IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Hershey PA 17033-1117, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4249

The Internet Game

Norman Pendegraft
Associate Professor of Management Information Systems, College of Business and Economics, University of Idaho, norman@uidaho.edu

ABSTRACT

This paper describes a simulation game used to assist students to understand the operation of TCP/IP. The game is used in a telecommunications management course taken primarily by MIS majors at the senior level. In the game students play various layers of TCP/IP on several machines and collaborate to transmit a message from one application to another. The game consumes about one hour, and anecdotal evidence suggests that it is help to the students.

ABSTRACT

This paper describes a simulation game used to assist students to understand the operation of TCP/IP. The game is used in a telecommunications management course taken primarily by MIS majors at the senior level. In the game students play various layers of TCP/IP on several machines and collaborate to transmit a message from one application to another. The game consumes about one hour, and anecdotal evidence suggests that it is help to the students.

INTRODUCTION

Many IS programs offer a telecommunications management course taken primarily by IS majors. The course attempts to balance the technical and managerial issues involved in TC management and because to the large subject matter, time is limited for each topic. The technical topics include TCP/IP and the more general ISO layered model. It has been my experience that many students, exposed only to reading and lectures, do not understand the interaction between the layers, nor why each is necessary, nor how the whole package works. One possible solution is to have the students build a computer simulation of TCP/IP, however, such an assignment is time consuming and technically demanding. For all its value, it seems a misuse of the limited time available in this course. Another alternative is to create a simulated system and let students experiment with it. Campbell (1996) simulated a computer and let students write assembly language programs to run on the computer.

Simulation gaming is predicated on the notion that students learn better from experience than from lectures and reading. They are actively involved rather than passive and the connections between pieces are often easier to see when experienced. (Greenblat 1988). After many years of experience using various games in class, the author has come to believe strongly in their value as pedagogical vehicles. (See for example Butterfield and Pendegraft 1996, Pendegraft and Watson 1996). Informal student response to this game has been most enthusiastic. Here, a simulation game was developed to let the students experience the operation of TCP/IP without mandating that they construct the simulation themselves. The purpose of the simulation was to help the students understand the function of each layer and in particular to understand how the layers interact with each other.

The rest of the paper is organized as follows. First the overall structure of TCP/IP as represented in the simulation is briefly discussed. Then, the design and operation of the game is described. Finally there is a brief discussion of the results.

TCP/IP

There are many excellent discussions of TCP/IP (for example, Hunt 1998). This discussion is not intended to give a detailed nor technical description of TCP/IP, but rather to clarify some of the simplifications imbedded in the simulation. TCP/IP is a layered protocol. Each layer performs a set of functions which together result in a reliable communications connection. Each layer adds to the packet its own header containing information needed by the same layer at the receiving end. In order to keep the simulation manageable, many of the specific functions are excluded from the simulation or are included only as notional entities. For example, error correction is handled notionally: CRC calculations are greatly simplified. Also, while each layer may take the packet from the higher layer and break it into smaller pieces, this is only done by TCP in the simulation.

APPLICATION LAYER

The application layer includes process like email, web services and the like. The simulation uses email as an example. A port number identifies each application running on top of TCP/IP. Several so-called well-known port numbers are used in the example.

TCP

The Transport control protocol is a service which creates and maintains connections between machines. It establishes a connection by sending a SYN message. It then breaks traffic into pieces called segments, calculates and appends a CRC (cyclical redundancy checksum) to each, and sends them. Each is addressed to the correct port (application) on the receiving machine. TCP on the receiving machine checks the CRC of each segment and acknowledges those correctly received. Those unacknowledged in time are resent. Error detection and retransmission are handled notionally in the simulation. That is, the instructor stipulates during the exercise whether a packet is transmitted and received correctly, and the system (i.e. the students) responds accordingly.

IP

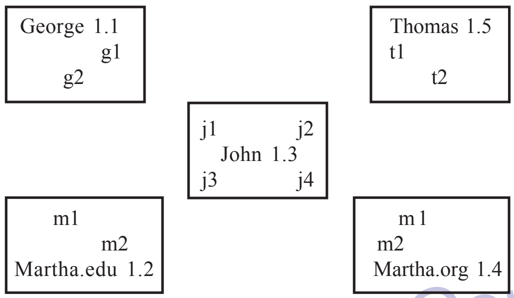
Packets at the internet protocol level are called datagrams. While segment may be broken into multiple datagrams, experience has shown this to introduce too much difficulty, so it is not included in the simulation. IP is primarily responsible for routing datagrams, that is, selecting the route they will take to the recipient. Two key ideas are the use of the routing table and routing protocol. The protocol is the rule which updates the route table as data about the network is received. In particular, it determines the first hop to any given IP address. The routing table maintains data about the network and in particular about the first hop to each destination.

NETWORK ACCESS LAYER

The NAL is responsible for converting an IP address into a hardware or MAC address. NAL also contains error correction data to ensure that frames are correctly received. Again, error correction is handled notionally.

This paper appears in *Issues and Trends of Information Technology Management in Contemporary Organizations,* the proceedings of the Information Resources Management Association International Conference.

Figure 1: Network layout



Physical

The physical layer contains the network interface cards (NICs) and the cables / hubs etc connecting machines.

DNS

The Domain Name service (DNS) is an application layer service which converts domain names like www. whitehouse.gov or ebay.com into their IP addresses. DNS services are outside the scope of this exercise and are notional in the simulation. DNS information is assumed available to IP in the simulation.

Figure 2: Routing tables–first hop

	Source				
Destination	1.1	1.2	1.3	1.4	1.5
1.1.0.0	0	1.1.0.2 /1	1.1.0.3 /1	1.3.0.4 /2	1.3.0.5 /1
1.2.0.0	1.2.0.1 /2	0	1.2.0.3 /3	1.3.0.4 /2	1.3.0.5 /1
1.3.0.0	1.2.0.1 /2	1.3.0.2 /2	0	1.3.0.4 /2	1.3.0.5 /1
1.4.0.0	1.3.0.1 /2	1.3.0.2 /2	1.4.0.3 /4	0	1.4.0.5 /2
1.5.0.0	1.3.0.1 /2	1.3.0.2 /2	1.5.0.3 /2	1.5.0.4 /1	0

* 0 indicates that the packet is addressed to the source machine
/i identifies NIC #i on the source machine

Figure 3: Application port map

Application	Port
FTP	21
SMTP	25
DHCP	67
HTTP	80
Oracle	1525

Figure 4: Domain name map

Domain Name	IP
George.gov	1.1.0.0
martha.edu	1.2.0.0
john.gov	1.3.0.0
martha.org	1.4.0.0
thomas.mil	1.5.0.0

Figure 5: Packet header and data forms

Dest MAC	Send MAC	MAC CRC	Dest IP	Send IP	Time to live	Dest port	Send Port	TCP CRC	Sequence

1	2	3	4	5	6	7	8	9	10

SIMULATION DESIGN

The design of the game is most easily described by illustrating the handout materials and then walking through its operation. The first handout describes the network. Figure 1 contains a diagram of a hypothetical network consisting of five machines named George, John, Thomas, Martha.edu, and Martha.org. The first two octets of the internet address of each is indicated in the drawing as well. Duplicate names (Martha) were chosen for two machines to illustrate the importance of root domain names. Each of the machines has two or more network ports numbered as indicated.

The routing table (Figure 2) shows for each source the first hop to each possible destination. The column for each source machine is assumed to be available to that machine. Figures 3 and 4 include a port map to determine the port used by each application and the name map used by the DNS resolver.

The second handout contains message forms in two parts. Figure 5 shows the message form and the header form. The use of all of these is described below.

OPERATION

The class is asked to separate into teams of 2-6 for each computer. They are "connected" by 6-10 foot long pieces of wire held at each end by the player playing the "NIC". (One player can play two NICs by using one hand for each. At two of the computers, one student is assigned to be the application (Email) and one is asked to send a message to the other.

The operation is summarized in Figure 6. The sending application writes the message on the message form which has only 10 characters per line. TCP tears the message into 10 character chunks, adds a sequence number and appends to each a header. Each of these combinations becomes a packet. TCP adds to each the port number of the sending and receiving applications and sends it to IP.

IP gets the IP address from DNS and appends it along with its own IP address to each packet. IP looks in the routing table and sends the packet to the NAL with the IP address of the sending port. It also

Figure 6: Operation summary

Layer	Send (read down)	Receive (read up)	Simulation (send)
Appl	Create a message (stream)	Read message	
	Append addressee DNS name	Confirm DNS name	
	Append recipient Appl Name		
	Append your DNS name		
	Pass to TCP		
TCP	Establish Connection (SYN) / End Connection (FIN)	Pass to Appl	
	Break message into segments add sequence #	Evaluate Dest Port#	tear paper into max=10 char segments
	Request IP address from DNS	IF correct Send ACK	
	Append destination Port#	Assemble segments	add dest P# to header
	Append source Port#		add source P# to header
	Append CRC to each		notional CRC
	Pass to IP	Calculate error code	
IP	Break segment into datagrams		notional
	Add destination IP address to each	Pass to TCP	Add source IP
	Add source IP address to each	Assemble Datagrams	Add Dest IP
	Lookup First Hop IP	IF for me keep ELSE store and forward	First hop IP
	Determine Output Connection (NIC)	Read IP address	
	Add Time to Live		
	Pass to DLL for NIC		
NAL	Break datagram into frames		
	Add dest. MAC address to each frame		source and dest MAC
	Add your MAC address to each frame	Calculate error code IF correct send to IP ELSE request retrans	
	Send to receiving MAC	IF for me store	
Physical			envelope & wire

NAL looks at the IP address for the sending and receiving ports and looks up their MAC addresses. These are appended to the packet which is then routed to the correct NIC. It also calculates a CRC which is added to the header. The packet is passed to the physical layer.

The physical layer is represented by a 6-10 foot long piece of wire which has threaded through a hole in an envelope. One end is held by each of the two linked Niches. The sending end lifts the wire and the packet runs down the wire to the next computer.

At the receiving end the process is reversed. There are some differences. For example, error checking is performed instead of CRC calculation. IP checks for destination and forwards the packet if necessary. TCP acknowledges each packet. (In my classes we do this verbally to simplify matters.)

DISCUSSION

No statistical data is available to demonstrate the value of this simulation. The informal anecdotal evidence is, in the author's opinion, strong enough to continue using and improving the simulation. Students have offered a number of unsolicited comments on the simulation, all positive so far. Finally, the major benefit seems to me to accrue to the instructor. By designing and running the simulation, the instructor develops a much deeper understanding the underlying technology; at least that has been my experience.

REFERENCES

Butterfield, Jeff, and Norman Pendegraft. "Gaming Techniques to Improve the Team-Formation Process," *Team Performance Management*, Vol. 2, No. 4, 1996, pp. 11-20.

Campbell, Robert A. "Introducing Computer Concepts by Simulating a Simple Computer," *SIGCSE Bulletin*, Vol. 28, No. 3, September 1996.

Corbeil, Pierre, Dany Laveault, and Michel Saint-Germain. Games and Simulation Activities, *Tools for International Developmental Education*, Canadian International Developmental Agency, Quebec, 1989.

Greenblat, Cathy Stein. *Designing Games and Simulations*, Sage, Newbury Park CA, 1988.

Hunt, Craig. *TCP/IP Network Administration*, O'Reilly, Sebastopol, 1998.

Pendegraft, Norman, and William Watson. "Gaming Techniques to Improve the Team-Formation Process," *Team Performance Management*, Vol. 2, No. 4, 1996, pp. 11-20.