The Structure of Information Networks	**Problem Set 2**
CS 685 Fall 2007	**Due September 28, 2007**

Two initial notes:

- For questions that ask for a formula in terms of some underlying variables, you should express the formula in closed form (i.e. not involving $\sum$ notation, or products of many terms) whenever this is possible. For example, we would consider "n^2" a better answer than "$\sum_{j=1}^{n}(2j-1)$," even though these both give the same value.
- For questions that ask you to provide an algorithm, you should also include a proof that the algorithm is correct.

---

(1) As we're in the process of discussing in class, a number of peer-to-peer systems on the Internet are based on overlay networks: rather than using the physical Internet topology as the network on which to perform computation, these systems run protocols by which nodes choose collections of virtual "neighbors" so as to define a higher-level graph whose structure may bear little or no relation to the underlying physical network. Such an overlay network is then used for sharing data and services, and it can be extremely flexible compared with a physical network, which is hard to modify in real-time to adapt to changing conditions.

Peer-to-peer networks tend to grow through the arrival of new participants, who join by linking into the existing structure, and this growth process has an intrinsic effect on the characteristics of the overall network.

Here's a simple model of network growth, for which we can begin to analyze some structural consequences. The system begins with a single node v_1. Nodes then join one at a time; as each node joins, it executes in a protocol whereby it forms a directed link to a single other node chosen uniformly at random from those already in the system. More concretely, if the system already contains nodes $v_1, v_2, \ldots, v_{k-1}$ and node v_k wishes to join, it randomly selects one of $v_1, v_2, \ldots, v_{k-1}$ and links to this node.

Suppose we run this process until we have a system consisting of nodes $v_1, v_2, \ldots, v_n$; the random process described above will produce a directed network in which each node other than v_1 has exactly one out-going edge. On the other hand, a node may have multiple in-coming links, or none at all. The in-coming links to a node v_j reflect all the other nodes whose access into the system is via v_j; so if v_j has many in-coming links, this can place a large load on it. To keep the system load-balanced, then, we'd like all nodes to have a roughly comparable number of in-coming links, but that's unlikely to happen here, since nodes that join earlier in the process are likely to have more in-coming links than nodes that join later. Let's try to quantify this imbalance as follows.

(a) Given the random process described above, what is the probability that v_j has no in-coming links? Give a formula in terms of n and j.

(b) Give a formula for the expected number of nodes with no in-coming links in a network grown randomly according to this model.

---

(2) Consider the following abstract model for a network that implements a kind of associative memory. There is a set of *sensor nodes* and a set of *storage nodes*; for simplicity, we will assume that each of these sets has the same size, denoted n. There is a bipartite graph G on the sensor and storage nodes: each edge has one end equal to a sensor node and one end equal to a storage node. Now, every time a sensor node records an observation, it writes it to each of the storage nodes to which it has an edge; and similarly, each sensor node is able to read all the observations recorded at any storage node to which it has an edge.

Given this set-up, we would like the graph G to have the following property:

> *($*) For every two sensor nodes u and v, there is some storage node w such that both u and v have an edge to w.*

If property ($*$) holds, then every sensor node will be able to read all the observations recorded by all the other sensor nodes. On the other hand, if property ($*$) fails to hold, then there are sensor nodes u and v such that neither can read the observations of the other. Thus, ($*$) is a crucial property for maintaining access to information by all sensor nodes in our network-based memory structure.

One way to achieve this property would be to have G be a complete bipartite graph, with an edge from every sensor node to every storage node. However, we can achieve property ($*$) using significantly fewer edges, and one way to do this is via a random construction.

Suppose that for some exponent α, we connect each sensor node and each storage node by an edge independently with probability $n^{-\alpha}$. Give a value α^* that is *critical* for property ($*$) in the following sense:

> *If $\alpha > \alpha^*$ then property ($*$) holds with probability converging to 0 as $n \to \infty$; and if $\alpha < \alpha^*$ then property ($*$) holds with probability converging to 1 as $n \to \infty$.*

Give a proof for your answer.

---

(3) Consider the following model for a collection of wireless devices trying to form a network using line-of-sight communication in an urban environment. The model is based on

a simple abstraction of the idea that there are n streets running east-west, n avenues running north-south, and wireless nodes can be placed at intersections of streets and avenues. Due to the line-of-sight constraints, any pair of these wireless nodes can communicate if and only if they are on the same street or the same avenue (since otherwise their line-of-sight would be blocked by a building).

More concretely, we have an $n \times n$ grid of equally spaced points in the plane (representing the intersections), and we have k wireless nodes residing at k of these points. Two nodes can communicate if and only if they belong to the same row or the same column of the grid. Based on this definition, we can build the *communication graph* G on the k nodes, with an edge connecting two nodes if they can communicate under this definition. Naturally, it would be nice if the graph G were connected, but it's easy to think of placements of the wireless nodes for which it won't be.

So here's the problem: given a grid as above, with k wireless nodes placed at points in it, you are allowed to place additional wireless nodes at other points of the grid with the goal of making the resulting communication graph G connected. (I.e. the communication graph on the union of the original and the new wireless nodes should be connected.) Give an efficient algorithm that places the minimum possible number of wireless nodes required to make G connected.

---

(4) We say that a path in a graph is *simple* if it does not repeat any nodes or edges. Finding a simple path of the maximum possible length in a graph (a "longest simple path") is a problem that has been known for a long time to be computationally hard. Despite this computational intractability, however, we can still ask and answer basic questions about the properties of longest simple paths.

(a) For example, notice that a graph can potentially have more than one longest simple path. (I.e. there can be multiple simple paths that are tied for the distinction of being longest.) Now, consider the following claim.

> *Claim: Given any connected, undirected graph G, and any two paths P and Q, each of which is a longest simple path in G, there must exist a node v that belongs to both P and Q. (In other words, no connected graph has two disjoint longest simple paths.)*

Decide whether you think this claim is true or false, and give a proof or a counter-example.

(b) We can ask a similar question about longest simple cycles: a simple cycle is a sequence of nodes such that (i) each node is joined to its successor in the sequence by an edge, and (ii) all nodes in the sequence are distinct except that the first node is equal to the last node. (So it's like a simple path except that it "cycles" back on the final step to the node where it began.) As with simple paths, a graph can potentially have more than one longest simple cycle. Now consider the following claim.

Claim: Given any connected, undirected graph G, and any two cycles C and D, each of which is a longest simple cycle in G, there must exist a node v that belongs to both C and D. (In other words, no connected graph has two disjoint longest simple cycles.)

Decide whether you think this claim is true or false, and give a proof or a counter-example.