Doctoral Written Exam in Networking, Fall 2006

December 18, 2006

Answer all questions. All questions are of equal weight. Turn in your answers by Friday, December 22, at the same time you picked up your exam, to Jennifer Streubel at MCS-140G.

Brevity is the soul of wit. Be brief and to the point. This is a depth exam. So, your answers should show research maturity and depth. Do not pad your answers and do not write for the sake of writing.

You are free to consult any notes, books and papers during the examination?make sure to include your references. You must develop your own solutions, which might use existing ideas and techniques, as long as you cite them and clearly explain how these existing ideas/techniques fit within your own solution.

You are not allowed to consult with any person during the examination.

If you have any doubt as to the interpretation of a question, make a reasonable assumption and explain your interpretation in your answer. No explanations will be given during the exam.

1. Consider the following architecture for the distribution of content.

   There are two sets of nodes: origin servers, and receivers. Periodically, each origin server gets a file with some fixed size F, and each receiver desires a copy of this file. Initially none of the receivers has any portion of the file, but over time they can assemble portions of the file. A receiver can obtain portions of the file from any of the origin servers and from other receivers that have parts of the file. A node x in the system (origin server or receiver) can transmit bits at a maximum rate of $u(x)$ and can download bits at a maximum rate of $d(x)$. For simplicity, assume that there is no interference whatsoever between the nodes due to the "network cloud" connecting all the nodes to each other (in other words, the only restriction on communication are the values of $u(x)$ and $d(x)$ for all nodes x).

   Now consider the use of the above architecture for the distribution of streaming media (e.g., a movie or live broadcast, etc.) by dividing the media stream into video clips (i.e., "files") of size F, which are distributed as described above. Let T denote the time it takes to transfer a video clip of size F to all receivers. Distributing the video file using this approach would allow a playout rate of F/T.

   (a) Assuming that there is only one origin server s_0 and n receivers $s_1, s_2, \ldots s_n$, and assuming a fluid communication model provide an upper bound on the playout rate F/T in terms of the parameters given above (i.e., $n, u(x), d(x)$, for all x). Please provide proof of any bound you come up with. You may find the paper by Rakesh Kumar and Keith Ross on "Peer-Assisted File Distribution: The Minimum Distribution Time" at HotWeb

2006 (available from `http://photon.poly.edu/~rkumar04/mindbntime.pdf`) useful.

(b) The bound you computed above did not account for how the various receivers will figure out who to contact for what content. In other words, it is assumed that an oracle is able to "schedule" all the interactions between the various nodes. Assuming that the values of the various upload and download speeds are known, what scheduling algorithm do you propose to use? Clearly, your algorithm must attempt to make the playout rate as close as possible to the upper bound you computed in (a).

(c) The bound you computed above did not account for the topology of the "network cloud" between the various end systems (the origin servers and receivers). Discuss how knowledge of the topology or characteristics of that cloud be used to leverage solutions to the video distribution problem. Provide the most important references for the techniques you think could be useful. Explain how each such technique could be leveraged. Please be as specific as you can (e.g., give illustrative examples).

2. Any network protocol that buffers and/or retransmits data from a lower-level network can be viewed as adding an abstraction that exploits tradeoffs between delay, jitter (variability in delay) and loss rate. Such protocols could be viewed as "transformations" that take as input the underlay network delay and loss characteristics and project it as a different network with a different delay and loss characteristics (e.g., a different delay distribution and loss rate). Clearly, such abstractions are often layered, for example NFS over TCP over 802.11.

   For example, protocols using retransmissions (e.g., for reliability) could be seen as transformations that accept an input with a wide range of loss rates and produce distributions with narrower losses. For instance, in the case of the TCP protocol, the input loss rate could range between 0% and (say) 10% and the resulting loss rate is 0%. Along the same lines, these protocols could be seen as transformations that accept an input with a narrow delay distributions but generate distributions with long tails, due to use of exponential back-off, for example. This is the essence of the tradeoff. For instance, in the case of the TCP protocol, the input delay is one-way delay of an IP packet, which is more or less in the range of the network transmission and queuing delays, but the output delay for TCP/IP packets would have much longer tail due to retransmissions, timeouts, etc.)

   (a) What (if anything) happens when the assumption about narrow delay range is broken due to layering? For example, what kind of delay distribution would result if TCP ran over 802.11 (802.11 re-transmits with exponential back off, but only eight times before giving up)?

   Consider the problem of implementing a virtual tunnel between two end-points to provide some desirable property. A VPN is an example of such a tunnel whereby the desired property is that of securing the content, e.g., through encryption. Now consider the design of a new type of tunnels between two end-points X and Y whereby the desirable property is that of emulating a physical lossless link with fixed delay D and capacity C. By "lossless," I mean that if the link is used to send at a rate less than C, then there will be no losses. Clearly, the network cloud between X and Y does not satisfy these constraints. In particular, the round-trip time $r(t)$ between X and Y is a random variable (e.g., due to cross traffic) whereas the "capacity" between X and Y (assuming a TCP-friendly transport) would be a function of the loss rate $p(t)$ and the round-trip time $r(t)$ between X and Y.

In an attempt to implement the above-mentioned virtual link between X and Y, it was proposed that the end points X and Y use an elastic TCP (eTCP) approach, whereby a varying number (n) of parallel TCP-friendly flows would be established such that the aggregate bandwidth of the n flows is greater than C. This idea is fine, except that sending traffic on TCP-friendly flows implies that some of the packets will be lost – recall that the loss rate between X and Y was $p(t)$. Since the virtual link is desired to be lossless, it follows that X and Y must employ some sort of protocol to ensure reliable transmission (e.g., retransmission, FEC, ...) with clear implications on what is possible to achieve for C and D.

(b) Prove that it is impossible to construct a lossless virtual link using a underlying best-effort, TCP-friendly transport unless $C/D \to 0$.

(c) In light of the negative result in (b), and assuming that retransmission is used as the mechanism ensuring reliability between X and Y. Provide a lower bound on the loss rate of the virtual link implemented as described above as a function of $C, D, p(t)$, and $r(t)$.

3. Consider the design of an ad hoc network where the objective is to minimize power consumption without sacrificing user performance. You have been asked to examine alternative solutions to both network routing and MAC scheduling functionalities.

Routing Problem:

Routing is an important functionality that you are asked to re-examine. Specifically, you are told that the current routing design supports a relatively small-scale application that generates low-frequency events due to rare mobility coupled with strong MAC-layer reliability through adaptive power control, coding, etc. So, the network employs a link-state routing protocol where Link State Advertisements (LSA) are triggered (and flooded) only whenever a "significant" event happens. Although link errors are rare, LSA messages can still get lost, albeit at low probability, due to either transmission errors or due to collisions over the shared wireless channel. Such loss of LSA messages may cause inconsistency between the nodes' routing databases, which may lead to routing loops and consequently energy waste and bad user performance.

You have been asked to analyze and compare two alternative solutions to routing database consistency: (1) periodic flooding of LSAs, in addition to event-based floods, so that a lost LSA may be received on a subsequent retransmission; and (2) periodic database synchronization between neighboring nodes so a node may recover missing or up-to-date LSAs from one of its neighbors.

Your analysis should capture the effect of various system parameters, including the rate of periodic LSAs, size of the LSA message, number of nodes in the network, average node degree, rate of database synchronization, size of the database synchronization messages, rate of event-triggered LSAs, and probability of losing an LSA message. Assume and model an efficient database synchronization scheme such as the one in `http://self.d.free.fr/cv/publications/HETNET_2004.pdf`. State any assumptions you make and provide clear design guidelines on when either of the two approaches is better in terms of message transmission overhead.

Scheduling Problem:

Regarding scheduling, you are told that to conserve energy, nodes follow a sleep/wakeup slotted schedule. Thus, a node can only forward packets that are next-hop destined to a neighbor that is awake. To this end, a node maintains per-neighor queues. A problem arises when the next-hop of the packet to be transmitted to a neighbor has been updated in the meantime by the routing module. This means that, in order to support Just In Time (JIT) forwarding, this packet has to be re-enqueued into the new next-hop/neighbor's queue, and inserted in FCFS order if such service order is to be preserved.

(a) Describe the data structures needed to implement the above scheme, and express the computation and storage overhead of the scheme in terms of parameters such as number of neighbors, number of destinations, etc. Clearly state any assumptions you make.

(b) You are told that you may devise an alternative implementation whose overhead is constant or at least better than that of (a) under certain conditions, by relaxing certain assumption(s) made in (a) and of course, as long as they are reasonable! For example, it may be reasonable to relax the service order of packets so FCFS service order is required only among packets going to the same destination.

Describe the data structures needed to implement your alternative scheme, and express the computation and storage overhead of the scheme in terms of parameters such as number of neighbors, number of destinations, etc. Clearly state any assumptions you make.

4. Design a scheme of addressing that allows stateless routing. That is, each forwarding node maintains no knowledge about nodes that are not within one hop.

The method must be able to handle both wired and wireless (mobile) connections.

(a) describe how addresses are assigned to nodes (you may want to describe wired and wireless cases separately).

(b) describe the forwarding algorithm used in this scheme.

(c) describe what happens when a new node is connected to the network.

(d) describe what happens when two independently operating networks are connected (ie, via a new connection between existing nodes)

(e) describe what happens when wireless nodes are in range of more than one base station.

(f) describe how nodes discover each other's addresses.

(g) Give a general justification for your choices in (a)-(f). Describe what alternatives you considered and why you chose the ones you did. Be sure to cite all sources that you used.