1. An ISP decides to limit the rate of P2P traffic to a maximum of $r\%$ in any given link. If this is exceeded, the ISP sends TCP reset packets to the P2P connection endpoints.

   (a) Explain how this reduces the rate of P2P traffic?

   TCP reset will close the TCP connection. If the peers re-establish the TCP connection, their TCP window sizes will be reset and thus the rate of communication will be decreased.

   that's a common endpoint response, yes — easier than implementing more complex behaviour

   and P2P will likely use a different peer…

   (b) Discuss how you might decide, in response to changing network traffic conditions at runtime, how to decide what proportion of TCL connections carrying P2P traffic should be "reset". (Hint: using a P.I.D. response, based on control theory, is one option. Any others?)

   Let k be the percentage of P2P channels which are reset each second.

   - Make k a function of the rate of change of queuing delay for the average queuing delay of the more important packets. i.e if the average loss increases by 1 microsecond then we could reset 1% of the P2P connections.
   - Make k a function of the queuing delay for more important packets. For example, a simple function could have $k = d$ where d is the average packet delay of an important packet (given in microseconds) and clamped at 100%.
   - Run a sawtooth-like windowing protocol where we allow a percentage $r\%$ of the traffic to be P2P. Each second, if there is no packet loss due to congestion then increase the percentage by some small constant ϵ each second. However, if there is packet loss due to congestion, then halve the window size.

   both hard to measure if there are no 'important' packets.

   machine learning approaches? feature vector definition for a classifier? or deep learning?

   (c) What is hop-by-hop flow control?

   How-by-hop flow control is where every node on a network runs flow control on every link. The result is a flow which is highly responsive to changes and the flow which the endhosts actually see is the flow on the worst link (*i.e.* the link with the least capacity). This is comparatively easy to implement: but it adds a lot of state and work for each intermediate node. Furthermore, it requires every node and link to implement it for to work properly. You cannot rely on this for internet-based systems and thus hop-by-hop flow control is not used in IP networks.

   OK,t hat's closer to control theory

   independently

   (d) Why are different flow control techniques required at Layer 2 and Layer 3?

   Flow control in Layer 2 is implemented via a windowing protocol such that a fast sender does not overwhelm a slow receiver. The receiver advertises a flow control window, which represents the maximum number of packets that they can receive at once. The sender then has at most that many packets in transit at once.

   I'm not aware of flow control being employed in the Internet Layer.

   if there is no known fixed upper bound on link latency

   Latency of feedback? Representative feedback? CPU complexity vs link speed? State update method? Any feedback at all?

   (e) Why do we need multiple user/network utility metrics?

   Differentiated Services. There are many different types of traffic which are sent over the network. They may have different requirements. For example, a user may be playing a game and require that low latency; while another user may be doing bulk data transfer and may care only about the bandwidth. Unifying these into a single utility metric will leave many users with suboptimal performance. by using many different bandwidth requirements, we aggregate many

   user vs network vs systems views

   metrics: hard to find representative, measurable, usable, etc. metrics