Comprehensive Final Exam for Computer Networks (CNT 6215) Fall 2012

>>> SOLUTIONS <<<

Welcome to the comprehensive Final Exam for *Computer Networks*. Read each problem carefully. There are 10 required problems each worth 10 points. There is also an additional extra credit question worth 10 points. You may have with you a calculator, pencils and/or pens, erasers, blank paper, and one 8.5 x 11 inch "formula sheet". On this formula sheet you may have anything you want (definitions, formulas, homework answers, old exam answers, etc.) as **handwritten by you in pencil or ink** on both sides of the sheet. Photocopies, scans, or computer generated and/or printed text are not allowed on this sheet. Note to tablet (iPad, etc.) users – you may **not** print-out your handwritten text for the formula sheet. You have 120 minutes for this exam. **Please use a separate sheet of paper for the answer to each question**. Good luck and be sure to show your work!

Problem #1

1 pt for each item (2 pts for top level goal)

In his paper "The Design Philosophies of the DARPA Internet Protocols" Clark describes a top-level goal and seven second-level goals of the DARPA Internet Architecture. Give these goals in order of importance. How do these goals differ from an architecture designed for commercial deployment?

The top level goal is "effective technique for multiplexed utilization of existing interconnected networks". The second second level goals are:

- Communications continue despite loss of networks or gateways
- Support multiple types of communication service
- Accommodate a variety of networks
- Permit distributed management
- Be cost effective
- Easy host attachment
- Resource use accountable

A commercial deployment would consider accounting of resource use as a top priority (this is order to be able charge for usage and make money).

Problem #2

5 pts each sub-problem. 3 pts for correct formulas and set-up for each sub-problem.

Consider a 1000 mile link of 1 Mb/s capacity that is error-free. If this link is used by a single sender with a saturated queue sending 1250 byte frames, what will the link utilization be for the following cases (hint: signal propagation is roughly 5 μs per mile):

a) Stop-and-Wait protocol

Assuming that processing and ack transmission delays are negligible we have that:

$$U_{SAW} = \frac{t_{fr}}{2 \cdot t_{pr} + t_{fr}}$$

Where t_{fr} is the frame length in bits divided by link capacity and t_{pr} is the length of the link multiplied by signal propagation. So, for this link we have that $t_{fr} = (1250 \cdot 8)/10^6 = 10\ \text{ms}$ and $t_{fr} = 1000 \cdot 5 \cdot 10^{-6} = 5\ \text{ms}$, thus

$$U_{SAW} = \frac{10}{2 \cdot 5 + 10} = 50\%$$

b) Sliding Window protocol with a window size of 7

For sliding window we have that:

$$U_{SW} = max\left(1, \frac{W \cdot t_{fr}}{2 \cdot t_{pr} + t_{fr}}\right)$$

thus

$$U_{SW} = max\left(1, \frac{7 \cdot 10}{2 \cdot 5 + 10}\right) = 100\%$$

Problem #3

3 pts for small frames, 3 pts for many queued stations, and 4 pts for explanation of higher efficiency

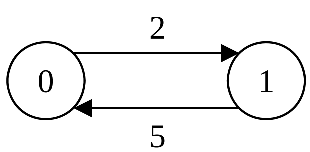
In his paper "Ethernet: Distributed Packet Switching for Local Computer Networks" Metcalfe has a result showing that Ethernet efficiency (or utilization) is about 37%. What were the key parameter values (or conditions) for this result and why might it be that actual Ethernets can achieve higher efficiencies than 37%?

A low efficiency comes from the analysis using a small frame size (48 bytes in the paper) and a large number of continuously queued stations (256 in the paper). Given the use of larger frame sizes and the presence of fewer saturated stations, the efficiency will be much higher. Simply increasing the frame size to 1024 bytes increases the efficiency to over 90%. It would be rare for an actual Ethernet to only use small frame sizes (data frames are usually large). In addition, artifacts of the BEB algorithm such as the short-term unfair "capture effect" also serve to increase efficiency even when using small frames.

Problem #4

3 pts for Q matrix and solution. 2 pts for uniformization. 3 pts for P matrix and solution. 2 pts for DTMC.

For the below two-state CTMC give the Q matrix and solve for steady state probabilities. Then convert the Q matrix to a P matrix using uniformization and solve it for steady state probabilities. Finally, give the DTMC corresponding to the P matrix.



The Q matrix is:

$$Q = \begin{bmatrix} -2 & 2 \\ 5 & -5 \end{bmatrix}$$

We write two equations in two unknowns (breaking dependency with sum of π is 1) for $0 = \pi Q$ and solve:

$$-2\pi_0 + 5\pi_1 = 0$$
$$\pi_0 + \pi_1 = 1$$

which solves to $\pi_0 = 5/7$ and $\pi_1 = 2/7$.

We can use uniformization, $P = I + (1/\gamma) \cdot Q$ where γ is the maximum absolute value in Q (5 in the case of the above example). So, we get

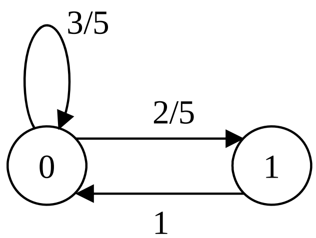
$$P = \begin{bmatrix} 3/5 & 2/5 \\ 1 & 0 \end{bmatrix}$$

We write two equations in two unknowns (breaking dependency with sum of π is 1) for $\pi = \pi P$ and solve:

$$(3/5)\pi_0 + \pi_1 = \pi_0$$
$$\pi_0 + \pi_1 = 1$$

which solves to $\pi_0 = 5/7$ and $\pi_1 = 2/7$ (as expected).

The corresponding DTMC is:



Problem #5 5 pts for latency dominates over capacity. 5 pts for proactive congestion control.

In his paper, “On the Modeling and Analysis of Computer Networks” Kleinrock says “Therein lies the fundamental change that come about with the introduction of gigabit links into nationwide networks. Specifically, we have passed from the regime (or pre-gigabit networking) in which we were *bandwidth limited*, to the new regime of being *latency limited* in the post gigabit world.” What does this mean? What implications does it have on the control of networks?

In a latency limited network it is latency, and not bandwidth or capacity, that dominates the response time performance. With long distance gigabit networks so many bits can be in flight that reactive congestion control is “too sluggish” and proactive congestion control (for example, giving users a maximum allowable transmit rate) becomes necessary.

Problem #6 2 pts for each subproblem.

Answer the following questions about queueing theory.

a) What is the condition for stability of a queue?

$\lambda < \mu$ (or $\lambda \leq \mu$ only for deterministic arrivals and service) where λ is the arrival rate of jobs and μ is the service rate.

b) State Little’s Law

$L = \lambda \cdot W$ where L is the number of jobs in the system, λ is the arrival rate of jobs, and W is the mean wait (or delay) of jobs in the system

c) FCFS is one possible service discipline. Name two others.

Last Come First Service (LCFS), priority, round robin, and processor sharing are other possible service disciplines (need only two of these!).

d) What does the P-K formula solve for?

The P-K formula solve the mean number of jobs (or mean wait or delay) for an M/G/1 queue.

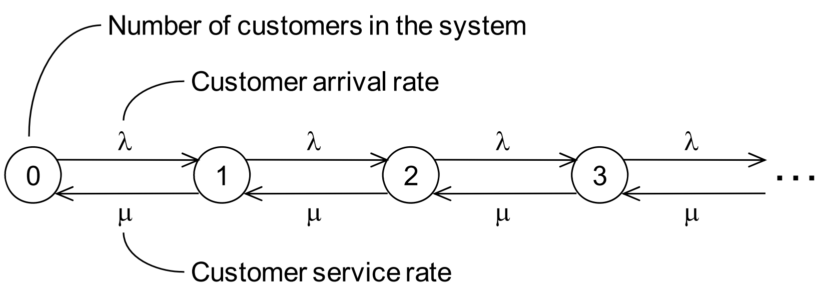
e) What do the Erlang-B and Erlang-C formulas solve for?

The Erlang-B formula solves for the probability of an arriving job to an M/M/m/m system being blocked (that is, all servers busy) and the Erlang-C formula solves for the probability of an arriving job to an M/M/m system being queued (that is, all servers busy). Said in another way, Erlang-B is for blocked calls cleared and Erlang-C is for blocked calls queued.

Problem #7 CTMC is 1 pt. Set-up is 5 pts. Correct steps and final expression is 4 pts.

Derive L (mean number of customers in the system) for the M/M/1 queue. Carefully show your work.

We first draw the Markov chain for the M/M/1 queue.



We solve the Markov chain for L. Once we have L we can easily find W using Little's Law ($L = \lambda W$). Solving for L, from local balance we write,

$$\pi_0 \lambda = \pi_1 \mu \Rightarrow \pi_1 = \left(\frac{\lambda}{\mu}\right)\pi_0$$

$$\pi_1 \lambda = \pi_2 \mu \Rightarrow \pi_2 = \left(\frac{\lambda}{\mu}\right)\pi_1 = \left(\frac{\lambda}{\mu}\right)^2 \pi_0$$

$$\pi_i = \left(\frac{\lambda}{\mu}\right)^i \pi_0 \text{ for } i = 1, 2, \ldots$$

We can now solve for π_0 with the knowledge that the sum of π_i must equal one,

$$1 = \pi_0 + \sum_{i=1}^{\infty} \left(\frac{\lambda}{\mu}\right)^i \pi_0$$

$$\pi_0 = 1 - \frac{\lambda}{\mu} = 1 - \rho$$

So, now we have the steady state probabilities as,

$$\pi_i = (1-\rho)\rho^i$$

The mean then follows directly from the definition of mean,

$$L = \sum_{i=0}^{\infty} i\pi_i = \sum_{i=0}^{\infty} i(1-\rho)\rho^i = \frac{\rho}{1-\rho}$$

Problem #8 5 pts for each line of code (one is changed, one is new).

In appendix A is a CSIM model for an M/M/1 queue. Modify this code such that it models an M/D/1/10 queue.

Solution is in the appendix.

Problem #9 Description and note of does not smooth out is 10 pts.

In the paper "On the Self-Similar Nature of Ethernet Traffic (Extended Version)" Leland et al. have a figure that is a "pictorial 'proof'" of self similarity of Ethernet traffic. It is noted that "A Picture is Worth a Thousand Words." Describe (or sketch) this figure. What does it show?

This figure shows Ethernet traffic and synthetic Poisson traffic at multiple time scales. At a small time scale both the Ethernet and Poisson traffic appear bursty. At a large time scale the Ethernet traffic is still bursty (and self-similar to itself at a small time scale), the Poisson traffic is smooth. This figure shows that Ethernet traffic is different from Poisson traffic in that it does not smooth-out.

Problem #10 2 pts for each item.

Answer the following questions about sensor networks.

a) What are the necessary and optional components of a WSN node?

Necessary components are: sensing unit, memory (or storage), processor, communications, and power source. Optional components include: power generating unit (solar, vibration, other), location finding (GPS or other), and a means of mobility.

b) What are the protocol stack layers and planes for a WSN node?

The layers are the usual physical, data link, network, transport, and application. The three planes are power management, mobility management, and task management.

c) Why are existing routing protocols not well suited for WSNs?

Existing routing protocols (RIP, OSPF) are too chatty thus making them energy inefficient.

d) Describe two general routing methods (or protocols) for WSNs.

Flooding with broadcast, gossiping to a random neighbor, clustering-based, and source routing (need only two).

e) What is IOT?

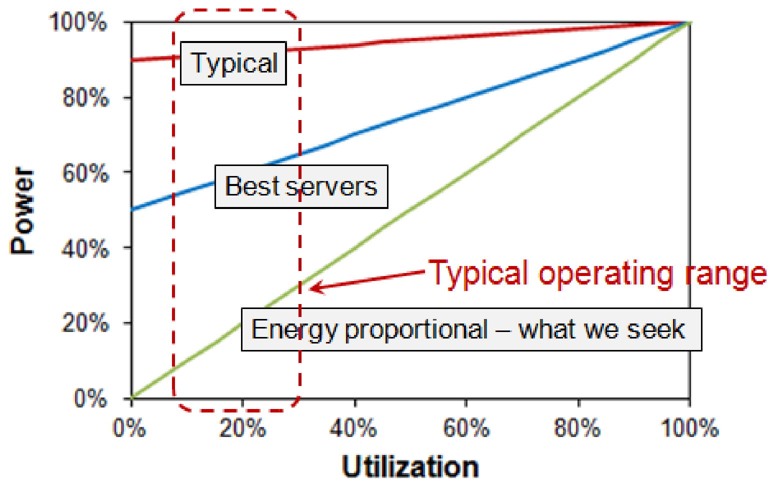
IOT = Internet of Things – a focus of future research where many of the things may be wireless sensor nodes.

Extra Credit

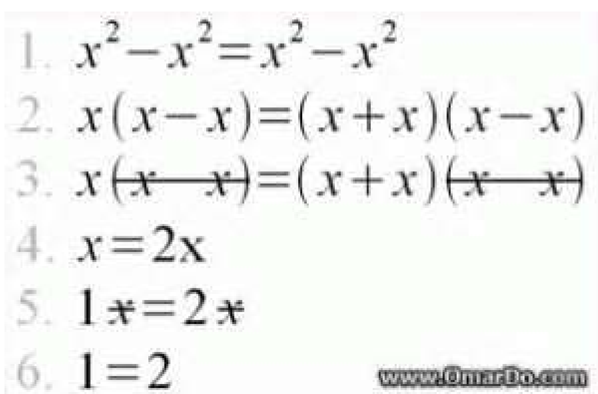
10 pts for concept.

Describe the concept of energy-proportional. Use a figure or graph as needed. Is current network equipment (routers, links, etc.) energy proportional?

In an energy proportional system the power use is directly proportional to utilization (the green line in the graph). This is what we seek. Existing systems usually consume about the same power when lightly utilized as when highly utilized (the red line in the graph).



Humor



From: http://www.picstopin.com/402/dear-math-jokes-fail/http:%7C%7Cimg*math-fail*com%7Cwp-content%7Cuploads%7Cmath1*jpg/

I hope that everyone did well ☺

Appendix A – Source code for Problem #8

```
#include <stdio.h>
#include "csim.h"

FACILITY Server;

void generate(double lambda, double mu);
void queue1(double service_time);

void sim(void)
{
  double lambda, mu;

  create("sim");

  Server = facility("Server");

  lambda = 0.90;
  mu = 1.0;

  generate(lambda, mu);
  hold(1.0e6);

  printf("============================================================= \n");
  printf("= Lambda   = %f cust/sec   \n", lambda);
  printf("= Mu       = %f cust/sec   \n", mu);
  printf("============================================================= \n");
  printf("= Utilization        = %f %%       \n", 100.0 * util(Server));
  printf("= Mean num in system = %f cust     \n", qlen(Server));
  printf("= Mean response time = %f sec      \n", resp(Server));
  printf("= Mean service time  = %f sec      \n", serv(Server));
  printf("= Mean throughput    = %f cust/sec \n", tput(Server));
  printf("============================================================= \n");
}

void generate(double lambda, double mu)
{
  double interarrival_time, service_time;

  create("generate");

  while(1)
  {
    interarrival_time = exponential(1.0 / lambda);
    hold(interarrival_time);

    service_time = exponential(1.0 / mu);
    queue1(service_time);
  }
}
```

Replace with `service_time = 1.0 / mu;`

```
void queue1(double service_time)
{
  create("queue1");

  reserve(Server);
  hold(service_time);
  release(Server);
}
```

Insert:

```
if ((qlength(Server) + num_busy(Server)) >= 10)
  return;
```