Packet Scheduling in Data Centers

Lecture 17, Computer Networks (198:552)

RUTGERS
UNIVERSITY | NEW BRUNSWICK

Datacenter transport

- Goal: Complete flows quickly / meet deadlines

Short flows

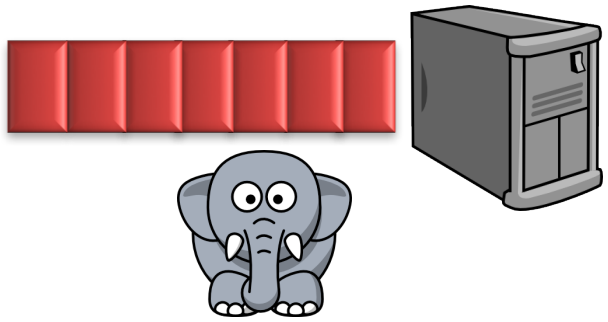
(e.g., query, coordination) → **Low Latency**

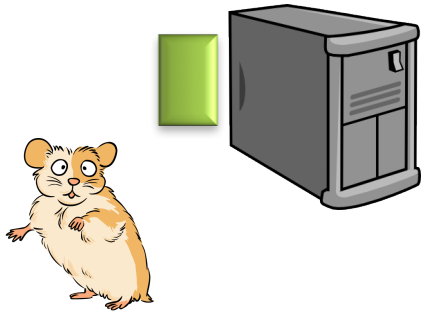
Large flows

(e.g., data update, backup) → **High Throughput**

Low latency cong control (ex: DCTCP)

- Keep network queues small (at high throughput)



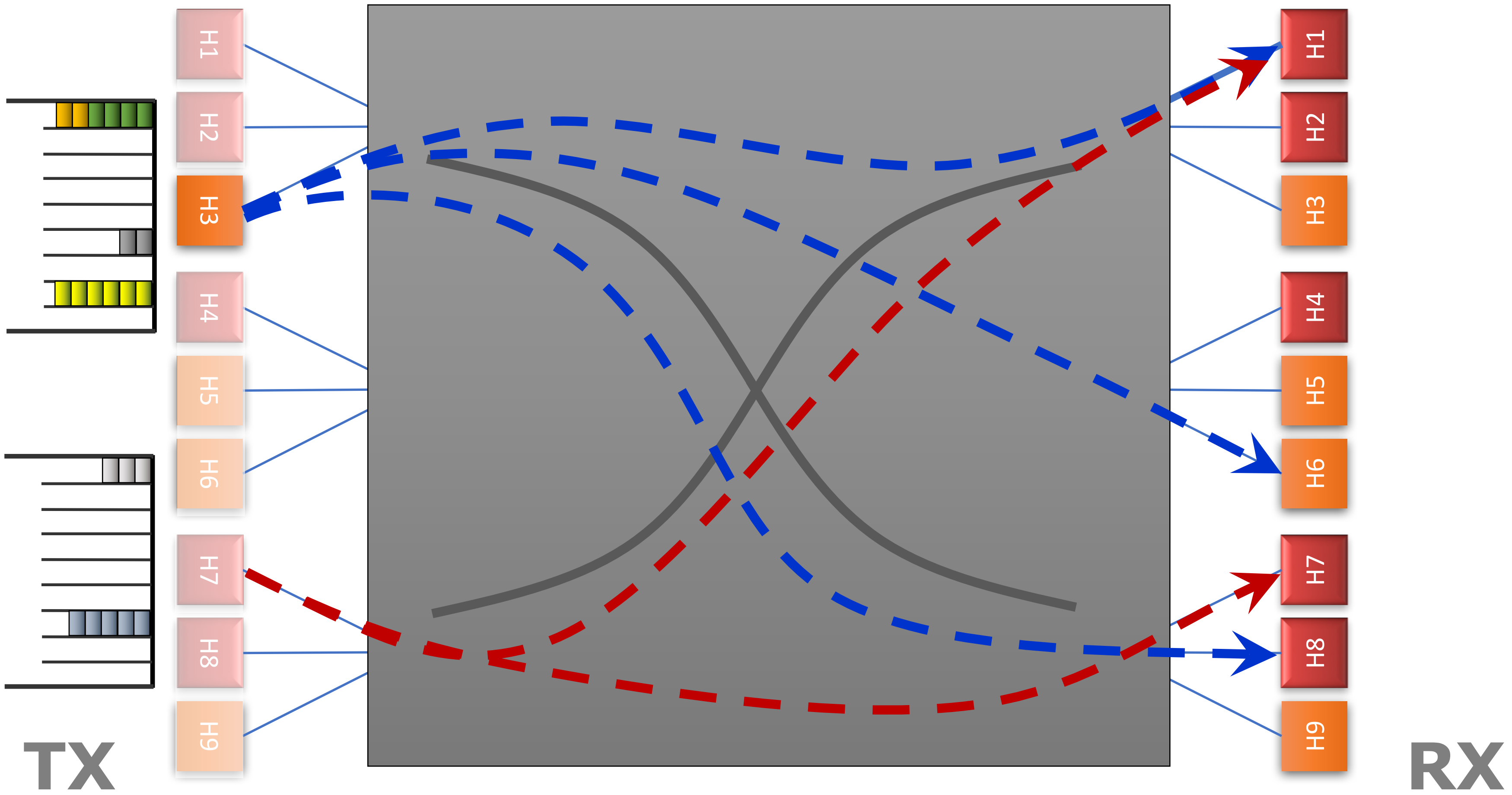


Want: implicitly prioritize mice
HOL blocking if long queues

Can we do better?

The opportunity

- Many DC apps/platforms know flow size or deadlines in advance
  - Key/value stores
  - Data processing
  - Web search

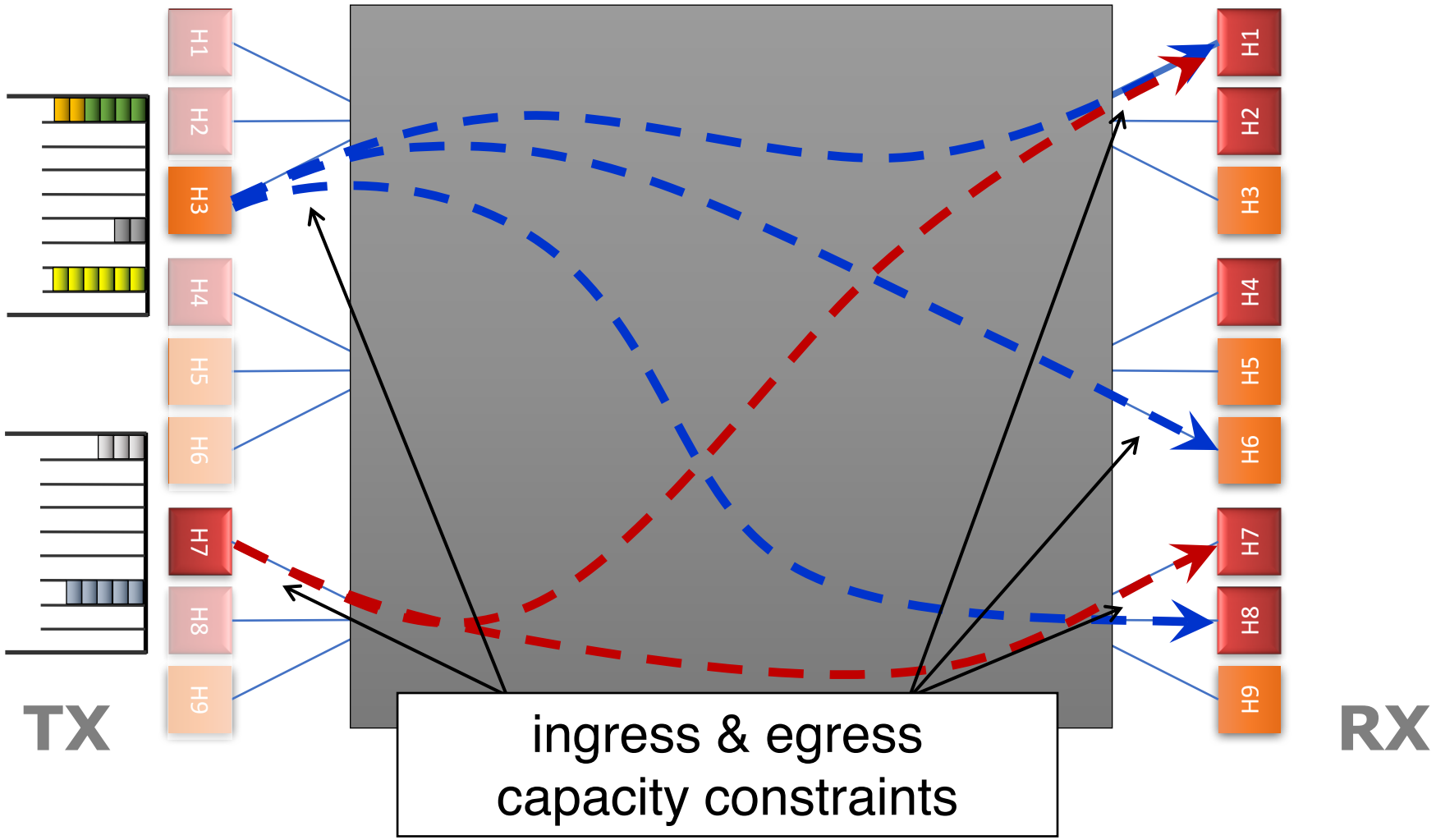


H1
H2
H3
H4
H5
H6
H7
H8
H9
TX
H1
H2
H3
H4
H5
H6
H7
H8
H9
RX

DC transport =
Flow scheduling on
giant switch

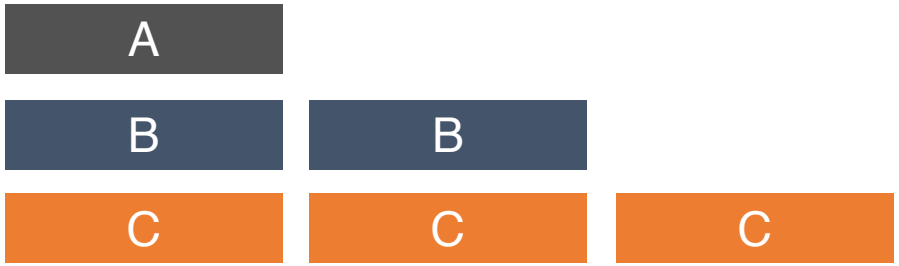
Objective?

- Minimize avg FCT
- Minimize missed deadlines



Example: Minimize average FCT

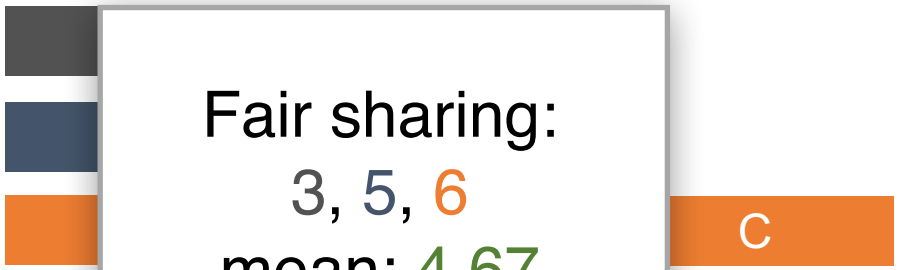
	Size
Flow A	1
Flow B	2
Flow C	3



arrive at the same time

share the same bottleneck link

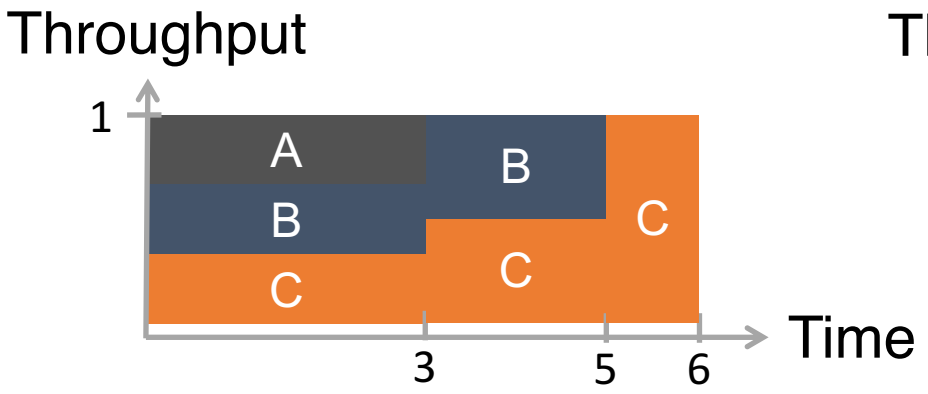
Example: Minimize average FCT

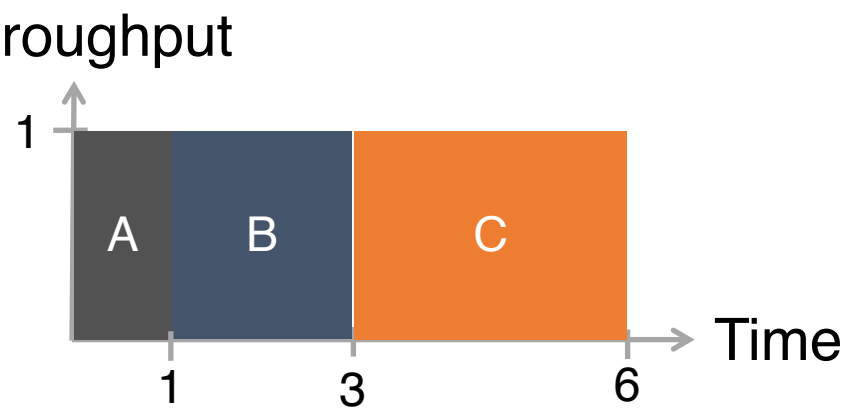


Fair sharing:
3, 5, 6
mean: 4.67

C

Shortest flow first:
1, 3, 6
mean: 3.33

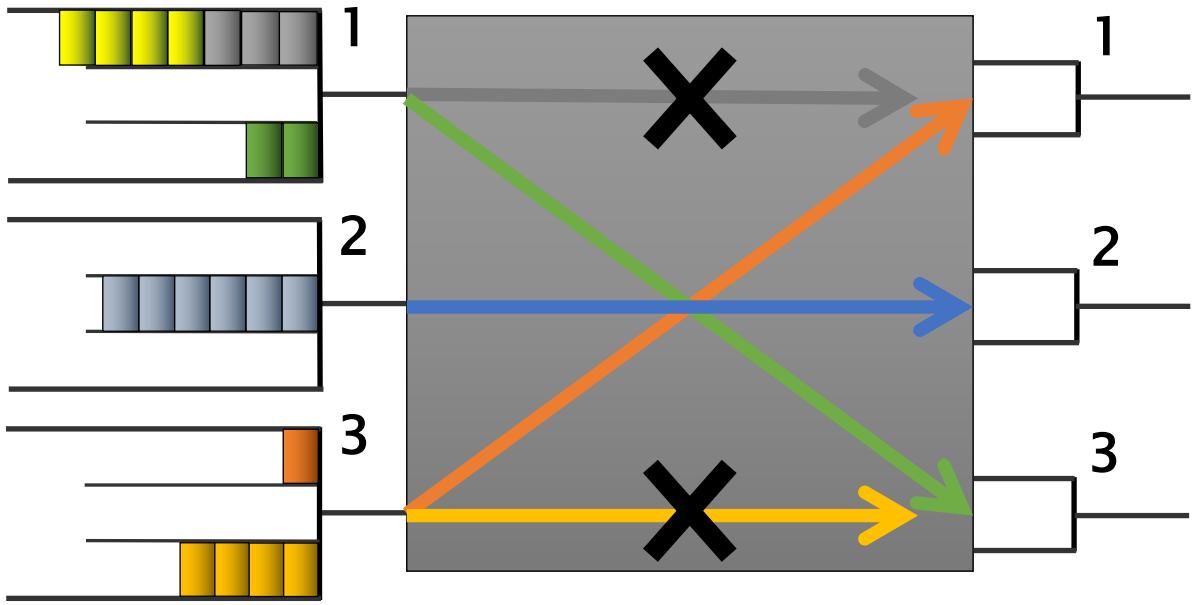




Optimal flow scheduling for avg FCT

NP-hard for multi-link network [Bar-Noy et al.]

- Shortest Flow First: **2-approximation**



How can we schedule flows based on flow criticality in a distributed way?

Some transmission order

pFabric

Mohammad Alizadeh et al., SIGCOMM'13

pFabric in 1 slide

Packets carry a single priority

- e.g., priority = remaining flow size

pFabric Switches

- Send highest priority / drop lowest priority packets
- Very small buffers (20-30KB for 10Gbps fabric)

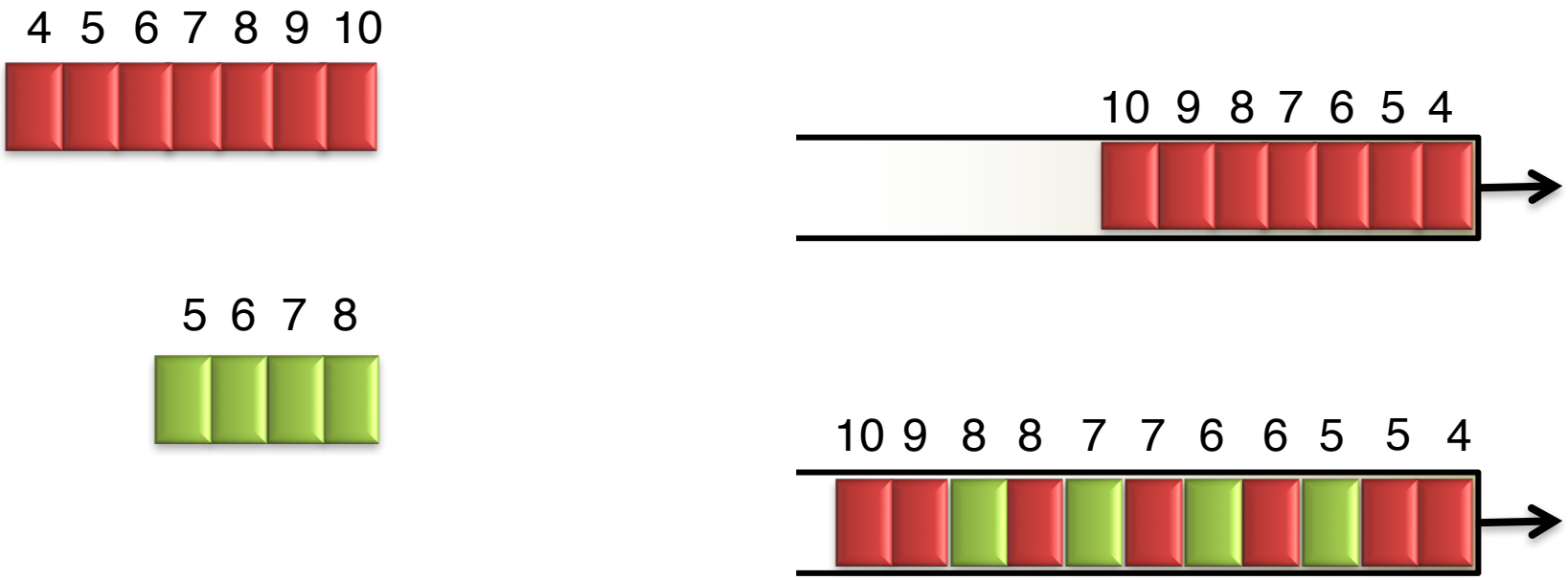
pFabric Hosts

- Send/retransmit aggressively
- Minimal rate control: just prevent congestion collapse

Main Idea:
Decouple scheduling from rate control

Starvation prevention

- Use remaining flow size as priority: What happens?



- Transmit *earliest packet* of flow with the highest priority packet

pFabric switch

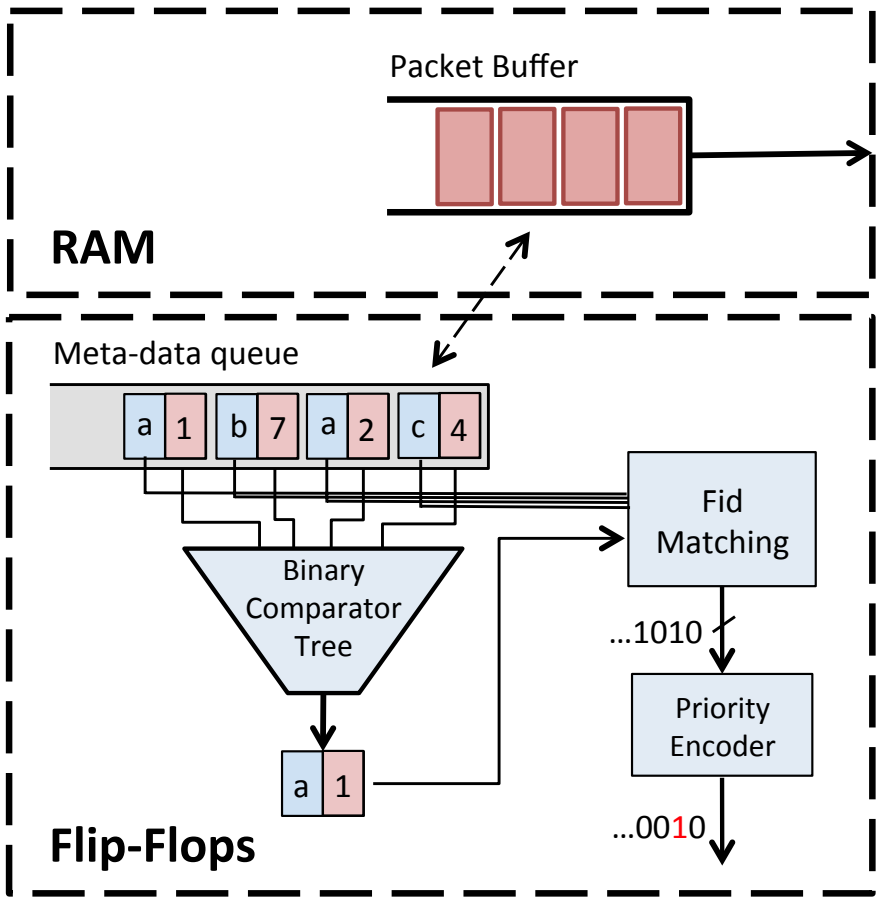
Boils down to a sort

- Essentially unlimited priorities
- Thought to be difficult in hardware

Existing switching only support 4-16 priorities

pFabric queues very small

- 51.2ns to find min/max of ~600 numbers
- Binary comparator tree: 10 clock cycles
- Current ASICs: clock ~ 1ns



pFabric rate control

Minimal version of TCP algorithm

1. Start at line-rate
   - Initial window larger than BDP
2. No retransmission timeout estimation
   - Fixed RTO at small multiple of round-trip time
3. Reduce window size upon packet drops
   - Window increase same as TCP (slow start, congestion avoidance, …)
4. After multiple consecutive timeouts, enter “probe mode”
   - Probe mode sends min. size packets until first ACK

What about queue buildup?

Why window control?

Why does pFabric work?

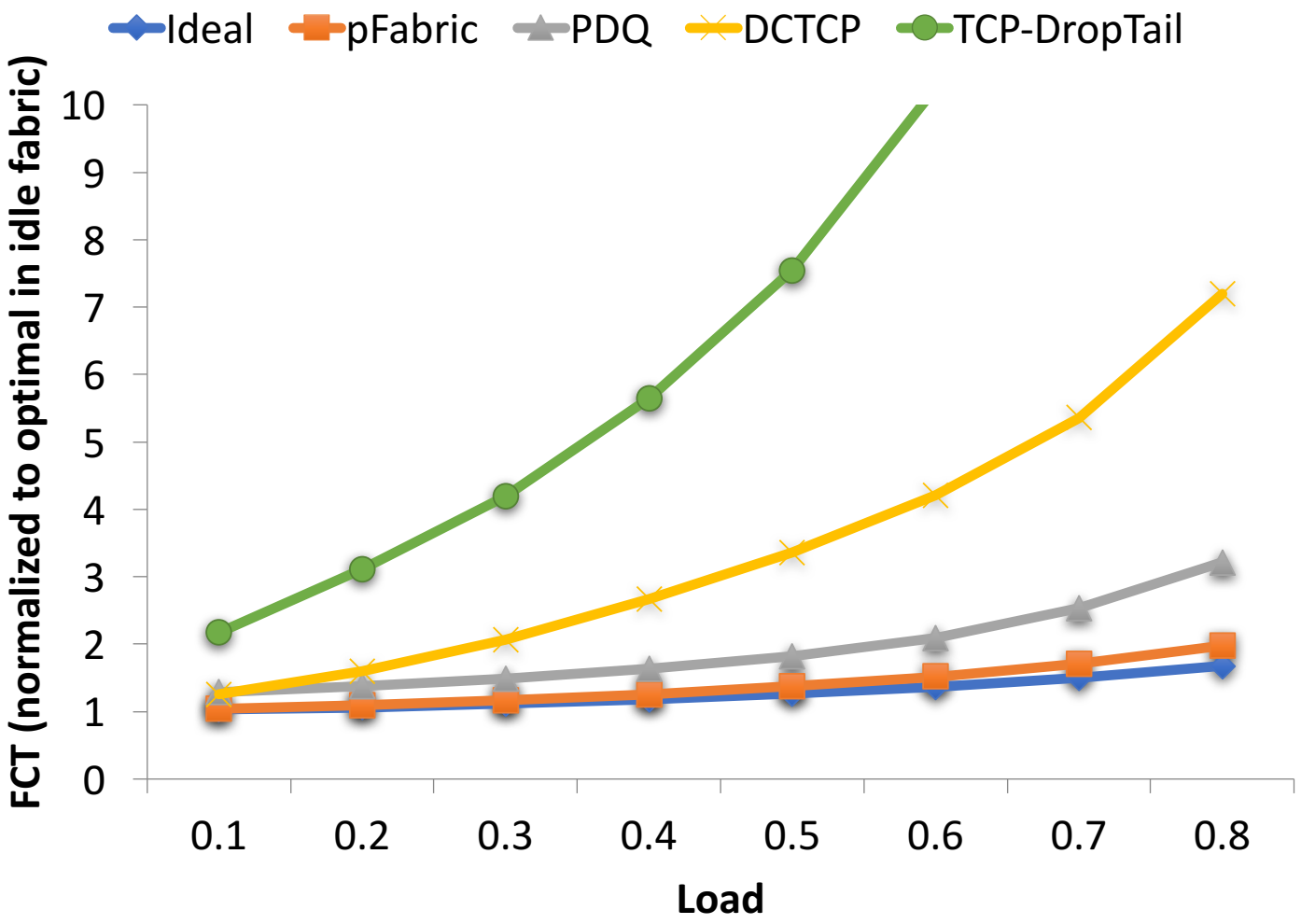
Key invariant:

At any instant, have the highest priority packet (according to ideal algorithm) available at the switch.

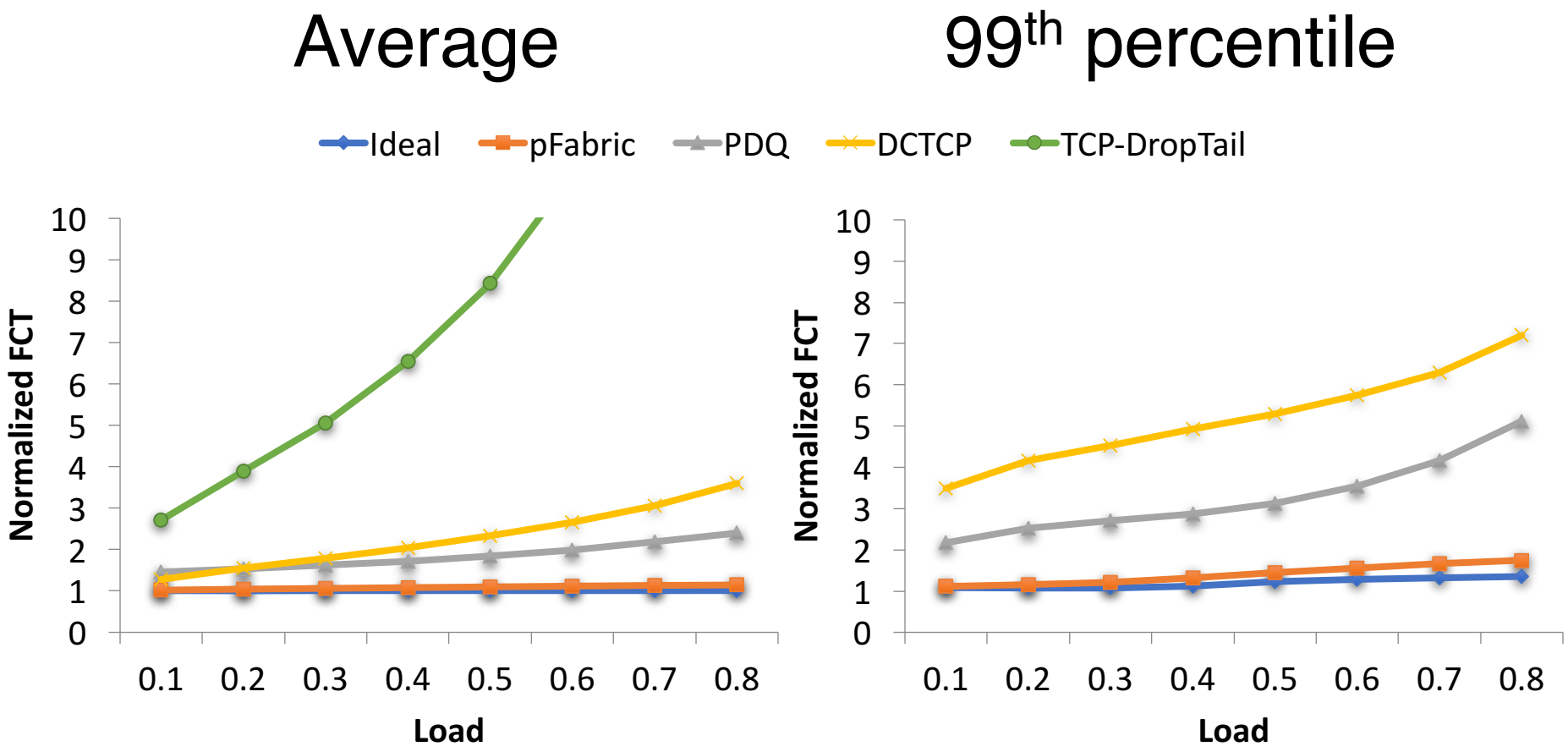
- Priority scheduling
  - High priority packets traverse fabric as quickly as possible

- What about dropped packets?
  - Lowest priority → not needed till all other packets depart
  - Buffer > BDP → enough time (> RTT) to retransmit

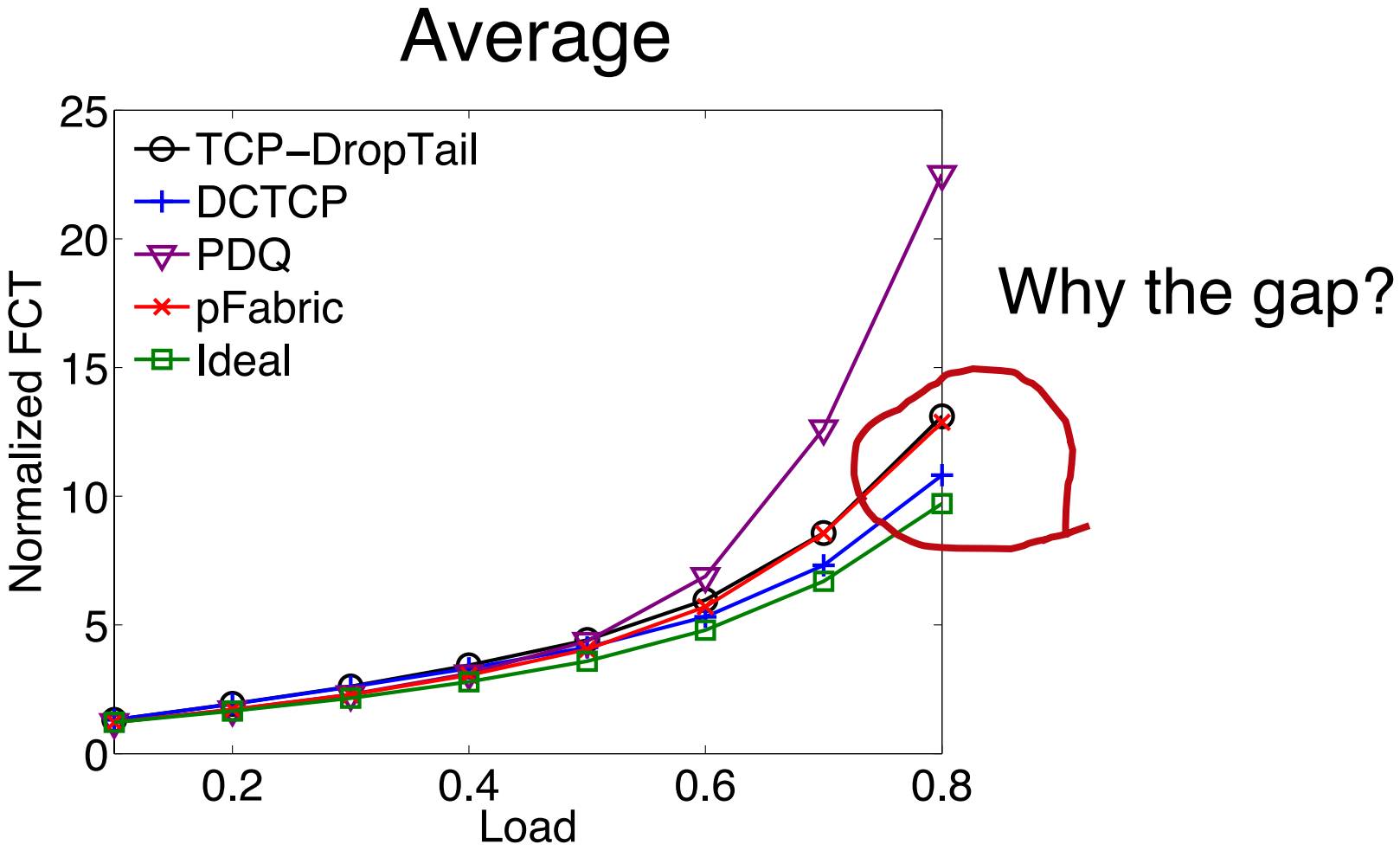
Overall mean FCT: Web search workload



Mice FCT (<100KB): Web search workload



Elephant FCT (>10MB): Data mining workload



Why the gap?

Discussion

- Priority-based scheduling: pros and cons
  - Gaming?
  - Starvation?

- Implementation using a small number of priority queues?
  - How to set priority thresholds?

- When is the “big switch” abstraction appropriate?

PIAS:
Information-agnostic scheduling

Wei Bai et al., NSDI'15

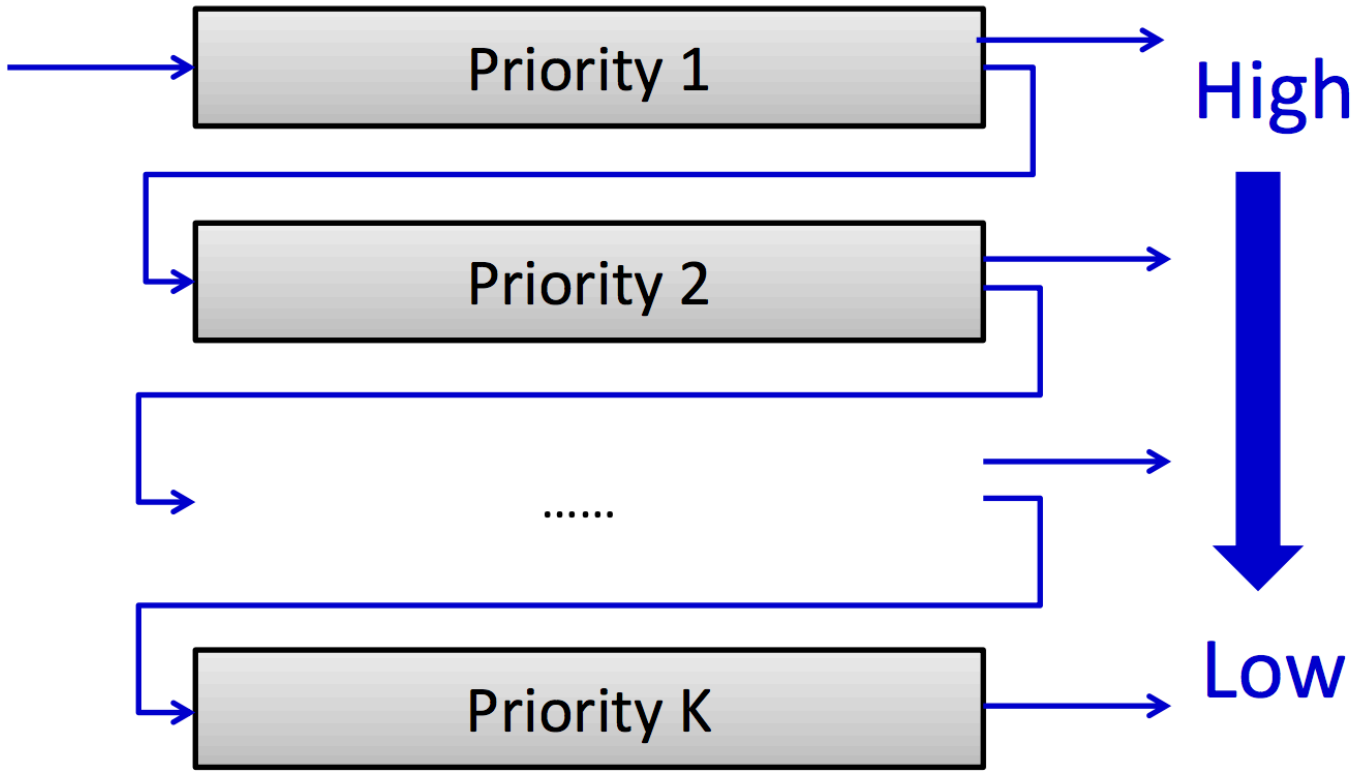
State of the art for packet/flow scheduling

- pFabric, PDQ, PASE
- Assume
  - ~~Prior knowledge of flow size information~~
  - **To approximate ideal preemptive shortest job first (SJF)**
  - ~~With customized network elements~~

Can we minimize FCT without flow size information with commodity components?

Multi-level feedback queue (MLFQ)

- Emulate shortest job first (SJF) **without** job size information



Realization with commodity components

- Don’t keep per-flow sizes and state in switch
  - Endpoint maintains per-flow state
  - … sets the priority as a packet tag
- Map packets to queues using thresholds on packet priorities
- What about bad thresholds? Use ECN to keep queues short
  - Use DCTCP at endpoints to react smoothly to ECN

Discussion

- What happens if a flow transmits with a bursty traffic pattern?
  - Transmit for x seconds, wait y seconds, transmit again for x

- What if a flow transmits just enough to always stay in the highest priority queue?

MLFQ rules (Arpaci-Dusseau)

- **Rule 1:** If Priority(A) $>$ Priority(B), A runs (B doesn't).
- **Rule 2:** If Priority(A) $=$ Priority(B), A & B run in round-robin fashion using the time slice (quantum length) of the given queue.
- **Rule 3:** When a job enters the system, it is placed at the highest priority (the topmost queue).
- **Rule 4:** Once a job uses up its time allotment at a given level (regardless of how many times it has given up the CPU), its priority is reduced (i.e., it moves down one queue).
- **Rule 5:** After some time period S, move all the jobs in the system to the topmost queue.

Acknowledgment

- Slides heavily adapted from material by Mohammad Alizadeh, Chi-Yao Hong, and Wei Bai