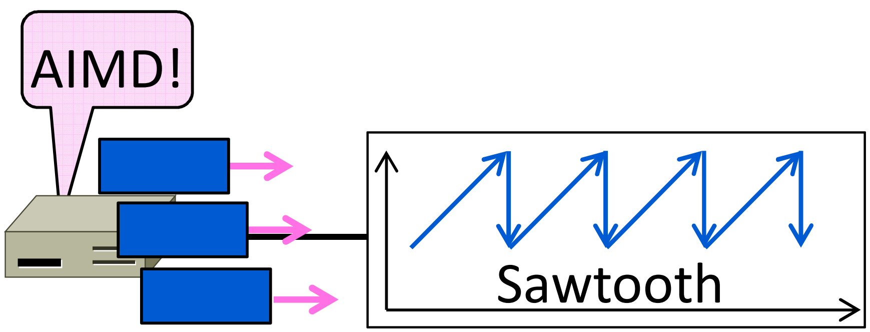
# Computer Networks

## Additive Increase Multiplicative Decrease (AIMD) (§6.3.2)

David Wetherall (djw@uw.edu)
Professor of Computer Science & Engineering
UNIVERSITY of WASHINGTON

# Topic

- Bandwidth allocation models
  - Additive Increase Multiplicative Decrease (AIMD) control law



# Recall

- Want to allocate capacity to senders
  - Network layer provides feedback
  - Transport layer adjusts offered load
  - A good allocation is efficient and fair
- How should we perform the allocation?
  - Several different possibilities …

# Bandwidth Allocation Models

- Open loop versus closed loop
  - Open: reserve bandwidth before use
  - Closed: use feedback to adjust rates
- Host versus Network support
  - Who is sets/enforces allocations?
- Window versus Rate based
  - How is allocation expressed?

TCP is a closed loop, host-driven, and window-based

# Bandwidth Allocation Models (2)

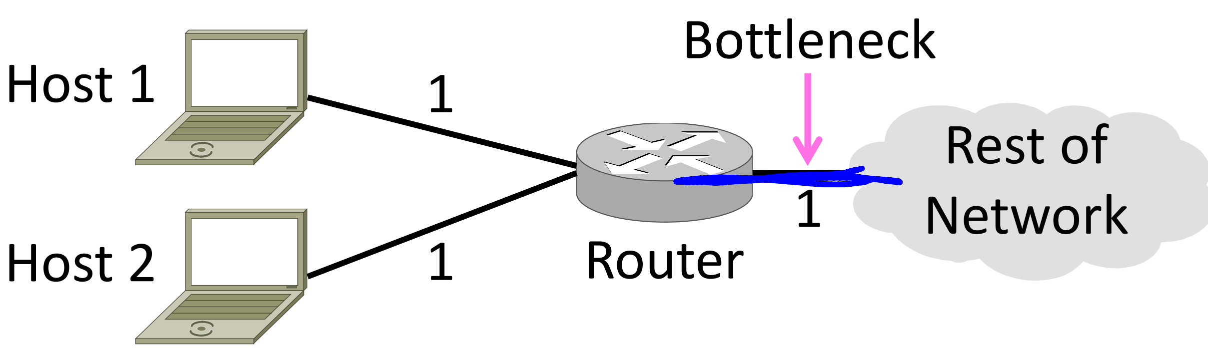
- We'll look at closed-loop, host-driven, and window-based too
- Network layer returns feedback on current allocation to senders
  - At least tells if there is congestion
- Transport layer adjusts sender's behavior via window in response
  - How senders adapt is a control law

# Additive Increase Multiplicative Decrease

- AIMD is a control law hosts can use to reach a good allocation
  - Hosts additively increase rate while network is not congested
  - Hosts multiplicatively decrease rate when congestion occurs
  - Used by TCP ☺
- Let's explore the AIMD game …

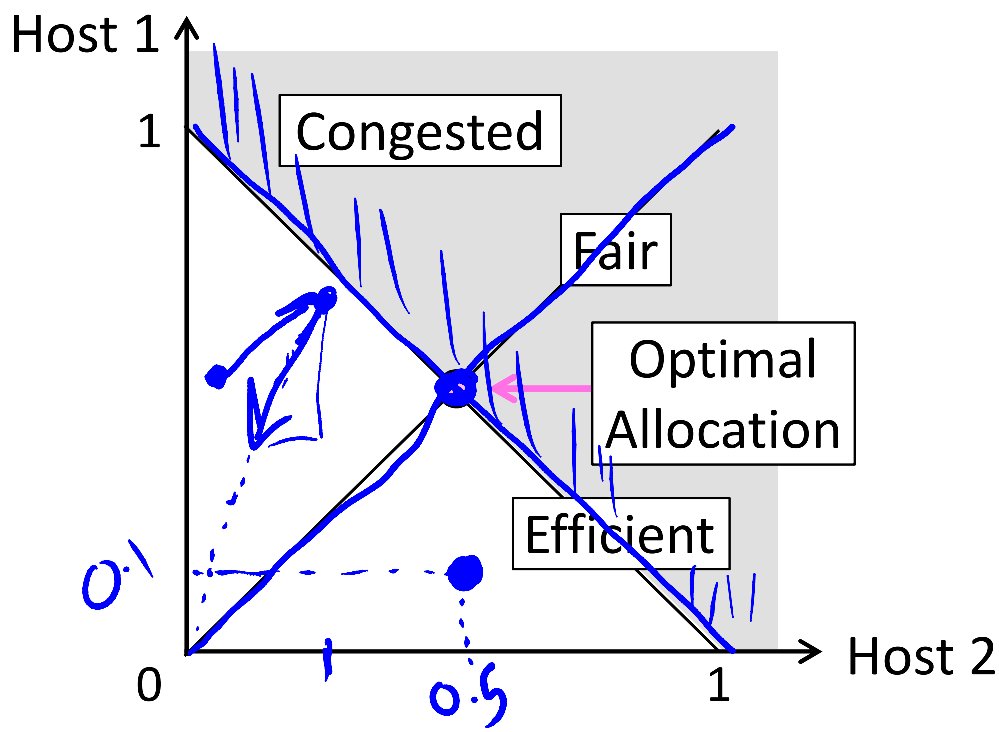
# AIMD Game

- Hosts 1 and 2 share a bottleneck
  - But do not talk to each other directly
- Router provides binary feedback
  - Tells hosts if network is congested



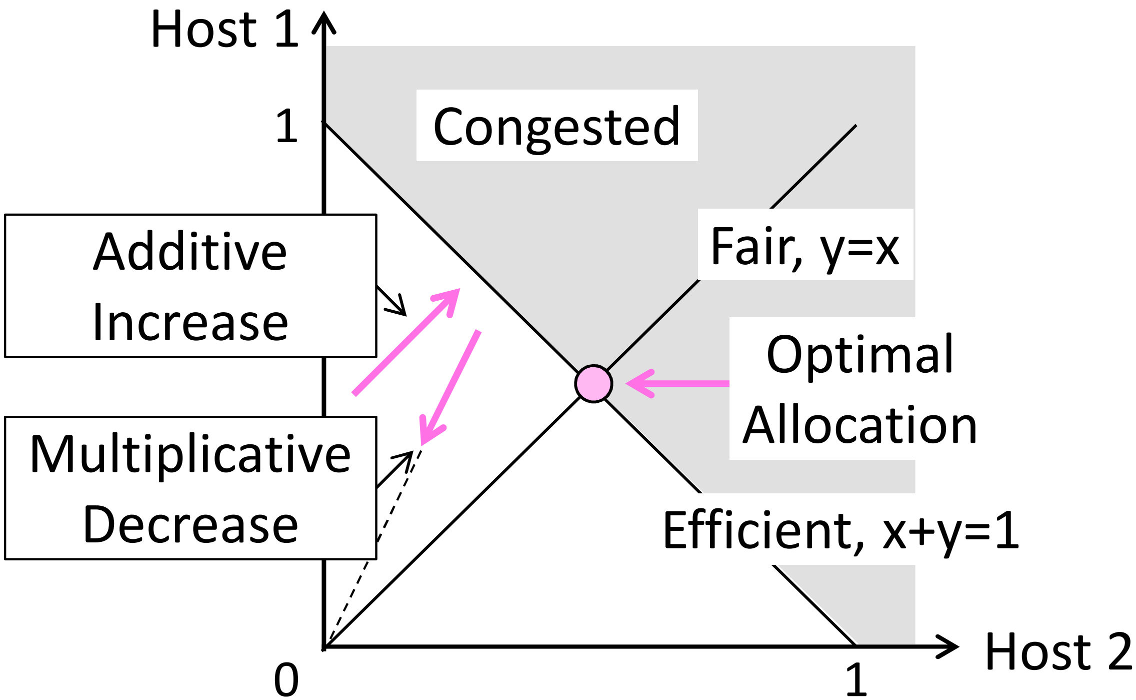
# AIMD Game (2)

- Each point is a possible allocation



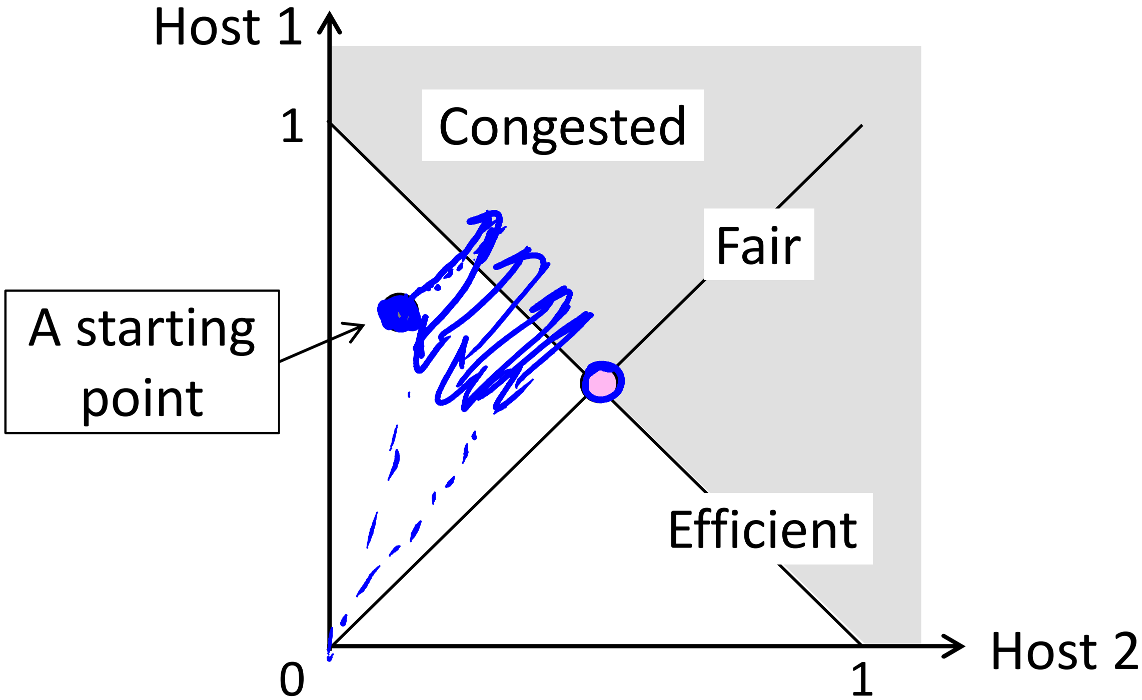
# AIMD Game (3)

- AI and MD move the allocation



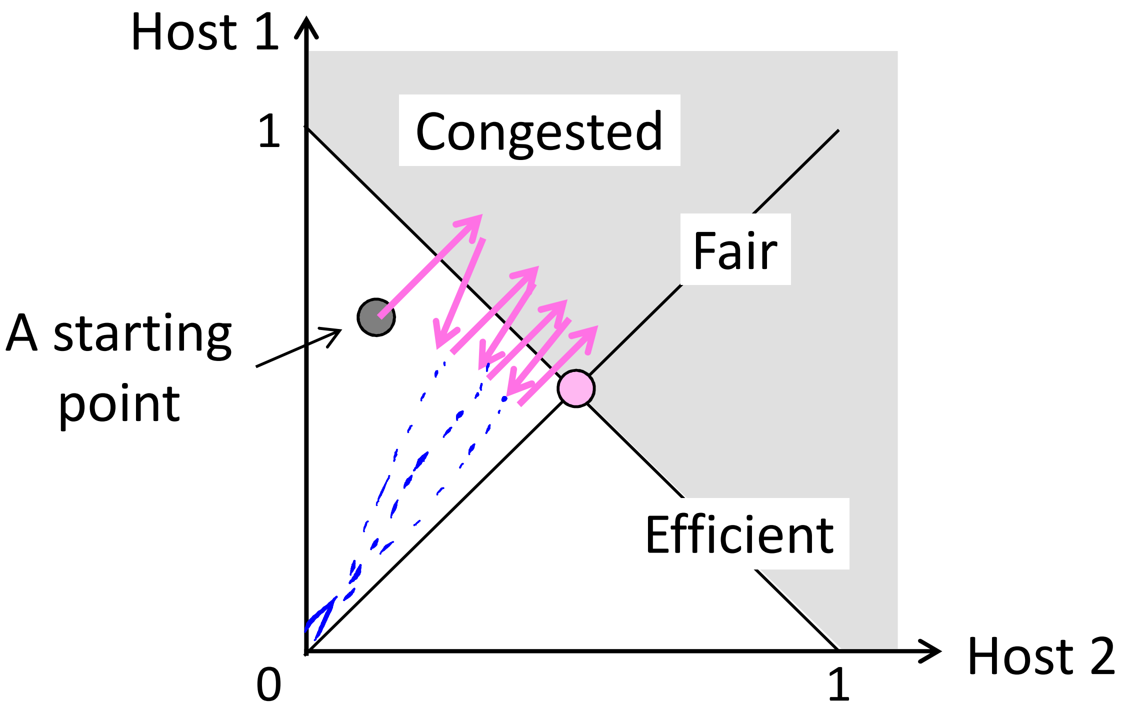
# AIMD Game (4)

- Play the game!



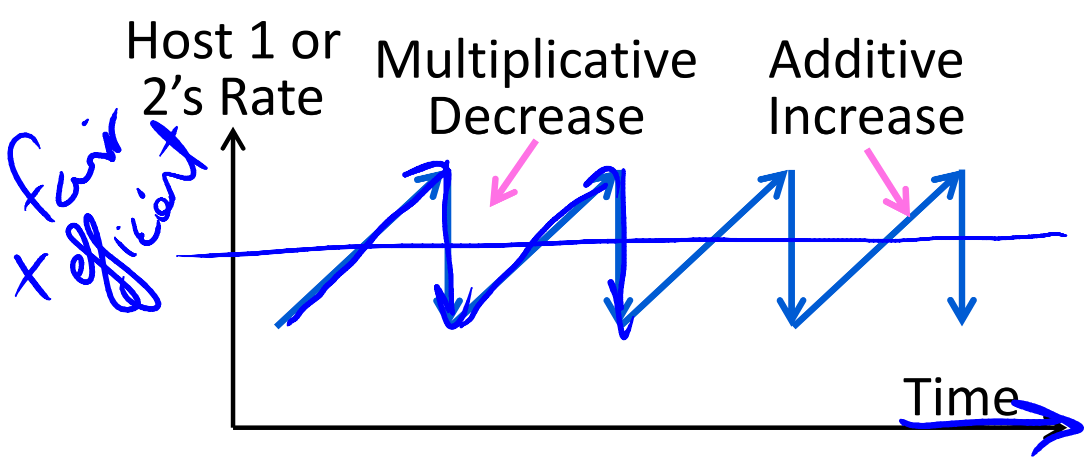
# AIMD Game (5)

- Always converge to good allocation!



# AIMD Sawtooth

- Produces a "sawtooth" pattern over time for rate of each host
  - This is the TCP sawtooth (later)



# AIMD Properties

- Converges to an allocation that is efficient and fair when hosts run it
  - Holds for more general topologies
- Other increase/decrease control laws do not! (Try MIAD, MIMD, ~~M~~IAD)
- Requires only binary feedback from the network

# Feedback Signals

- Several possible signals, with different pros/cons
  - We'll look at classic TCP that uses packet loss as a signal

| Signal | Example Protocol | Pros / Cons |
|---|---|---|
| Packet loss | TCP NewReno<br>Cubic TCP (Linux) | Hard to get wrong<br>Hear about congestion late |
| Packet delay | Compound TCP<br>(Windows) | Hear about congestion early<br>Need to infer congestion |
| Router indication | TCPs with Explicit Congestion Notification | Hear about congestion early<br>Require router support |

# END