University of Waterloo
ECE 358, Spring 2017
Pencil-n-Paper Assignment 2

Shiranka Miskin, Dhruv Lal, Sam Maier

Problem 1

(a)

The two would be equivalent, since both would require the entire link bandwidth. In parallel it would take $3\frac{200}{150} + \frac{100000}{150} + (3\frac{200}{\frac{150}{10}} + \frac{100000}{\frac{150}{10}}) = 7377.33$ seconds, and serially it would take $3\frac{200}{150} + \frac{100000}{150} + 10(3\frac{200}{150} + \frac{100000}{150}) = 7377.33$ seconds.

(b)

Persistent provides minor gains, as a TCP connection would have to be established only once for all 11 transfers, rather than 11 times, saving 2 RTT delays for each TCP handshake. Compared to the thousands of seconds that the entire transaction requires, this gain is insignificant.

(c)

Bob will have an advantage, as his effective link bandwidth will be much greater than that of his peers. Each one of his many parallel connections receives $\frac{1}{14}$ of the bandwidth, while his peers get a total of $\frac{1}{14}$ of the bandwidth each.

(d)

Bob will no longer have an advantage as now the other users are in the exact same situation as himself, using parallel non-persistent HTTP.

Problem 2

(a)

By following aliases, `www.ibm.com` eventually resolves to `96.16.43.195`. `www.ibm.com` is not the actual host name of `96.16.43.195`, which is what `-x` returns.

(b)

Name	Type	Value
www.ibm.com	CNAME	www.ibm.com.cs186.net
cs186.net	NS	ns.events.ihost.com
www.ibm.com.cs186.net	CNAME	www2.ibm.com.edgekey.net
www2.ibm.com.edgekey.net	CNAME	www2.ibm.com.edgekey.net .globalredir.akadns.net
akadns.net	NS	a18-128.akadns.org
www2.ibm.com.edgekey.net .globalredir.akadns.net	CNAME	e2874.dscx.akamaiedge.net
dscx.akamaiedge.net	NS	n0dscx.akamaiedge.net
e2874.dscx.akamaiedge.net	A	96.16.43.195

Problem 3

(a)

If the server uploads the file to all of its peers at the same time at a rate of $\frac{u_s}{N}$, and $d_{\min} \geq \frac{u_s}{N}$, since the server is the bottleneck, the distribution time will be $\frac{NF}{u_s}$, the time it takes to send the total NF bits across from the server.

(b)

If $\frac{u_s}{N} \geq d_{\min}$, then if the server distributes to all of its peers at the same time at a rate of d_{min}, the distribution time will be $\frac{F}{d_{\min}}$.

(c)

Each peer could be connected to every other peer, and each peer p_i could transmit parts of the data it has to all other peers at a rate of u_i/N. The total rate of data being transferred through the network is $u_s + u_1 + \cdots + u_N$, therefore a given peer will be receiving data at a rate of $\frac{u_s+u_1+\cdots+u_N}{N}$ on average. The server wouldn't send the same bits to every client, that way each client receives data that the other clients do not have, and they can begin distributing that. Since $u_s < \frac{u_s+u_1+\cdots+u_N}{N}$, the server, which must send all F of its bits to distribute the entire file over the network, will be the bottleneck, and the distribution time will be $\frac{F}{u_s}$.

(d)

With the same setup as (c), if $u_s \geq \frac{u_s+u_1+\cdots+u_N}{N}$, then u_s is no longer the bottleneck and the distribution time is instead dependant on the average transfer rate. A given $u_i, i \neq s$ will not bottleneck the network as it need not send all F bits. A peer which uploads at half the

rate of the average could be compensated for by a peer which uploads at three halfs of the average. The distribution time will therefore be $\frac{F}{(u_s+u_1+\cdots+u_N)/N}$, which can be written as $NF/(u_s+u_1+\cdots+u_N)$.

Problem 4

Let $D(k)$ be the difference between a key k and its nearest neighbor

$$E[k] = \sum_{i=0}^{8} i \cdot \Pr\{D(k) = i\}$$

$$\begin{aligned}\Pr\{D(k)=0\} &= 1 - \Pr\{\text{All other points are different values than } k\} \\ &= 1 - \left(\frac{15}{16}\right)^4 \text{ (15 valid positions left to choose from)}\end{aligned}$$

$$\begin{aligned}\Pr\{D(k)=i,\, i\in[1,7]\} &= \Pr\{\text{All other points are } i \text{ or more away from } k\} \\ &\quad - \Pr\{\text{All other points are } i+1 \text{ or more away from } k\} \\ &= 1 - \left(\frac{2(i-1)+1}{16}\right)^4 - 1 - \left(\frac{2i+1}{16}\right)^4 \\ \Pr\{D(k)=8\} &= \Pr\{\text{All other points are at the furthest point from } k\} \\ &= \left(\frac{1}{16}\right)^4 \text{ (Only one valid position to choose from)}\end{aligned}$$

$$0 \cdot (1 - \frac{15}{16}^4) + 8 \cdot (\frac{1}{16})^4 + \sum_{i=1}^{7} i \left(\left(\frac{15-2(i-1)}{16}\right)^4 - \left(\frac{15-2i}{16}\right)^4 \right) \approx 1.57922$$

Problem 5

(a)

```
lookup(k):
    if (itsMe(k)) return p

    next_peer = FT_p[m]
    for i = 2 .. m:
        if FT[i] >= k:
            next_peer = FT_p[i - 1]
            break

   return next_peer
```

(b)

Each hop travels at least half the arclength from a starting node p and the peer r where r is the peer immediately before succ(k). Since the peers are evenly distributed along the domain, travelling at least half the arc length will also result in travelling at least half the number of peers between p and r. This gives us the upper bound on the worst case number of hops being $O(\log n)$ hops, where n is the number of peers. We can prove that this is a tight bound by constructing the case where r is the immediate predecessor of p. r will only be hopped to once p appears in the finger table of a peer that is calling `lookup`, and since each hop can at most halve the distance around the circle, it will take $O(\log n)$ hops to finally be able to check p again.