Problem 9 (4 points):
Our goal in this problem will be to show that NTO is universally stable. We prove this by induction. Consider any \((w, \lambda)\)-bounded adversary with \(\lambda = 1 - \epsilon\) where \(\epsilon > 0\).

(a) Show, as the induction basis, that at any time, the maximum number of times a packet is delayed at its first edge is at most \(w/\epsilon\). Suppose otherwise, i.e. some packet experiences a delay of more than \(w/\epsilon\), and let \(e\) be any edge at which this happens. Let \(t\) denote the corresponding time step, and let us follow \(e\) backwards in time till we reach the first time point \(t'\) at which no packet is waiting at \(e\) that has not traversed any edge so far. Consider the sequence of time steps \(I = \{t' + 1, t' + 2, \ldots, t\}\), i.e. \(|I| = t - t'\). Argue that all packets that were moved along \(e\) during \(I\) must have been generated during \(I\). Use this to show that this means that \(\lambda(|I| + w) \geq |I|\), which leads to a contradiction when using our assumption above. Thus, any packet needs at most \((w + 1)/\epsilon\) steps to traverse its first edge. (2 points)

(b) Suppose that it has already been shown that every packet traveling along a path of length at most \(d\) needs at most
\[
\sum_{i=1}^{d} \frac{w + 1}{\epsilon^i}
\]
time steps for this. But suppose that it is possible for a packet to be delayed at least \((w + 1)/\epsilon^{d+1}\) times while waiting to cross its \((d + 1)\)st edge. Show in a similar argument as in (a) that in this case, for the sequence of time steps \(I\),
\[
\lambda \left( \sum_{i=1}^{d} \frac{w + 1}{\epsilon^i} + |I| + w \right) \geq |I|
\]
Use this to prove a contradiction. (2 points)

Thus, the same worst-case time bound can be shown for NTO as for SIS.

Problem 10 (3 points):
Show that for \(\lambda \geq 0.76\) there is a network and an adversary that causes NTG to be unstable. (Hint: use the same strategy as for FIFO.)

Problem 11 (3 points):
Use the techniques in the E-program “xy-mesh-router.e”, which is available on the course web page, to write an E-program that implements a routing service for the bit adjustment strategy in a hypercube.