# Theory of Network Communication

Fall 2002

# Solution to Assignment 5, problem 14

As for problem 13, the proof is still not entirely certain, so I omit any explanations.

### Problem 14:

The problem is to prove that NTG is not universally stable. The same counterexample graph and sequence of injections can be used as presented in the proof of FIFO; the inductive hypothesis is the same, and the initial step is the same as well. Stage 1 can proceed as in the FIFO proof, so that at the end of stage 1 we have all $\lambda m M_1$ packets blocked by the packets in $M_0$ and $\lambda m$ packets with remaining path $(f_0)$ in the buffer of $f_0$.

The central difference between NTG and FIFO is that FIFO is far more arbitrary. In FIFO, the $\lambda^2 m$ new packets injected with path $(f_0')$ in Stage 2 mix with packets in $M_1$, in the sense that the new packets may take turns since some of the new packets will have to be injected after some packets of $M_1$ arrive, and thus more packets from $M_1$ will be transmitted than is optimal (in the sense that we want to delay as many packets from $M_1$ as possible since they're terminal). But in NTG all the new packets will be preferred over the packets in $M_1$, and therefore we can alternate the arrivals so that even the early $M_1$ packets don't get through since later arriving packets with less distance to travel are pushed through ahead of the packets in $M_1$.

More thoroughly, after stage 2, consider the number of packets left in the buffer of $f_0$; there are $\lambda^2 m$, all of which have remaining path $(f_0 e_1 f_1)$ (the set $M_2$ in its entirety). Now consider the number of packets left in the buffer of $f_0'$: there are now $\lambda^2 m$ packets remaining from set $M_1$ with remaining path $(f_0' e_1 f_1)$, since those packets were unable to cross due to the $\lambda^2 m$ new packets which got in the way (since all new packets were routed ahead of the $M_1$ packets). So in the next part of the stage the packets in these two sets ($M_2$ and what's left of $M_1$ that was delayed) are routed through the edges $f_0$ and $f_0'$, and mix in the buffer of $e_1$. Then there are $2\lambda^2 m$ packets in the buffer of $e_1$ which have remaining path $(e_1 f_1)$. If this is greater than $m$ then we are done, since we have established an increasing sequence (since we began with $m$ packets in the buffer of $e_0$ which were to be sent along the path $(e_0 f_0)$). This is accomplished whenever $2\lambda^2 > 1$, which is true for $\lambda > \frac{1}{\sqrt{2}}$ or approximately 0.71.

(The 0.76 comes I think from using $\lambda^3 m$ new packets in stage 3 rather than the $\lambda^2 m$ packets which we used in stage 2 (set $M_2$). The final outcome is that you have $\lambda^2 m + \lambda^3 m$ packets in the buffer of $e_1$ waiting to cross instead of $2\lambda^2 m$; this new equation which is weaker is bad whenever $\lambda > 0.76$.)