Problem Set 2

Due: Wednesday, December 3, at the beginning of class
CSE/EE 461: Fall 2003
(3 pages)

1. Media Access

Your startup, skylab.com, has the brilliant idea to use rooftop satellite TV dishes as the basis of 2-way Internet service. Skylab's service is broadcast in the downward direction (all the ground stations can hear the satellite), and unicast in the upward direction. Ground stations can only talk to the satellite – they can't hear each other (the dishes are all on rooftops pointing upward!), but if more than one ground station transmits at the same time, the satellite will not be able to correctly receive the packet. None of the stations (ground or sky) can send and receive simultaneously. The propagation delay from ground to sky (and vice versa) is relatively large – about 200 milliseconds round trip. The bandwidth available is also large, around 100 Mb/s.

Your task is to design a media access protocol that efficiently arbitrates access to the shared media, enabling the ground stations and/or satellite to individually and/or collectively send at or near the full bandwidth available. You should not make assumptions about the traffic – the network needs to be designed to be efficient at sending traffic in any and all directions. Many designs are possible; we're looking for one that is as efficient as possible at statistically multiplexing the available bandwidth.

a) What protocol does your system use for communicating downward – from satellite to ground?

b) What protocol does your system use for communicating upward – from ground to satellite?

2. TCP Congestion Control

This question concerns the factors that affect TCP performance when the network bandwidth is not the bottleneck. We define the following constants:

RTT = round trip time, in seconds
MSS = TCP maximum segment (packet) size (in bytes)
P = packet loss rate (0 <= P <= 1
L = file size in maximum-sized TCP segments
W = advertised receive window size (in bytes)

You should assume that the network bandwidth is large enough that the packet loss rate is independent of the behavior of the sending node (e.g., its packets are dropped because of the behavior of other nodes) and that the round trip time is independent of the packet size.

You should also assume that once the file is split into TCP segments, the network does not further fragment the packets.

a) TCP small file performance

Devise an approximate formula, using the constants above and any others that you may find necessary or convenient, that specifies the latency of a small web transfer, from the time that the user clicks on a link in a web browser, until the last byte of the requested web page arrives at the client machine. (Note: the client web request fits into a single packet; the reply fits in L packets.) Please explain your reasoning.

For this sub-problem, you should assume that P = 0 and W >= L*MSS (that is, TCP stays in slow start throughout the transfer). However, you should model as carefully as possible all other TCP effects on small file performance, including the initial three-way handshake, initial window size, delayed acknowledgments, Nagle's algorithm, slow start, etc. (Note that the TCP rule during slow start is to add an additional packet to the existing congestion window size whenever a successful acknowledgement is received.)

b) TCP large file performance

Devise an approximate formula, again using the constants above and any others that you may find necessary or convenient, that specifies the steady-state bandwidth of a (very!) large file transfer, given a specified background loss rate. Please explain your reasoning.

Note that in steady state, TCP will increase its window by one (maximum-sized) packet every time an entire window is successfully transferred, and decrease its window by half every time a packet is lost. You should assume TCP fast recovery – that is, that the loss event is immediately detected, and transfers are resumed without delay from that point. Also assume that the round trip time is a constant -- it normally will vary somewhat as the network becomes congested, but that is a second-order effect. (Hint: the integral over one iteration of the TCP "sawtooth" defines how many packets are transferred between loss events, and therefore must equal the inverse of the loss rate.)

3. Traffic Engineering

A feature of the popular link state protocols in widespread use today, such as OSPF and IS-IS, is that an administrator can specify a "weight" for each link in the network. The protocols then compute routes that minimize the shortest path in terms of minimizing the sum of the link weights. In this way, for example, an ISP could encourage traffic to take low latency or high bandwidth paths (by giving those links relatively low weight) and avoid the inverse (by giving high latency or low bandwidth links relatively higher weight).

a) Suppose an ISP bought two parallel links between the same two routers (e.g., because there was too much load to be carried on a single link). Explain why one link could be completely idle while the other was completely full, even if the ISP is very smart about choosing link weights.

A further feature of these protocols is called "equal cost splitting" – that when two alternate paths from a node to a destination have the same (minimal) cost (sum of link weights), traffic is "split" or sent round-robin between the two (or more) paths, by sending each successive packet down a different path. This provides a crude form of load balancing to solve the problem in part (a).

b) Show that equal cost splitting is not a panacea, by devising a topology such that there are multiple paths from a source to a destination, but that no matter how an ISP chooses to assign link weights, some paths will be underutilized when other paths are full. (Note that since the equal cost splitting happens at the IP layer, invisibly to the transport layer, TCP will back off as soon as any of the equal cost paths become fully utilized.)

c) Explain how a combination of virtual circuits and weighted fair queueing (where weights are given to virtual circuits) can be used to fully utilize the network bandwidth between source and destination in your example from part (b). (Flexible traffic engineering is a central motivation behind the choice made by some ISPs to adopt virtual circuit technologies such as MPLS.)