

# CS 5565 Sample Final Exam Spring 2006

---

## 1 TCP (20 pts)

a) (12 pts) Modeling TCP's Congestion Avoidance.

Misra et al have modeled TCP's congestion avoidance phase using a so-called Poisson Counter driven Stochastic Differential Equation. For a given Poisson process  $N$ , a Poisson counter  $dN$  is defined as

$$dN = \begin{cases} 1 & \text{at Poisson arrival} \\ 0 & \text{elsewhere} \end{cases}$$

Intuitively,  $dN$  is 1 if an event occurs, and 0 otherwise. Based on this definition, the change in window size  $dW$  for a TCP Reno implementation operating in congestion avoidance mode can be modeled as:

$$dW = \alpha dt + \beta dN_{TD} + \chi dN_{TO}$$

Where  $dN_{TD}$  is a Poisson counter for a process that models the occurrence of triple-duplicate acknowledgements, and  $dN_{TO}$  is a Poisson counter for a process that models the occurrence of timeout events.  $dt$  is defined as in an ordinary differential equation.

What are  $\alpha$ ,  $\beta$ , and  $\chi$ ? (Ignore slow start!)

$\alpha =$

$\beta =$

$\chi =$

*TCP Reno's congestion avoidance can be modeled as a Poisson Counter driven Stochastic Differential Equation as follows:*

$$dW = \frac{1}{RTT} dt + \left(-\frac{W}{2}\right) dN_{TD} + (1-W) dN_{TO}$$

*The values of  $\alpha$ ,  $\beta$ , and  $\chi$  follow from the basic properties of TCP's congestion control algorithm:*

$\alpha = \frac{1}{RTT}$  - the congestion window is increased for every received

*acknowledgement; acknowledgement arrive on average once every RTT.*

$\beta = -\frac{W}{2}$  - on a triple-duplicate ack, TCP Reno halves the congestion window

$\chi = 1 - W$  - on a timeout loss event, the window is set to 1.

- b) (2 pts) Explain, in your own words, the difference between equation-based and loss-based congestion control!

*Loss-based congestion control requires that packets actually be lost before the sender realizes that congestion is about to occur and throttles back. In equation-based control, the sender adjusts its send rate based on an equation that takes as input delay and loss events. In most cases, the sender can adjust its rate before packet loss actually occurs.*

The next question refers to: [FAST TCP: From Theory to Experiments](#). C. Jin, D. X. Wei, S. H. Low, G. Buhrmaster, J. Bunn, D. H. Choe, R. L. A. Cottrell, J. C. Doyle, W. Feng, O. Martin, H. Newman, F. Paganini, S. Ravot, S. Singh. IEEE Network, 19(1):4-11, January/February 2005.

- c) (6 pts) The FAST paper motivates the use of a novel primal congestion control algorithm by pointing out that active queue managements (AQM) can only achieve stability at the expense of a large queue.
- i. (2 pts) Explain what is meant by stability! (You may draw a chart to illustrate it.)

*Stability here means that the network reaches a new equilibrium point after changes in topology or flow pattern. In other words, the congestion control process should converge.*

- ii. (2 pt) Why is a large queue undesirable?

*A large queue is generally undesirable for the natural reason, because it means a large queuing delay.*

- iii. (2 pts) What is the primary advantage of achieving source-based congestion control?

*The primary advantage of achieving source-based congestion control is that it works with the existing link infrastructure, i.e., it does not require changes at the routers (see pg 6, left column.) Unlike a TCP/AQM approach, which requires that all routers (i.e., links) run the same global AQM algorithm to stabilize the network.*

## 2 Error Correction (15 pts)

Consider a (6,3) Hamming Code.

- a) (2 pts) How many bit errors can this code detect? Justify your answer!

*A (6,3) Hamming Code uses 3 parity bits for every 3 data bits. The Hamming condition is fulfilled since  $(3 + 3 + 1) < 2^3$  – hence the code can correct 1 bit errors.*

- b) (2 pt) Alternatively, how many bit errors can this code correct?

*Alternatively, it can detect, but not correct, 2 bit errors.*

- c) (8 pts) Here is a table for this code. Unfortunately, some values in the table are missing. Fill in the missing values.

Data Bits	Code Word
000	000__
001	000111
010	__001
011	011110
100	101010
101	101__
110	1__11
111	1__00

*The Hamming distance between any two code words must be 3 or higher.*

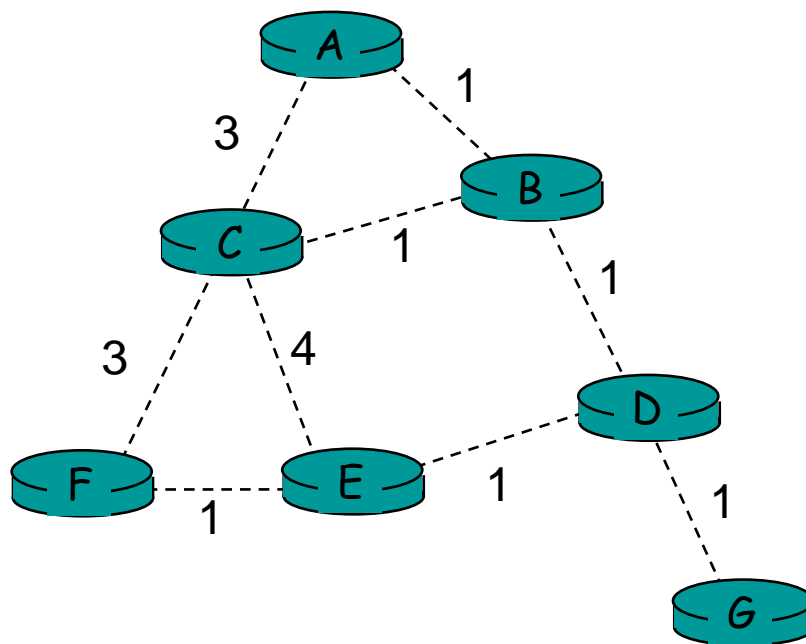
<i>Data Bits</i>	<i>Code Word</i>
<i>000</i>	<i>000000</i>
<i>001</i>	<i>000111</i>
<i>010</i>	<i>011001</i>
<i>011</i>	<i>011110</i>
<i>100</i>	<i>101010</i>
<i>101</i>	<i>101101</i>
<i>110</i>	<i>110011</i>
<i>111</i>	<i>110100</i>

- a) (3 pt) Under what circumstances would you use the code table from part c) compared to an alternative of using a single parity bit?

*Since this code imposes an overhead of 100%, compared to the 33% imposed by a single parity bit, you would probably only use it for links with a high error rate where you would otherwise have to retransmit a large amount.*

### 3 Routing Algorithms (19 pts)

a) (5 pts) Consider the following topology:



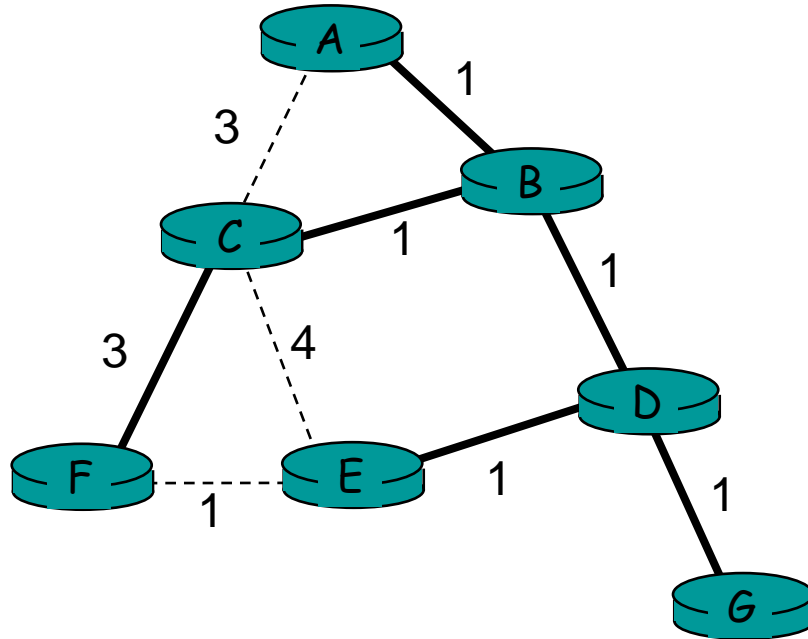
Consider a synchronous version of the distance vector routing algorithm. Assume that after iteration 1, a node has learned the costs to its neighbors, after iteration 2 the costs to its neighbors' neighbors, and so on. How many iterations does this synchronous version of the distance vector algorithm take to converge on this topology? Justify your answer and show how you are counting.

*F needs to learn the FEDBA route to A with a cost of 4. This route is propagated via BDE.*

- o After iteration 1, B knows its cost to A is 1.*
- o After iteration 2, D knows about the BA route to A with a cost of 2.*
- o After iteration 3, E knows about the DBA route to A with a cost of 3.*
- o After iteration 4, F knows about the EDBA route to A with a cost of 4.*

*All other nodes will have learned their shortest route earlier, hence the distance vector algorithm converged after 4 iterations.*

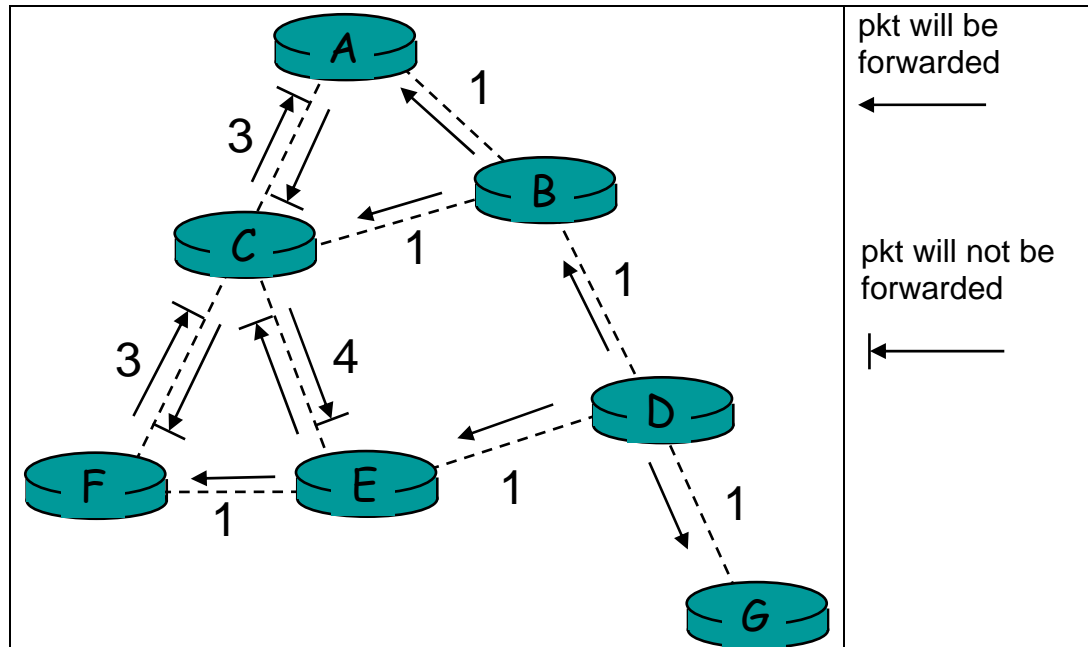
- b) (5 pts) Once the unicast cost is known, suppose a center-based algorithm is used to build a spanning tree with C as the center. Mark the resulting spanning tree.



- c) (4 pts) Is the spanning tree the minimum spanning tree? Say why or why not.

*This spanning tree is not the minimum spanning tree – the minimum spanning tree would include FE rather than FC, leading to an overall cost that’s less by 2.*

- d) (5 pts) Consider a message that is broadcast by node D using reverse path forwarding. Show which packets are forwarded and which are not.



#### 4 Implementing Routing (16 pts)

- a) (4 pts) In the paper by Rexford et al on network-wide decision making, the case of shared link risk groups is discussed as an example of a protection problem that could be better handled if a centralized decision plane were used. The paper states that absent such a centralized plane, a current management plane solution would need to find a “good setting of the IGP link weights” to increase protection. Give an example scenario and explain what such a “good setting” might be!

The paper referred to is: [Network-Wide Decision Making: Toward a Wafer-Thin Control Plane](#), J. Rexford, A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, G. Xie, J. Zhan, H. Zhang. HotNets-III. San Diego, CA, November 2004.

*Shared risk link groups (or SRLG) are groups of links that depend on some shared element, such as a shared conduit of fiber. If this conduit is damaged, all links running through it might fail.*

*Multipath routing is used to provide redundancy. OSPF weights are set such that multiple equal lowest-cost paths result from Dijkstra’s computation. If all lowest-cost paths between nodes traverse links that are part of a single SRLG, the intended effect is not achieved. Therefore, weights must be set such that there are at least two lowest-cost paths that don’t share a risk element.*

- b) (4 pts) The paper states:  
 “The dissemination plane’s primary objective is the timely, reliable dissemination of information to and from the network elements.”

- i. (2 pts) Give one example of information that would be disseminated to the network elements.

*The main information that's disseminated to the network elements are the forwarding tables, which are centrally computed.*

- ii. (2 pts) Give one example of information that would be disseminated from the network elements.

*Network elements would still have to learn about their neighbors, and the status of their neighbors, and would have to communicate that information to the central decision plane.*

- c) (4 pts) The past years have seen a nearly exponential growth in the number of hosts connected to the Internet. Yet, the size of the routing tables kept in Tier-1 ISP routers has only risen linearly. Explain why that is the case!

*Tier-1 ISP routing tables grow only linearly because of route aggregation.*

- d) (4 pts) In project 2B, you had to handle the case of a link failing, but we did not ask you to handle the case of a node failing. What, if anything, would you have to change in your implementation to handle node failures? Address both your distance-vector and link-state implementation.

*A node failure is indistinguishable from the simultaneous failure of all of its attached links – so both your distance-vector and your link-state implementation should already handle it, at least as far as computing alternate routes.*

## 5 IP/NAT/Ethernet (20 pts)

- a) (4 pts) Your roommate powercycles your shared NAT gateway that connects you to your ISP. What, if anything, happens to the ssh connection you're typing on and why?

*Assuming that the ssh connection was from inside the NAT'ed network to the outside, you will lose your connection and will have to log back in. The reason is that NAT must store the translation information for each TCP connection. This information is stored in volatile memory, so it is lost. The NAT gateway also cannot recreate it, since it is established at TCP connection time.*

- b) (4 pts) Which subnet mask does the notation /21 correspond to?

*The notation /21 corresponds to subnet mask 255.255.248.0.*

- c) (4 pts) Give one example of management functionality that an (expensive) Ethernet switch designed for use by large institutions might provide!

*Expensive Ethernet switches provide remote filtering and VLAN management functionality, allowing an administrator to set up and maintain virtual LANs and control the flow and forwarding of frames in them.*

- d) (4 pts) In lecture, we discussed HOL-blocking in the context of router architectures. Could HOL-blocking also occur in an Ethernet switch? Justify your answer!

*HOL-blocking could absolutely occur in an Ethernet switch – switches perform tasks similar to routers in that they have a number of input and output ports and forward frames between them. In practice, one tries to avoid HOL-blocking by using output-queuing rather than input queuing.*

- e) (4 pts) Why do Ethernet switches keep their switching table in content-addressable memory (CAM) rather than ordinary DDR RAM?

*Ethernet switches keep their switching table in content-addressable memory (CAM), because the lookup has to be done at line speed. In ordinary RAM, a data structure such as a hash table would have to be used, which would be hard to implement at Gigabit speeds.*

## 6 RPC (10 pts)

- a) (4 pts) Where in your RPC implementation in project 2A did you – albeit loosely - apply the idea of application layer framing that is presented in the Clark/Tennenhouse paper?

This question refers to: [Architectural Considerations for a New Generation of Protocols](#), Clark & Tennenhouse (SIGCOMM 1990)

*The fact that you were very careful about how sending only one TCP segment per RPC request could be understood as poor man's application layer framing. It provided similar benefits to those discussed in the paper, namely better performance by reducing the number of roundtrips required, and simplified processing (e.g., avoided short reads wherever possible) at the receiver side.*

- b) (6 pts) More advanced remote object systems such as CORBA have the ability to support multiple classes (called interfaces) of RPC objects, as well as multiple instances of a given interface within the same ORB. Suppose you were to extend your RPC implementation in project 2A to support multiple instances of multiple classes, while keeping a single TCP port to reach your object request broker (ORB – the runtime system you developed to manage objects and handle RPCs.)

Describe two aspects of your implementation that would have to change as a consequence!

*Examples include*

- *Adding an instance id and (possibly) class id field to the request messages and reply messages*
- *Adding a notion of “interface” to the IDL and changing the language binding accordingly*
- *Adding a way to multiplex multiple requests over the single TCP connections, for instance by using multiple threads.*
- *Add a way to register/unregister multiple instances to the ORB, including a way to assign and find/lookup instance ids.*