neighbour.
- Share - Discuss with entire class.

- Instructor: List down answers that come up.

Clicker Question

A layer at one level on the sender side of a network communicates with layers at the ________ level on the receiver side.

1. Corresponding level
2. Higher level
3. Lower level
4. All levels
5. Either corresponding or higher levels but not lower levels

It is not sufficient to just pick an answer. You need to also think about how you will justify your answer!

Key points regarding Interfaces

- Basic idea:
  - Lower layer (N-1) provides some services (S1...Sk) to the upper layer (N).
  - Upper Layer (N) gives data (D) and control instructions (Si) to layer below (N-1).
  - Lower layer (N-1) does the processing of D (as per Si).
  - Lower layer (N-1) returns data (D') to upper layer (N).

- What is an interface definition?
  - Specifying the services (S1...Sk) provided by the layer.
  - Specifying the format (type) of D and D', for each Si.
- Another name for this concept is: **Service Access Point (SAP).**

Example: Interfaces in sending from Application to TCP

1. User space (your program): send_msg()
2. Kernel space (net/socket.c): sock_sendmsg()
3. net/ipv4/tcp.c: tcp_sendmsg()

Interface definition:

- int tcp_sendmsg

  (struct kiocb *iocb, struct socket *sock,

  struct msghdr *msg, size_t size)
- Note: Different implementations may have different interface definitions.

More about Interfaces

- Why is detailed interface specification important?
  - Class discussion:
    - Enables layering; Enables abstraction; Single point of contact.
    - Error free exchange of information.
    - …

- Maintains clarity on services being provided by a layer, while abstracting away details.
-
- Captures clearly the form of data and control to be passed between the interacting layers (entities).
-
- Brings out the assumptions being made by one layer regarding the functions of another.

Other examples of Interfaces

- Analogy: Railway ticket counter.
  - You give money and instruction regarding destination.
  - Ticket vendor does the processing.
  - You get the change and ticket.
- Another example: Function calls (subroutines).
  - Calling routine passes the parameters for the function.
  - Called routine does the processing.
  - Called routine returns the results to the calling routine.
- Yet another example: Client-Server applications

Key Points regarding protocols

- Protocol defines a layer's interaction with its peer entity across the network.
  - It is an agreement on:
    - Interpretation of data: Syntax and Semantics.
    - Sequence of control flow: Request and Response.
  - It can be thought of as an abstract (logical) connection between the peer entities.
    - The details of the lower layers are abstracted away.

More on protocols

- How does it relate to layering?
  - Layering is vertical (higher layer to lower layer on the same computer).
  - Protocol is horizontal (peer-to-peer, across the network).

- How is it different from processing at a layer?
  - Processing at a layer includes all the functionality, such as handling of Service Interfaces, i.e., handling requests from higher layer, making requests to lower layer.
- ...

In this course ...

- We will be studying some protocols at each layer.
- We will be (re)-inventing many protocols!

- We will ask questions such as:
  - What factors influence the design of a protocol?
  - What are the characteristics of a 'good' protocol?
  - What parameters are relevant for evaluating a protocol?

- And many more ...