Network Layer
Dmitri Loguinov
Texas A&M University

April 3, 2024
Homework #3

- Part3 requires three threads in SenderSocket
  - `ss.Send()` is the producer into a bounded buffer of \( W \) packets (\( W = \) sender window)
  - Worker thread is the consumer from this buffer (ACK arrival that moves \( \text{sndBase} \) by \( X \) pkts releases \( X \) slots in buffer)
  - Requires two semaphores

Main thread

- `ss.Send()`
- `ss.Worker()`
- `ss.Stats()`

Application layer

Transport layer

Producer

Consumer

Shared buffer of pending packets
Homework #3

• Interesting aspect is how to release semaphore to accommodate flow control
  ─ Assume sndBase, nextSeq, window W are known
  ─ Receive ACK with sequence y > sndBase, recvWnd = R
  ─ By how much to release semaphore?

```c
lastReleased = 0;
sndBase = -1; // SYN-ACK 0 will move this to 0
while (not end of transfer)
{
    get ACK or SYN-ACK with sequence y, receiver window R
    if (y > sndBase)
    {
        sndBase = y
        effectiveWin = min (W, R)
        // how much we can advance the semaphore
        newReleased = sndBase + effectiveWin - lastReleased;
        ReleaseSemaphore (s, newReleased);
        lastReleased += newReleased;
    }
}
```
Chapter 4: Network Layer

Chapter goals:

• Understand principles behind network layer services:
  - How a router works (forwarding)
  - Routing (path selection)
  - Dealing with scale
  - Other topics: IPv6, multicasting

• Traceroute program as hw#4

• Big picture: