6.033 Spring 2016
Lecture #10

- Reliable Transport
- Window-based Congestion Control
Internet of Problems

How do we **route** (and address) scalably, while dealing with issues of policy and economy?

How do we **transport** data scalably, while dealing with varying application demands?

How do we **adapt** new applications and technologies to an inflexible architecture?
question: what is the correct value for $W$?

too small $\rightarrow$ underutilized network

too large $\rightarrow$ congestion
\textbf{question:} how can a single reliable sender, using a sliding-window protocol, set its window size to maximize utilization — but prevent congestion and unfairness — given that there are many other end points using the network, all with different, changing demands?
claim: AIMD is efficient and fair
AIMD + Slow Start

Window Size vs. Time (RTTs)

Retransmission due to timeout
• **TCP** provides **reliable transport** along with **congestion control**: senders increase their window additively until they experience loss, and then back off multiplicatively. Senders also use slow-start and fast-retransmit/fast-recovery to quickly increase the window and recover from loss.

• TCP has been a massive success, but **senders don’t react to congestion until queues are already full**. Is there a better way?