Network Game
CS442

The purpose of this game is to help you understand how messages are transmitted in computer networks. The game loosely mirrors the way that TCP/IP works in a LAN setting (actually, more like the way UDP/IP works; UDP doesn't do all the bookkeeping that TCP does). However, keep in mind that it does skip a lot of details in a real networking implementation. Ideally, each team is composed of four people. Each person in the team assumes the role of some layer(s) in a communications network and works with the others to send messages through the network. Dr. Alan Dennis at the University of Georgia originally developed this game.

In this game, messages will be generated by the Application layer and passed to the Transport/Internet layer. The Transport/Internet layer will address and route the message, and pass the message to the Data Link layer. The Data Link layer will format the message, perform error control, and pass the message to the Physical layer. The physical layer will transmit the message and potentially introduce error.

Each layer will be given a set of instructions to follow. At some point, mistakes may be made – if you discover a mistake in how the game is played, hand the message back to the person who gave it to you and explain the error to them.

Game Forms:

Data Link					Internet		Application		
Source	Dest	Control	#	FCS	Next IP	Final Dest IP	From	To	Message

Physical Layer Transmission Form

Application Layer Instructions

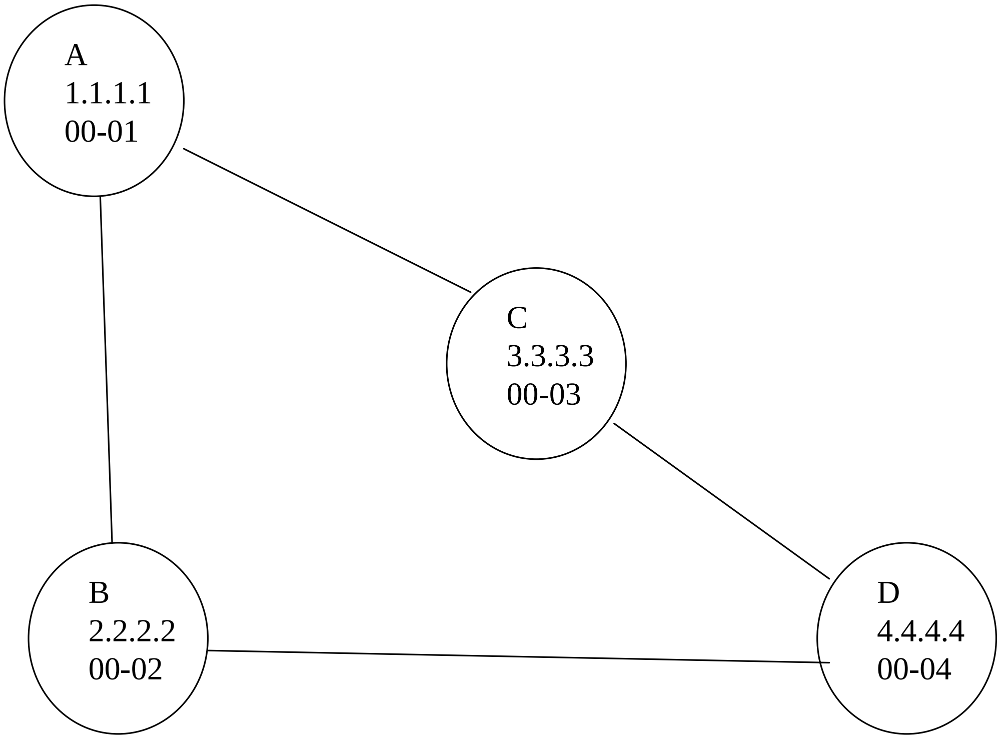
The job of the application layer is to send messages to other computers, and to respond to messages received from other computers.

Sending Messages: To send a message you need to:

1. Find a blank message form.
2. Write the *alphabetic application layer address* of your computer in the **From** box.
3. Use the network map to select a destination computer to send the message to. Write the application layer address of this computer in the **To** box.
4. Write a short message in the **Message** box. Initially, a simple question such as "Favorite class?" or "Your major?" or "A/S/L" (if you want to be annoying!)
5. If you send a lot of messages, you might want to keep track on a separate piece of paper what messages you sent. This is to help keep things straight when you start to get replies back from multiple machines.
6. Pass the message to the Transport/Internet layer.

Responding to messages: If all goes well you should eventually be passed a message back from the Transport/Internet layer. Follow the same steps as above if it is a question, but give your answer as the message.

Sample Network Map with Application, Internet, and Data Link addresses:



Transport/Internet Layer

The Transport/Internet layer is responsible for accepting messages from the application layer, addressing them, routing them, and passing them to the data link layer. Additionally, the Transport/Internet layer will accept messages from the data link layer. If addressed to you then it will pass the message up to the application layer. If not, the message will be re-routed through the network.

In this game, the Transport/Internet layer plays the role of a network router as well as software on a sending or receiving computer. Normally the Transport layer would also perform functions such as session management, flow control, breaking data up into packets, and error checking. These tasks have been omitted from this game for simplicity. Since these functions are gone, the game more closely models the simpler UDP (User Datagram protocol) than TCP protocol.

Accepting Messages from the Application Layer: When given a message to transmit follow these steps:

1. Address the message by translating the destination application layer address into the destination IP address. Write the IP address in the **Final Dest IP** box.
2. Route the message by finding the next computer to which the message should be sent in the network map. Write its IP address in the **Next IP** box. If your computer is directly connected to the final destination, the Next IP should be the same as the final. If your computer is not directly connected to the destination, you must select the best route that the message should follow.
3. Pass the message to the Data Link layer.

Accepting Messages from the Data Link Layer. This will happen when you have received a message from someone else:

1. If the message is addressed to you, i.e. the **Final Dest** box is your network address, then pass the message to the application layer.
2. If the message is not addressed to you, route the message by finding the next computer to which the message should be sent in the network map. Do not send the message back to the computer it just came from! Write its IP address in the **Next IP** box. Pass the message to the Data Link layer.

Data Link Layer

The data link layer is responsible for accepting messages from the Transport/Internet layer, formatting them, adding error information, and passing them down to the Physical layer.

Additionally, the data link layer will take messages from the Physical layer, and send an ACK or a NAK depending upon whether or not an error was detected. Similarly, this layer will retransmit a message if a NAK is received.

The duties of this layer are the most complex and time-consuming. Please familiarize yourself with the rules to avoid becoming a bottleneck to everyone else!

It may be helpful to keep three piles:

- Messages from the Transport/Internet layer
- Messages from the Physical Layer
- Messages Transmitted

Accepting Messages from the Transport/Internet layer:

1. Format the message. Write your Data Link address in the **Source** box. Write the Data Link address of the **Next IP** box in the **Dest** box. Write an asterisk (*) in the **Control** box. The * means that this is a message you are sending.
2. Number the message. Write a two digit number in the **#** box. Use 01 for the first message you send, 02 for the second, etc.
3. Add error control information. Count the number of letters (include punctuation) in the **Message** box and write it in the FCS (frame check sequence) box. Obviously, this is a very simple error control protocol.
4. Pass the message to the Physical layer. In a minute or so you should receive it back. When it is returned, place it in the “Messages Transmitted” pile.

Accepting Messages from the Physical Layer: If you receive more messages than you can handle, buffer them in the “Messages from the Physical Layer” pile.

1. Copy the contents of the Transmission Form onto a new Message Form. Destroy the transmission form.
2. If the **Control** box has an *, this is a data message:
    a. Perform error checking by counting up the letters in the **Message** box. If this number is the same as the number in the **FCS** box, there was no error. Otherwise, there was an error.
    b. If there was no error, send an ACK to the sender to tell them all is well.
        i. With a new Message Form, write your Data Link address in the **Source** box. Write the Data Link address contained in the **Source** box of the incoming message in the Dest box. Write the word ACK in the **Control** box. Copy the contents of the **#** box from the incoming message to the **#** box of the outgoing message.
        ii. Pass the ACK to the Physical Layer.
        iii. Pass the incoming message to the Transport/Internet layer.
    c. If there was an error, send a NAK to the sender to ask him to resend.
        i. With a new Message Form, write your Data Link address in the **Source** box. Write the Data Link address contained in the **Source** box of the incoming message in the Dest box. Write the word NAK in the **Control** box. Copy the contents of the **#** box from the incoming message to the **#** box of the outgoing message.
        ii. Pass the NAK to the Physical Layer.
        iii. Destroy the incoming message containing the error.
3. If the Control box contains an ACK then this is an acknowledgement. Do the following:
    a. Find the original message you sent in the “Messages Transmitted” pile that has the same message number (#) as the ACK.
    b. Destroy the original message and the ACK.
4. If the Control box contains a NAK, this is a negative acknowledgement. Do the following:
    a. Find the original message you sent in the “Messages Transmitted” pile that has the same message number as the NAK.
    b. Give the original message to the physical layer to transmit. In a few minutes you should have it returned to you by the physical layer. Save it in the “Messages Transmitted” pile.
    c. Destroy the NAK message.

Physical Layer

The physical layer accepts messages from the data link layer and passes them to the physical layer of the destination, possibly introducing a transmission error. This layer also accepts messages from the physical layer of other computers and passes them to the data link layer.

Accepting Messages from the Data Link Layer. Every few minutes you should receive a message from the data link layer to transmit.

1. Determine if there will be an error in transmission. If the **Control** box contains an asterisk (*), then toss two coins. If they are both heads, you will introduce an error. If the **Control** box is an ACK or a NAK, do not introduce an error (things are complicated enough without ACKs and NAKs getting destroyed!)
2. Copy the entire contents of the message from onto a transmission form. If you are to introduce an error, omit a random letter from the **Message** box.
3. Pass the transmission form to the physical layer of the computer whose address is listed in the **Dest** box.
4. If the **Control** box contains an asterisk (*), pass the message form back to the Data Link layer and make sure the he or she understands that this was a transmitted message being returned, not a new message.
5. If the **Control** box contains ACK or NAK destroy the message form.

Accepting Messages from the Physical Layer:

1. When another computer's physical layer hands you a message, simply pass it along to your data link layer. (If your data link layer is getting overloaded, you can help him or her by doing some of their busy work by copying the transmission form onto a message form.)

Playing the Game

While the game is going, it will help to play different roles. Every 15 minutes try to rotate and play a different layer. If you have a group of 3 students, have one person assume the duties of Application and Transport/Internet layer.

Questions:

1. Why are standards important?
2. How could you improve network performance by changing topology?
3. How could you improve network performance by changing the protocol?
4. What layer is busiest and how might you help improve performance?
5. Instead of both a data link and an IP address, could we just have one address?