Remarks: Keep the answers compact, yet precise and to-the-point. Long-winded answers that do not address the key points are of limited value. Binary answers that give little indication of understanding are no good either. Time is not meant to be plentiful. Make sure not to get bogged down on a single problem.

PROBLEM 1 (40 pts)

(a) What is the role of Slow Start in TCP congestion control? Why is it a misnomer? Is there a compelling need to perform Slow Start following a timeout at which the Congestion Window is halved? How is this issue addressed in TCP?

(b) In the U.S., broadband (i.e., cable/DSL modem speed or higher) deployment has become the dominant mode of Internet access. Unlike dial-up access, broadband users may stay connected indefinitely so that the savings brought about by dynamically assigned IP addresses has diminished significantly. What technique has come to the rescue, alleviating some of the IP address space crunch? Using two households as examples where each has a single PC that is used to browse the World Wide Web (i.e., each PC makes a TCP connection to a port 80 web server speaking HTTP), describe how an ISP can use a single IP address to serve both households. Can such a household use its PC to run a web server?

(c) Suppose three coffee shops are adjacent to its each other on the same side of a street. Call them Florence, Seattle, and Vienna. Each offers free wireless Internet access through a single 802.11g AP. Unwittingly, the coffee shop owners have all set their AP channels to 6 so that simultaneous data transmissions by customer laptops and PDAs collide. After a couple of months of operation (the three coffee shops deployed WLANs at about the same time), Seattle—the coffee shop that sits between Florence and Vienna—lost many of its customers to its two competitors because it gained a reputation as providing bad throughput. Because Seattle's owner graduated from IU, whereas the owners of Florence and Vienna are Purdue graduates, poor performance was attributed to the perceived technical shortcomings of Seattle's owner (in this instance undeserved).

What is the likely real culprit behind Seattle's lackluster throughput? What could the three owners do to alleviate Seattle's throughput problem? Will this necessarily reduce Florence and Vienna's throughput? (If so, economically speaking, it may not be in their best interest.)

(d) What is the technical meaning of congestion? Why does packet drop not imply congestion? What mechanism in TCP addresses this issue? How is the congestion control problem faced by two protocols, each transferring data over a shared bottleneck link, related to the Prisoner's dilemma problem? What is the "best" course of action to take upon encountering signs of congestion: back off (cooperative behavior) or not back off (noncooperative behavior)? Explain what your criterion of "best" is.

PROBLEM 2 (30 pts)

(a) There are at least three layers in the protocol stack where "routing" of packets is dealt with: at the LAN layer (e.g., forwarding of frames in an Ethernet LAN comprised of several switches), intra-domain layer (e.g., routing of packets within Purdue), and inter-domain layer (e.g., routing of a packet from Purdue to one of its network providers so that it reaches a destination on the west coast). What are the main distinguishing characteristics of how routing is performed by the three layers?

(b) Two recent developments in the wireless arena are high-speed very short distance communication standardized by IEEE 802.15 (e.g., high-speed real-time wireless streaming from a camcorder to your laptop, wireless streaming from laptop to high definition TV or a projector) and high-speed medium distance communication standardized by IEEE 802.16 (e.g., data connectivity at distances covered by today's cell towers). Given the arsenal of bandwidth multiplexing techniques—contention-free MAC protocols such as TDM, FDM, TDMA, CDMA, token ring, and contention-based MAC protocols such as CSMA/CD and CSMA/CA—what would you argue are the most appropriate ways of mediating very short distance and medium distance wireless data communication? In the former, how is multiplexing related to FireWire (IEEE 1394) and USB 2.0?

(c) An employee in a company located in Copenhagen, Denmark, sends an e-mail to an employee at another company in Copenhagen whose building is just across the street. The two companies access the Internet through different

ISPs. It may happen (and it has happened) that such an e-mail (TCP packets transported by IP) travels across the Atlantic to the U.S., then makes a U-turn to cross the waters again, before reaching the building just across street. Describe a possible scenario based on the idiosyncrasies of inter-domain routing that may have led to such a long-winded journey.

PROBLEM 3 (30 pts)

(a) HTTP is a simple text-based client/server protocol that runs on top of TCP. Although TCP is stateful because it maintains sender/receiver related information until a connection is terminated, HTTP itself is not (except for cookie IDs, an optional feature that clients can disable). The same holds for the client/server application programmed and tested in the homework assignments. The most popular HTTP request type is GET which has the format

GET URL HTTP/1.0

The request line is followed by one or more header lines. The essential information is conveyed by the request line. Sketch how you would modify the concurrent client/server application (i.e., the server side) so that it acts as a port 80 web server. You may assume that the URL specifies a file or directory under `/home/apache/web` on a generic file system. Note that a full-fledged web server must contend with non-networking issues such as running various scripts (e.g., JavaScript, PHP) whose output is returned to the client (not unlike the remote command client/server application in the homeworks that returns the output generated by a legacy command).

How does the server code change when the client uses HTTP/1.1? Would it make sense to turn the concurrent server into an iterative server? Would it make sense to use UDP in place of TCP for HTTP? To create a "secure" version of HTTP that provides encryption/decryption service using a private or public key cryptosystem, do routers need to get involved? If not, should they get involved? Where, in your server code, would encryption and decryption be carried out?

(b) What is the fundamental difference between pseudo real-time multimedia streaming and real-time multimedia streaming? What does congestion control method A do, and what problem does it encounter? What does method B bring to the table? Is it any better than method A? In what way does method C try to be more clever than A and B? Does it succeed? Suppose method D, which outperforms methods A, B, and C, is to be ported to TCP. What are some of the challenges associated with doing so? How would you overcome these challenges?

BONUS PROBLEM (10 pts)

A solution to the selfishness problem in congestion control where some sessions may gain a significant fraction of shared bandwidth at the expense of others is to institute usage pricing so that those who use more resources are charged accordingly. Usage pricing for data services is wide-spread in some parts of the world including countries in Asia and Europe. In the U.S., flat pricing is the norm (i.e., pay a fixed amount per month independent of usage) although cellular data service may follow a usage based service structure. Of course, several daily household commodities such as electricity and gas are charged usage based, so the usage pricing concept is familiar to all. From the perspective of addressing the selfishness problem in present/future Internet protocols, discuss your views on the advantages and disadvantages of employing usage pricing for everyday data services.