Protocol Stack

This exercise aims to illustrate how communication work between layers in a stack e.g. OSI (Open Systems Interconnection) at a high-level as well as an aid to understand the benefits of this approach in general.

We presume that you have done quite a few assignments using the similar user interface. Given prior experience already acquired, we will start off straight into simulation according to scenario description.

Launching the Protocol Stack Simulator

Ensure that your default browser supports applets. For access to all animations, click on the rotating globe at this book's Campanion Web site at
http://williamstallings.com/DataComm . Then click on the link to the Protocol Stack.
This will load the Protocol Stack Simulator applet along with description of the main commands we will be using for our experiments.

Main Commands Description

The following commands in the Protocol Stack Simulator are only abstractions of what would be in actual implementations and are simplified to illustrate the approach.

Command	Description
Application: Send application message to transport layer	Application to send data, which will go via the transport layer to handle underlying transmission
Transport: Send transport message to network layer	Transport layer appends its header to the application data, resulting in a transport message, and sends to the network layer to handle underlying transmission
Network: Send network message to data link layer	Network layer appends its header to the transport message, resulting in a network message, and sends to the data link layer to handle underlying transmission
Data Link: Send data link message to physical layer	Data link layer appends its header to the network message, resulting in a data link message, and sends to the physical layer to handle underlying transmission
Physical: Send physical message to medium	Physical layer appends its header to the data link message, resulting in a physical message, and sends to the medium to handle underlying transmission
Medium: Send medium message to physical layer	Instruction as it is
Physical: Send physical message to data link layer	Physical layer interprets physical header information, resulting in a data link message, and sends to the data link layer
Data Link: Send data link message to network layer	Data link layer interprets data link header information, resulting in a network message, and sends to the network layer

Network: Send network message to transport layer	Network layer interprets network header information, resulting in a transport message, and sends to the transport layer
Transport: Send transport message to application layer	Transport layer interprets transport header information, resulting in a application data, and sends to the application as destination

Applet Interface

The applet graphical interface consist of the control panel (bottom left corner), and commands panel (bottom right), and the simulation view (top half).

The control panel consist of several buttons. **Run** will perform an automatic simulation of the protocol. **Stop** will halt the automatic simulation. **Undo** will revert back one step from the latest command. **Redo** will revert what undo does. **Clear** will clear off the current simulation, resulting in a clean simulation view. **Load**, **Save** and **Print** buttons are only available when the protocol simulator is launched in standalone mode, which are not available as we are using the applet mode.

The command panel displays the available commands at the current point of simulation. Clicking on one of the commands will progress the simulation, seen in the simulation view.

The protocol simulation shows how data is being packaged and transferred across the Application, Transport, Network, Data Link, Physical, Medium layers, where the leftmost layer is higher than the one that follows on the right.

Experiment One: Understanding the model of protocol stack

Start with a fresh simulation by clicking the **Clear** button, then do the following simulation with these commands.

1. Application: Send application message to transport layer
2. Transport: Send transport message to network layer
3. Network: Send network message to data link layer
4. Data Link: Send data link message to physical layer
5. Physical: Send physical message to medium

Observe the simulation so far and from what you already learnt about protocol stack, answer the following questions:

1. Why is there information appended as the message traverse downwards through the layers? Explain with an example for the Network layer
2. What does the above notion generally known as?
3. Give an example in reality what could be the other instances of Application, Transport, and Medium.

Complete the simulation with the following commands

1. Medium: Send medium message to physical layer
2. Physical: Send physical message to data link layer
3. Data Link: Send data link message to network layer
4. Network: Send network message to transport layer
5. Transport: Send transport message to application layer

Observe the completed simulation and answer the following question.

1. With respect to your answer to Questions 1 and 2 above, explain what does this notion help achieve?
2. Do you find Protocol Stack familiar to things or systems you come across often? Give an example that you think *may* relate.