Indirect TCP Analysis

- Some assumptions:
  - Cumulative acknowledgment is used (consequence: an entire window is lost if an acknowledgment fails)
  - Round trip time >> segment length
  - TCP timeout ~= round trip time
  - Indirect TCP is used, so packets are acknowledged when they arrive at the router

(Fig. 1)

- Say the downstream link can send up to k packets at once
- So the router buffer will accumulate packets if more than k packets are sent in a round trip time
- (Note that this assumption is not realistic to analyze slow start – in reality the router would advertise a window size of k, and congestion would never occur)
- What does this look like?

(Fig. 2)

- Area above dotted line is the contents of the router queue
- Say the router queue size is n – if the area above the dotted line is greater than n, packets are dropped

- If ACKs are dropped, all the packets in that column are dropped (due to cumulative acknowledgment)
- For convenience, choose n small enough so that router queue has a chance to drain during slow start
- Example. k=5, n=8.

(Fig. 3)

- Total throughput in steady state: 33 segments per 8 round trip times, or 4.125 segments per round trip time
- Note that link capacity is 5 segments per round trip time
- Now let's suppose that, after each slow start, the 10^{th} segment is lost to fading, and indirect TCP fails to return acknowledgments for these lost packets

(Fig. 4)

- Total throughput in steady state: 6 segments per 4 round trip times, or 1.5 segments per round trip time
- This is much less than the capacity of the link, even accounting for losses

- Now consider packet losses when indirect TCP is working. For each packet loss to fading, an extra transmission is needed from the router, so at worst we will add one to the size of the queue

(Fig. 5)

- steady state throughput is the same as for regular TCP
- so indirect TCP delivers a large improvement!