Rate-Based Execution Models For Real-Time Multimedia Computing

Proportional Share Resource Allocation

Kevin Jeffay
Department of Computer Science
University of North Carolina at Chapel Hill
jeffay@cs.unc.edu
September 24, 1997

http://www.cs.unc.edu/~jeffay/courses/pisa

Rate-Based Execution Models For Real-Time Multimedia Computing

Outline

- Rate Based Execution: The case against Liu & Layland style models of real-time computing
- A Liu & Layland extension for rate-based execution?
- Fluid-flow models of resource allocation for real-time services
- Proportional share CPU scheduling
- On the duality of proportional share and traditional Liu & Layland style resource allocation

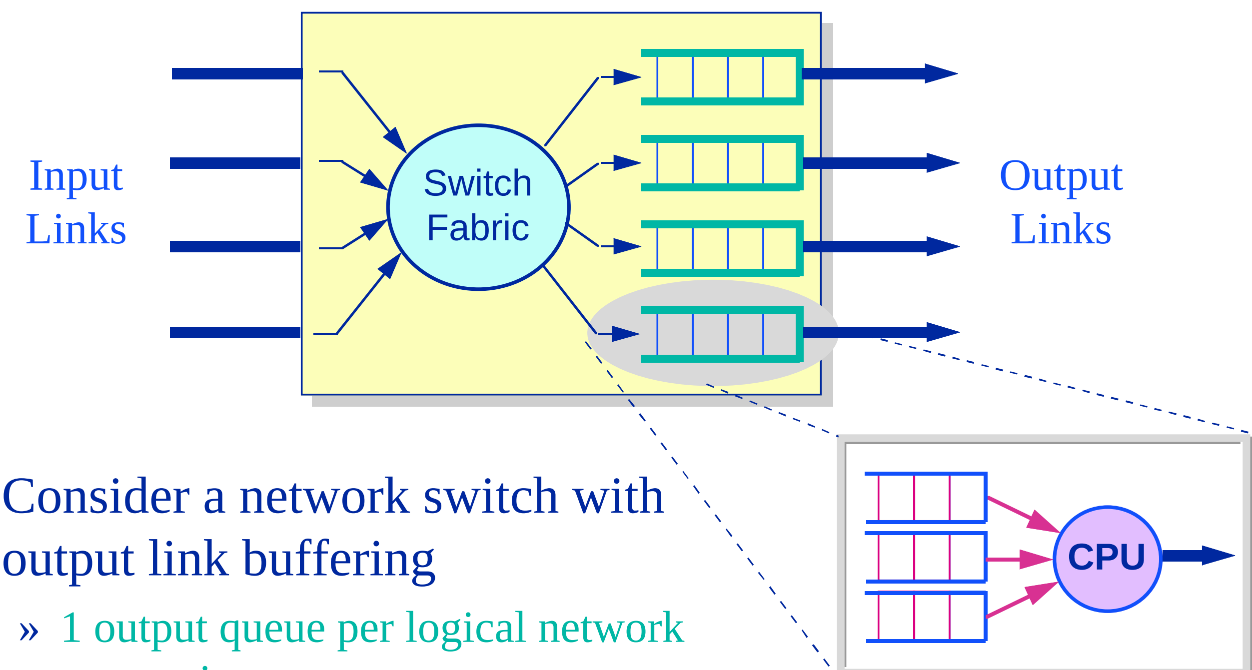
Proportional Share Resource Allocation

Outline

- Fluid-flow resource allocation models
  - » Packet schduling in a network
- Proportional share resource allocation models
  - » CPU scheduling in an operating system
- On the duality of proportional share and traditional real-time resource allocation models
  - » How to make a provably real-time general purpose operating system

Fluid-Flow Resource Allocation Models

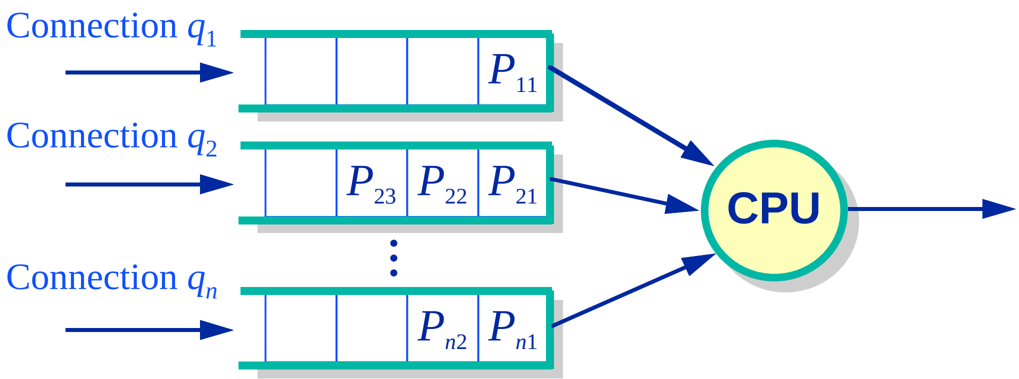
Packet/Cell scheduling in a network



- Consider a network switch with output link buffering
  - » 1 output queue per logical network connection

Fluid-Flow Resource Allocation Models

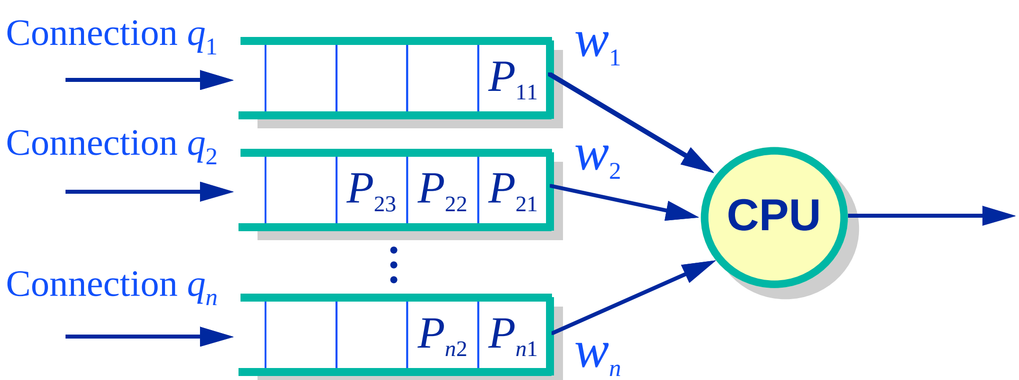
Packet/Cell scheduling in a network



- Generalized processor sharing (GPS)
  - » service proceeds in *bit-by-bit rounds*
  - » service 1 bit from queue *i* during each round
  - » provides *fair* allocation
  - » provides *isolation* from other connections

Fluid-Flow Resource Allocation Models

Generalized processor scheduling



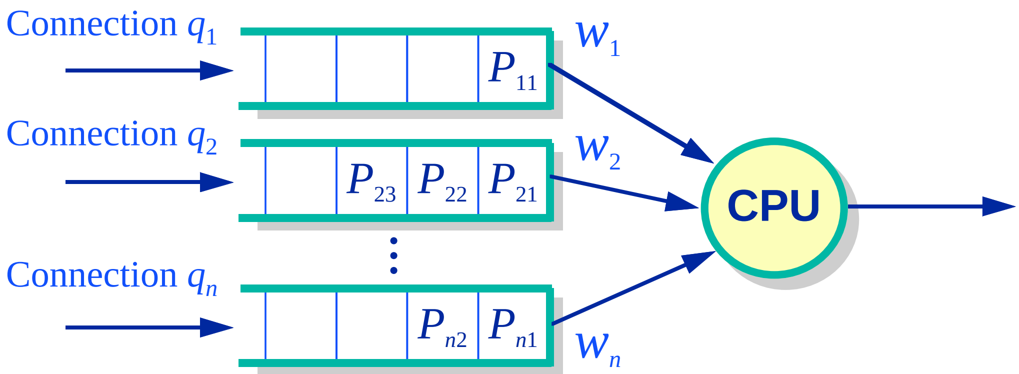
- Connections can be differentiated by integer weights
  - » w_i bits transmitted from connection *i* during each round
  - » each connection recieves a *fair share* of the link's capacity

$$\frac{S_i(t_1, t_2)}{S_j(t_1, t_2)} \geq \frac{w_i}{w_j}$$

where $S_i(t_1, t_2)$ is the number of bits transmitted from connection *i* during $[t_1, t_2]$

Approximations to GPS

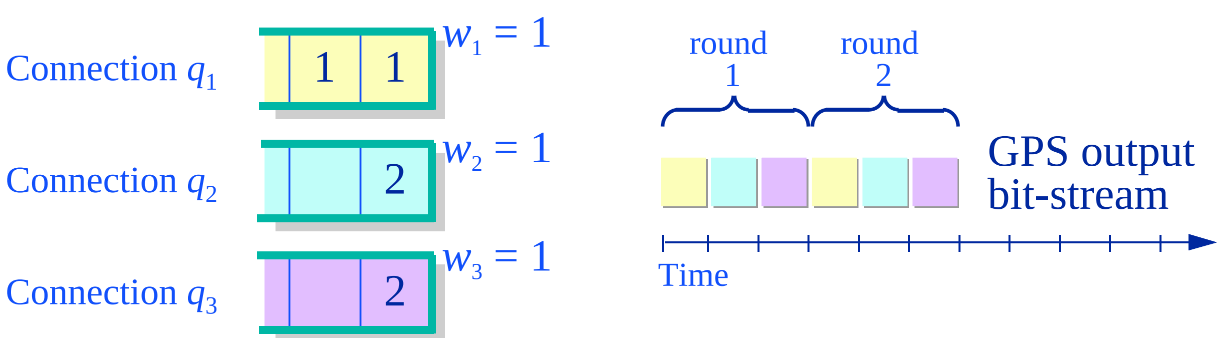
Packet-by-packet generalized processor sharing (PGPS)



- Transmit packets as whole units rather than bit-by-bit
- Simulate GPS by scheduling packets by *finish number*
  - » the time a packet would have completed transmitted in a GPS system
- Packets with smaller finish number are scheduled before packets with larger finish numbers

Packet-by-Packet GPS

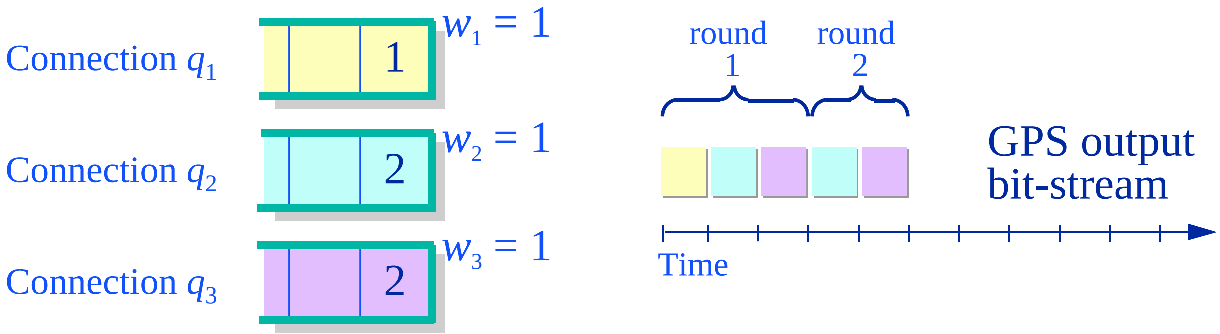
Finish numbers example



- Model GPS as proceeding in *rounds*
  - » $\Sigma_i w_i$ bits are transmitted in each round
  - » *finish number* = *round number* in which the last bit of a packet is transmitted

Packet-by-Packet GPS

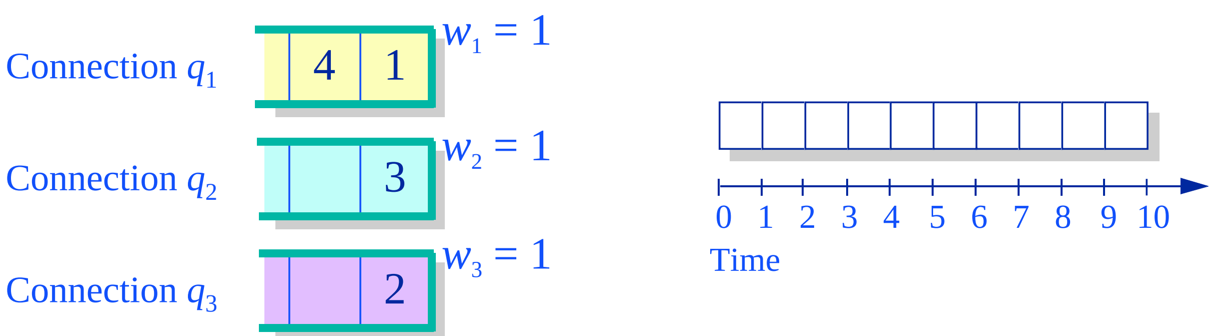
Finish numbers example



- Rounds proceed at a variable rate depending on the availability of work
  - » thus finish numbers do not necessarily deterministically map onto real-time values

Packet-by-Packet GPS

Finish numbers example

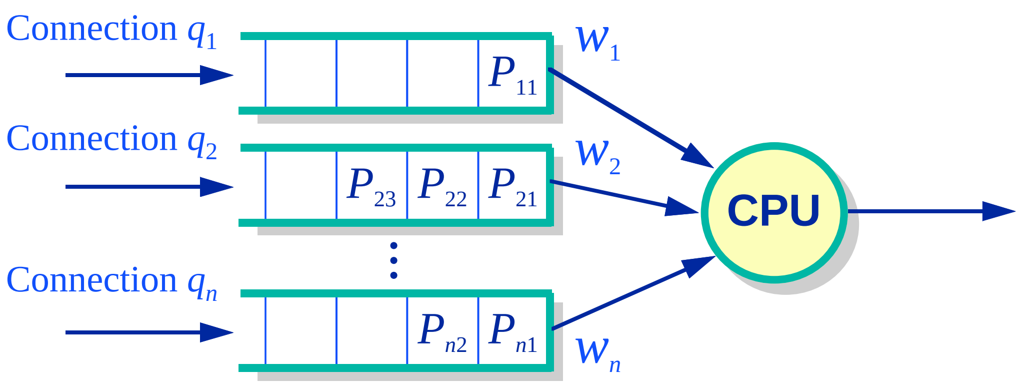


- Finish numbers of packets are:
  - » P_{11} =
  - » P_{12} =
  - » P_{21} =
  - » P_{31} =

    (Assume the second packet from connection q_1 arrives at time 4)

Packet-by-Packet to GPS

Finish number computation



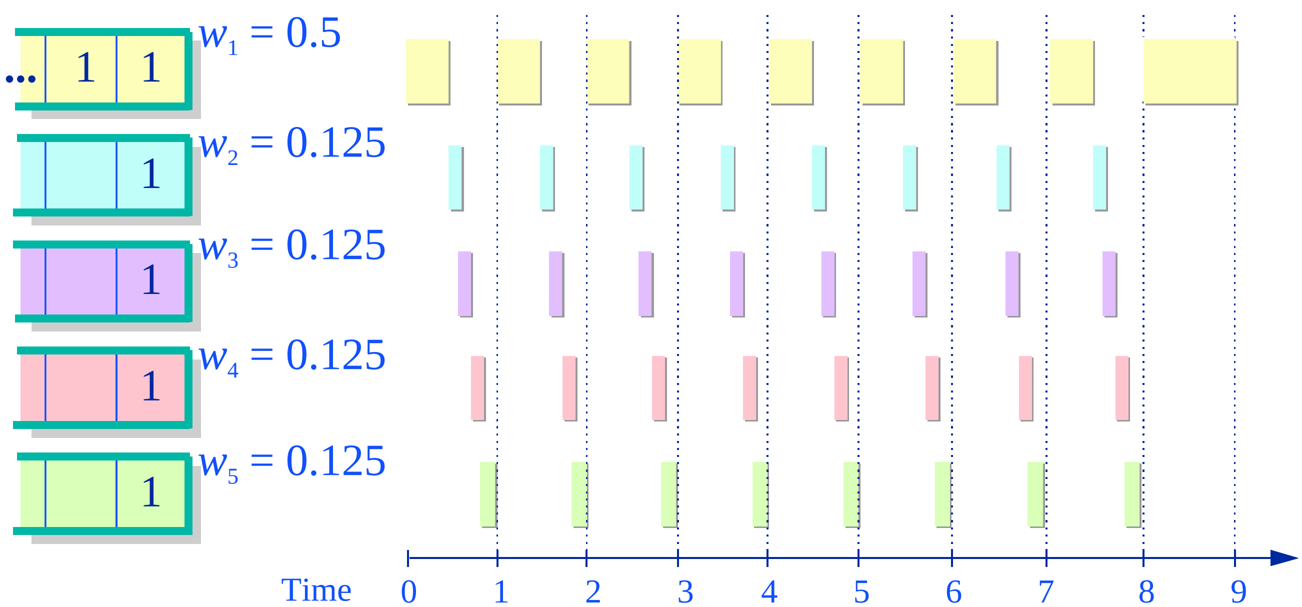
- Transmit packets by *finish number F*

$$F_{ik} = \max(F_{ik-1}, R(t)) + p_{ik}/w_i$$

  - » where t is the arrival time of the packet
  - » $R(t)$ is the round number at time t
  - » p_{ik} is the size of packet P_{ik}

Packet-by-Packet GPS Example

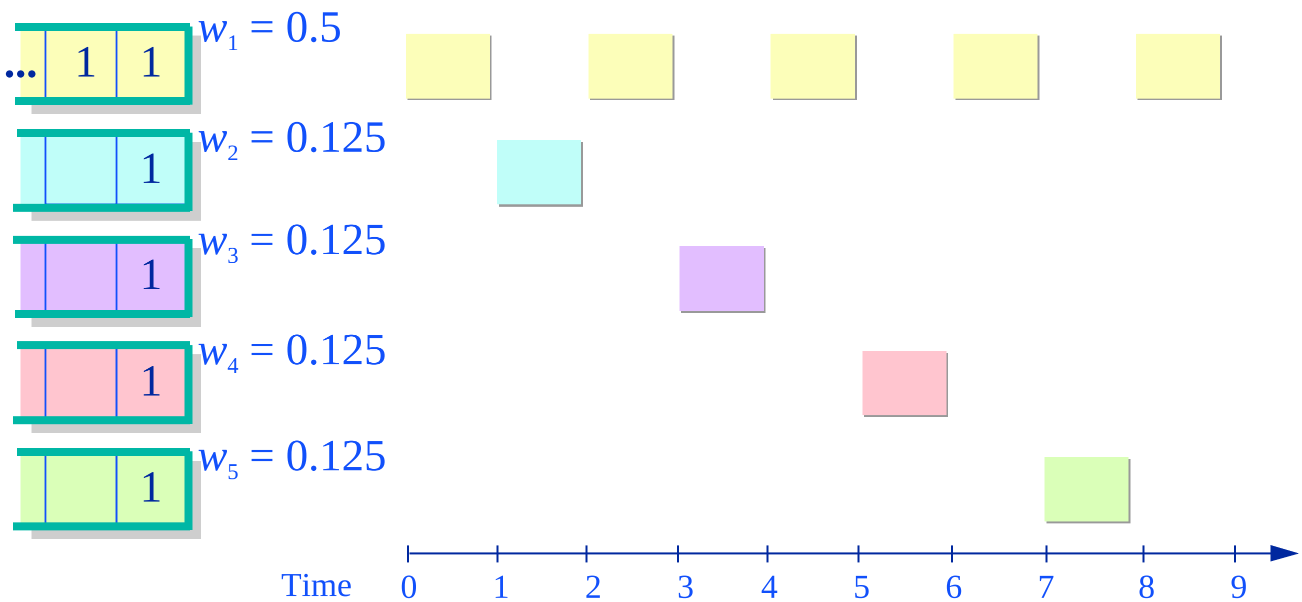
Comparison with GPS



- GPS schedule for 5 connections
  - » Packets for connection 1 arrive every 2 time units

Packet-by-Packet GPS Example

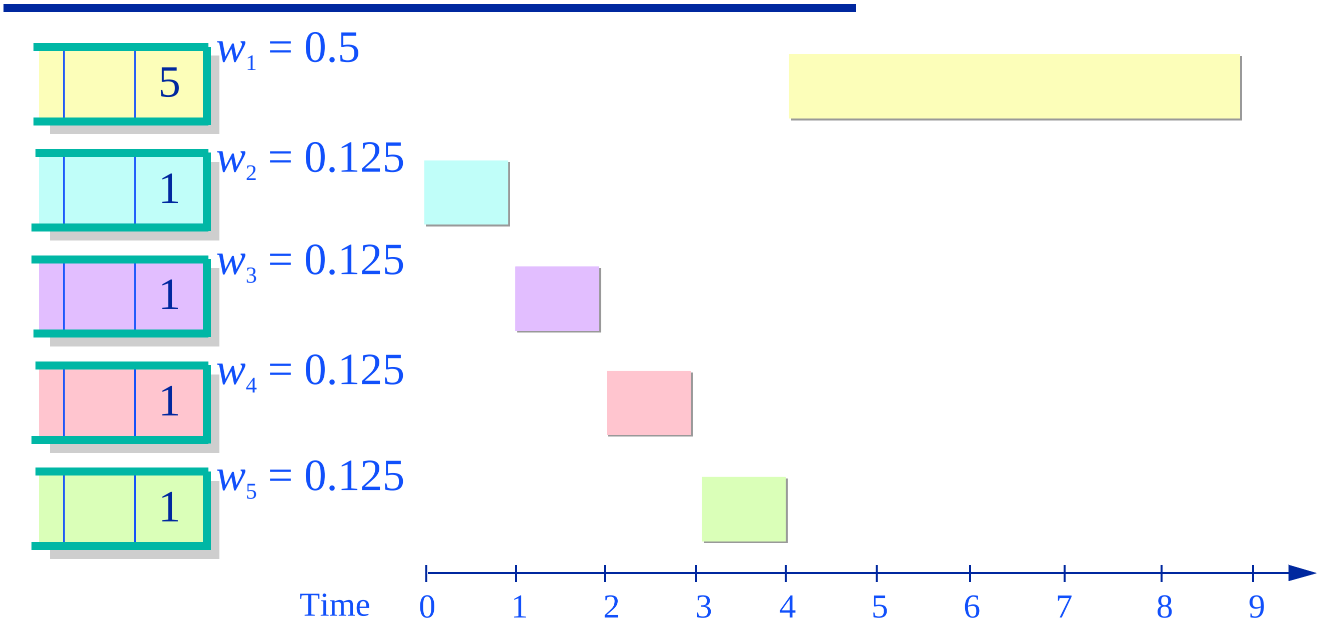
Comparison with GPS



- Example: PGPS schedule for 5 connections
  - » Packets for connection 1 arrive every 2 time units

Packet-by-Packet GPS Example

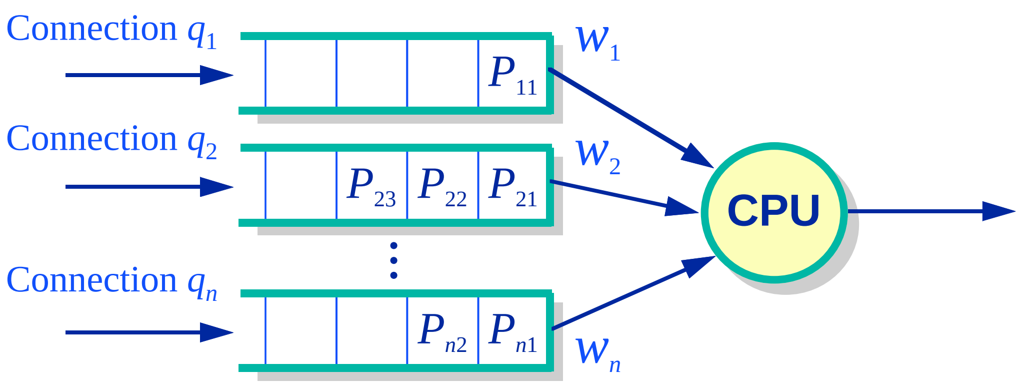
Comparison with GPS



- Example: PGPS schedule for 5 connections
  - » connection 1 has a single packet that is 5 times as large

Packet-by-Packet to GPS

Summary



- PGPS allocates link capacity *fairly* and *uniformly*
  - » a connection is guaranteed to receive a share of the link's capacity equal to

$$\frac{w_i}{\Sigma_j w_j}$$

Packet-by-Packet GPS

What does this have to do with real-time?!

- Under certain assumptions about the distribution of packet arrivals...
- Connections can acheive bounded end-to-end delay in a network of PGPS scheduled switches
- And this bound is (asymptotically) independent of:
  - » the number of connections in the network
  - » the number of switches in the network