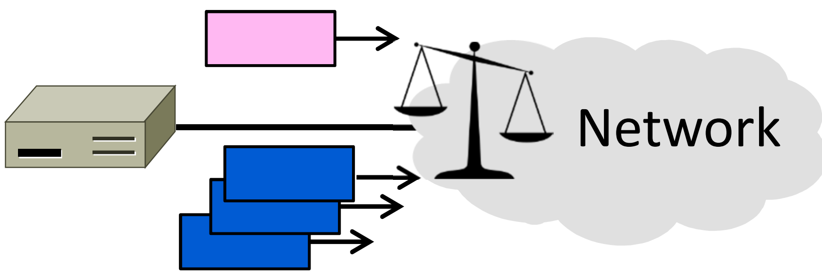
Introduction to Computer Networks

Fair Queuing (§5.4.3)

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

Topic

- Sharing bandwidth between flows
  - WFQ (Weighted Fair Queuing)
  - Key building block for QOS

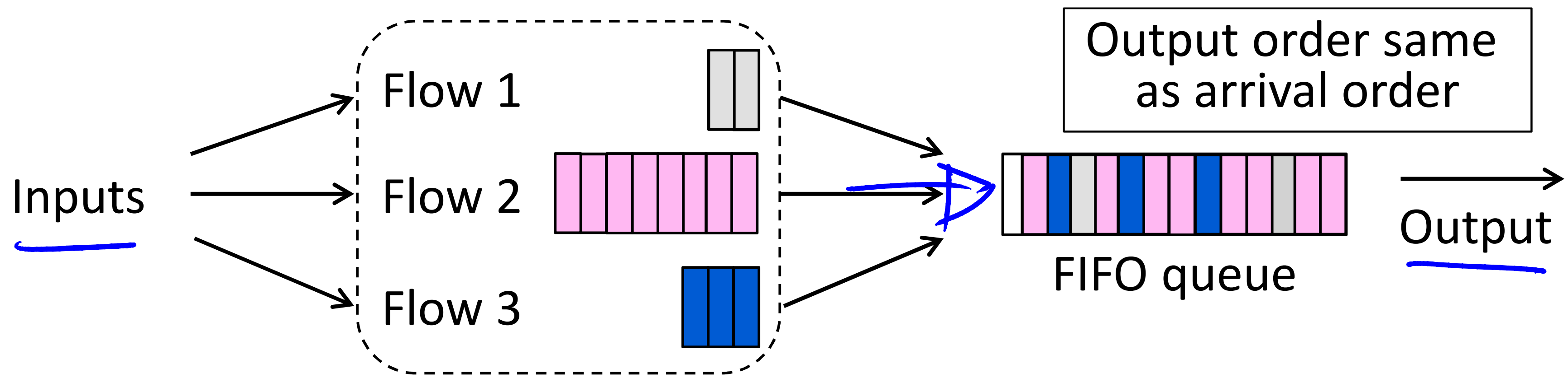


Sharing with FIFO Queuing

- FIFO "drop tail" queue:
  - Queue packets First In First Out (FIFO)
  - Discard new packets when full
  - Typical router queuing model
- Sharing with FIFO queue
  - Multiple users or flows send packets over the same (output) link
  - What will happen?

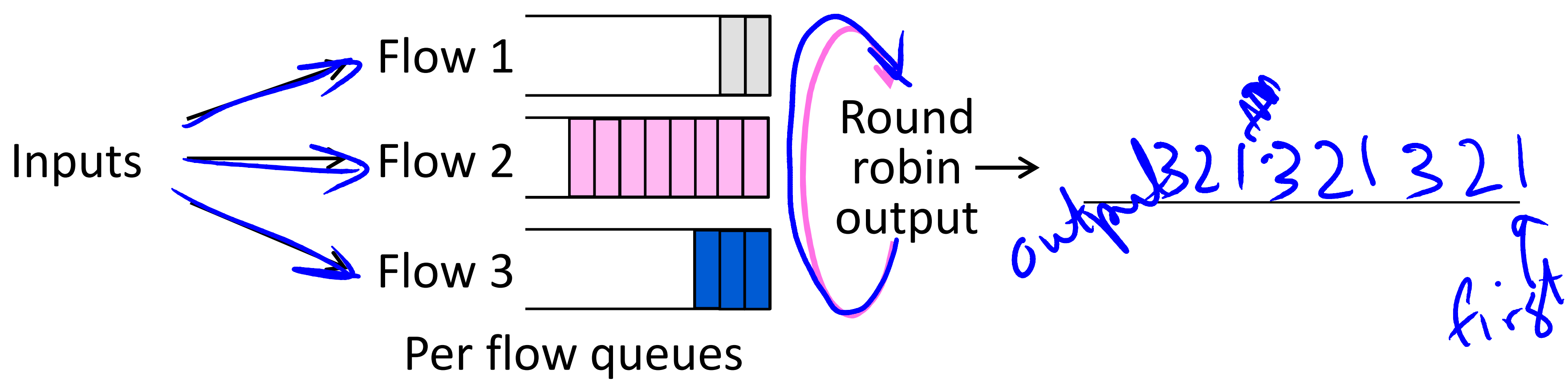
Sharing with FIFO Queuing (2)

- Bandwidth allocation depends on behavior of all flows
  - TCP gives long-term sharing – with delay/loss, and RTT bias
  - Aggressive user/flow can crowd out the others



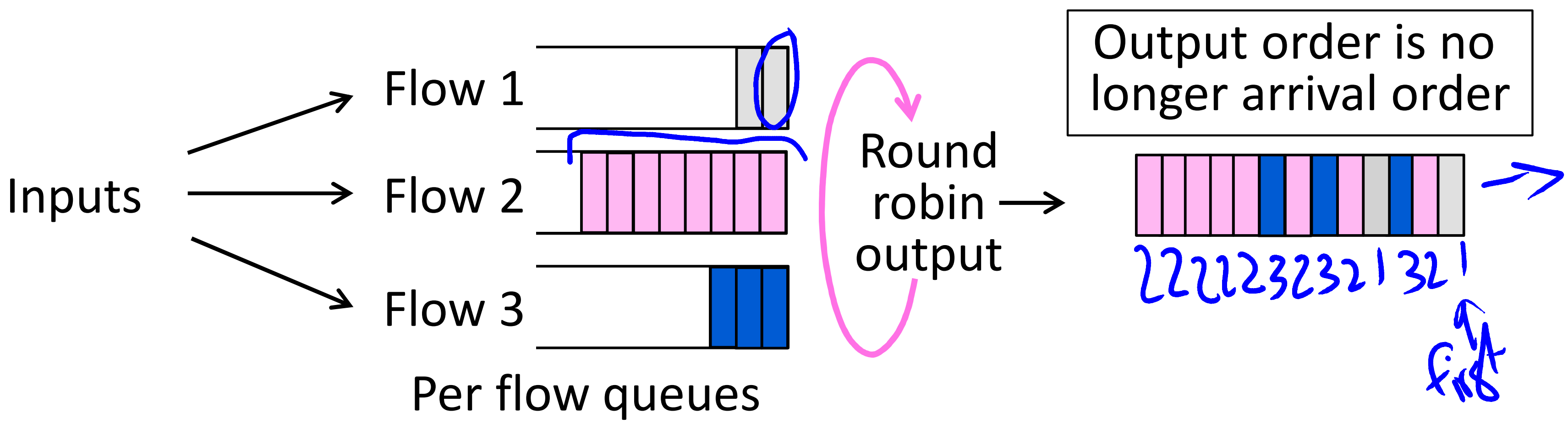
Round-Robin Queuing

- Idea to improve fairness:
  - Queue packets separately for each flow; take one packet in turn from each non-empty flow at the next output time



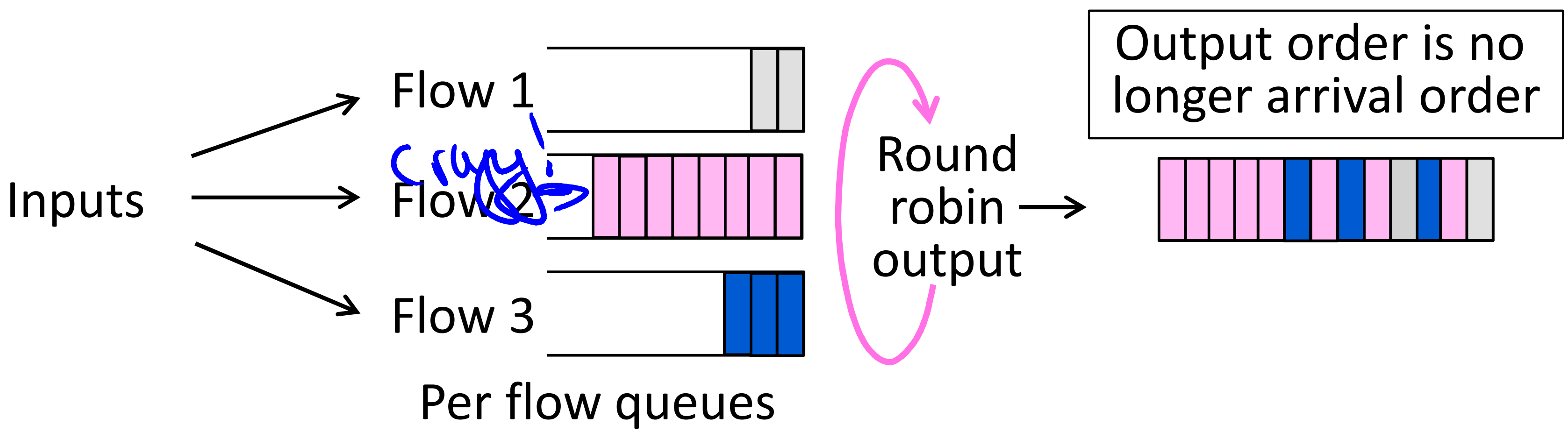
Round-Robin Queuing (2)

- Idea to improve fairness:
  - Queue packets separately for each flow; take one packet in turn from each non-empty flow at the next output time
  - How well does this work?



Round-Robin Queuing (3)

- Flows don't see uncontrolled delay/loss from others!
- But different packet sizes lead to bandwidth imbalance
  - Might be significant, e.g., 40 bytes vs 1500 bytes



Fair Queuing

- Round-robin but approximate bit-level fairness:
  - Approximate by computing virtual finish time
  - Virtual clock ticks once for each bit sent from all flows
  - Send packets in order of their virtual finish times, $Finish(j)_F$
  - Not perfect – don't preempt packet being transmitted

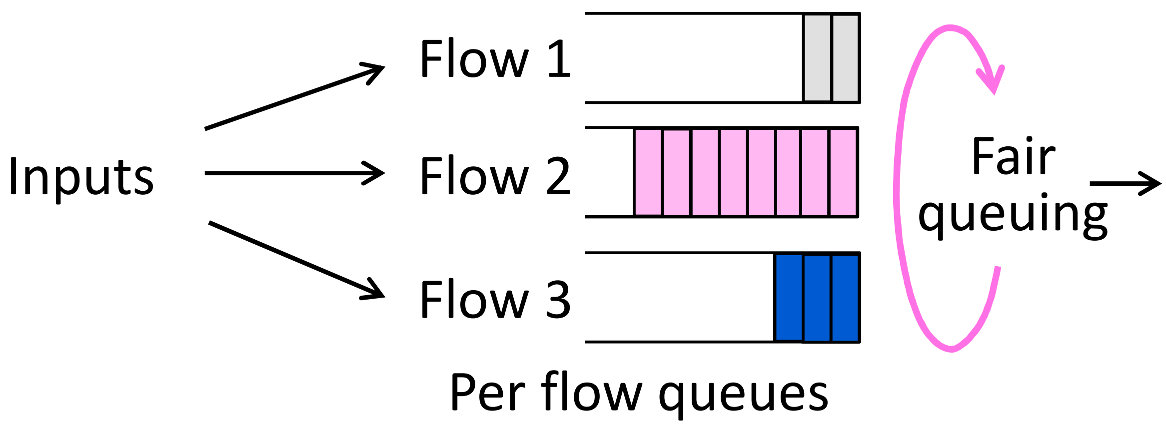
$Arrive(j)_F$ = arrival time of j-th packet of flow F

$Length(j)_F$ = length of j-th packet of flow F

$Finish(j)_F = \max(Arrive(j)_F, Finish(j-1)_F) + Length(j)_F$

Fair Queuing (2)

- Suppose:
  - Flow 1 and 3 use 1000B byte packets, flow 2 uses 300B packets
  - What will fair queuing do?



? compute finish times

first

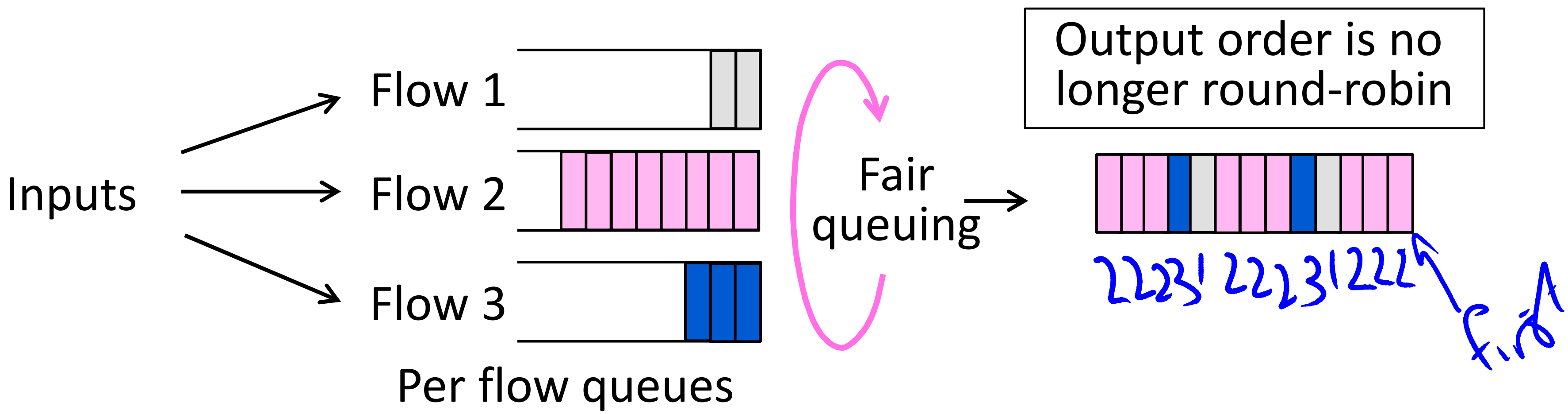
Fair Queuing (3)

- Suppose:
  - Flow 1 and 3 use 1000B packets, flow 2 uses 300B packets
  - What will fair queuing do?

Let $Finish(0)_F=0$, queues backlogged [$Arrive(j)_F < Finish(j-1)_F$]
$Finish(1)_{F1}=1000$, $Finish(2)_{F1}=2000$, ...
$Finish(1)_{F2}=300$, $Finish(2)_{F2}=600$, $Finish(3)_{F2}=900$, 1200, 1500, ...
$Finish(1)_{F3}=1000$, $Finish(2)_{F3}=2000$, ...

Fair Queuing (4)

- Suppose:
  - Flow 1 and 3 use 1000B byte packets, flow 2 uses 300B packets
  - What will fair queuing do?



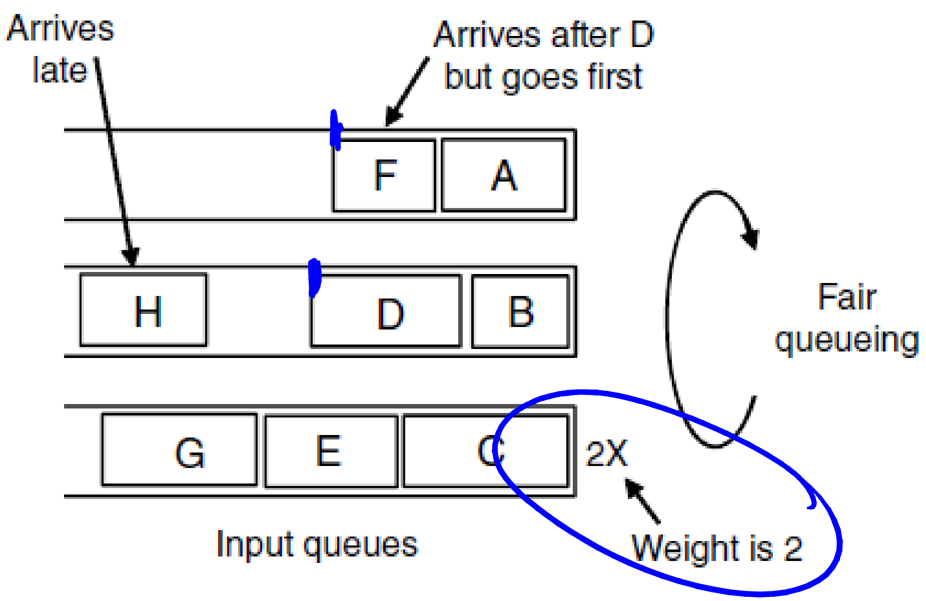
WFQ (Weighted Fair Queuing)

- WFQ is a useful generalization of Fair Queuing:
  - Assign a weight, $Weight_F$, to each flow
    - Higher weight gives more bandwidth, e.g., 2 is 2X bandwidth
  - Change computation of $Finish(j)_F$ to factor in $Weight_F$

$Arrive(j)_F$	= arrival time of j-th packet of flow F
$Length(j)_F$	= length of j-th packet of flow F
$Finish(j)_F$	$= \max(Arrive(j)_F, Finish(j-1)_F) + Length(j)_F / Weight_F$

WFQ Example

- An example you can work through …



	Packet	Arrival time	Length	Finish time	Output order
	A	0	8	8	1
	B	5	6	11	3
x2	C	5	10	10	2
	D	8	9	20	7
x2	E	8	8	14	4
	F	10	6	16	5
x2	G	11	10	19	6
	H	20	8	28	8

input

Using WFQ

- Lots of potential!
  - Can prioritize and protect flows
  - A powerful building block
- Not yet a complete solution
  - Need to determine flows (user? application? TCP connection?)
  - Difficult to implement at high speed for many concurrent flows
  - Need to assign weights to flows

END