CS 716: Introduction to communication networks

- 5th class; 5th Aug 2011

Instructor: Sridhar Iyer
IIT Bombay

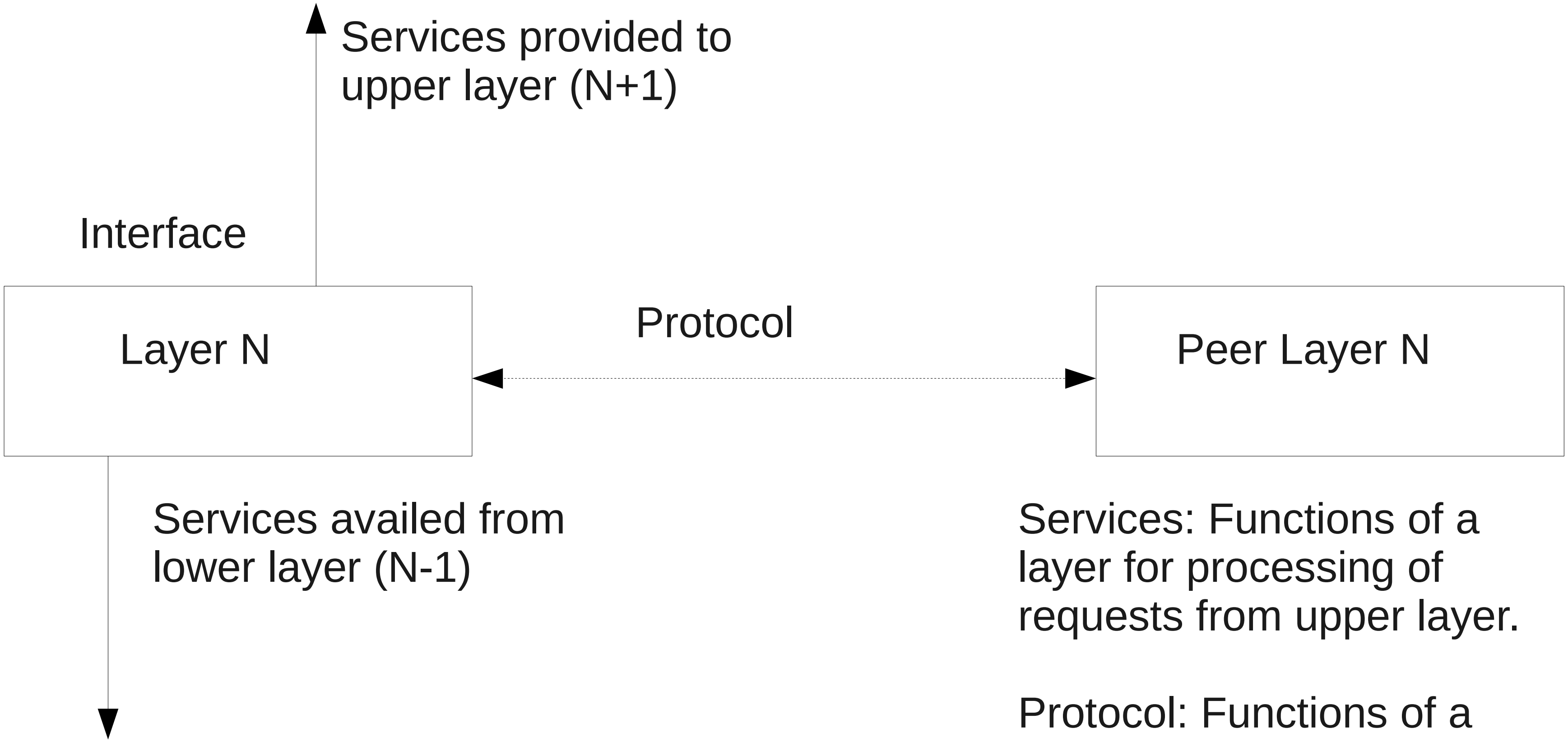
Clicker Question (from last class)

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!

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.

Activity: Think-Group-Share

- Questions:
  - What all needs to be defined in an interface?
  - Why is it important to specify an interface in detail?

- Think - Think individually (2-3 min).
- Group - Discuss within your group (5-6 min).
- Share - Discuss with entire class.
  - Hint: Recall examples of interfaces that you may have come across in other areas and think about it.

Some points from last year

- There is information flow between layer (N) and (N-1).
- Services provided by (N-1) to (N).
- Services and types decides the interface.
- It fixes data format.
- Mutual agreement between two layers (similar to protocol).
- Discrepancy of data can be resolved.
- It is not only data information but also control information.
- ...

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

Protocols

Recall previous discussion:

- What is the need for having protocols?
  - What will happen if we do not have them?
- How does the concept of protocols relate to the concept of layering?
  - What is the difference between the processing functions of a layer and the protocols for that layer?
- How does the concept of protocols relate to the concept of interfaces?
  - What are the similarities? What are the differences?

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 ...

Assignment 2 (To do in-class)

Design a web-based email Application layer protocol, that uses the service of your TCP-like protocol (from Assignment 1). Consider the scenario of you sending an email from your Gmail account to your friend's Yahoo account, and your friend reading the email.

1. Draw a diagram of the various computers involved in this communication and the relevant layers on each computer. You may assume TCP to be the lowest layer, so you need not show routers and lower layers/ devices.
2. Explain the main steps in the working of your protocol using the above example. Give details of the actions at the Gmail server, when you hit the “Send” button.
3. Show the data and control information passed between the application layer and transport layer on each computer. You may show these on the diagram itself.

At the end of this topic

You should be able to do:

- State the need for layers, interfaces and protocols.
- Identify situations where using layering is 'beneficial'.
- Identify situations where not using layering is 'beneficial'.
- Identify the components in an interface definition.
- Create the interface definition for a specified 'service'.
- Distinguish between an interface and a protocol.

- Brain-storming in a group to identifying pros and cons of a solution.
- Debate with other groups regarding the pros and cons.

Reflection

- What did I learn in today's class?
- Each student to mention one point.

- Take-home questions:
  - Can one entity have multiple interfaces?
  - What will happen if a protocol violates layering?