THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

Networked Systems for Big Data and Clouds

Kai Chen
CSE Department, HKUST

System Platform Implemented

Electrical Packet Switch, 1G (x10)

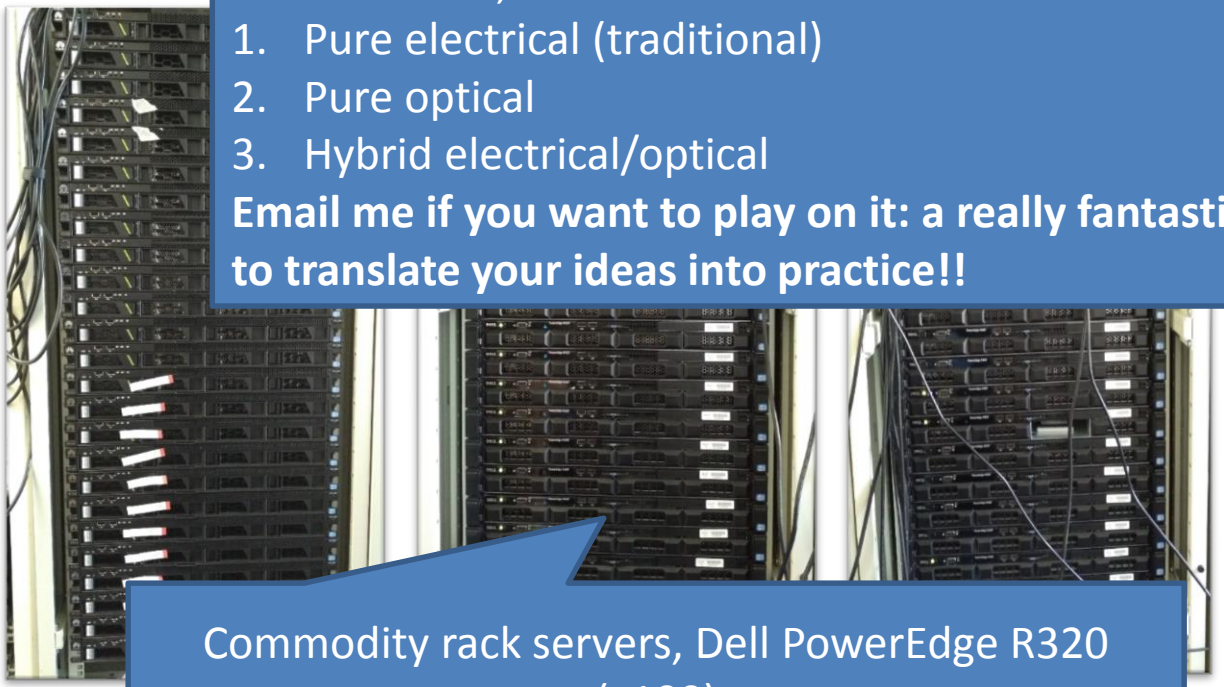
Electrical Packet Switch, 10G (x3)

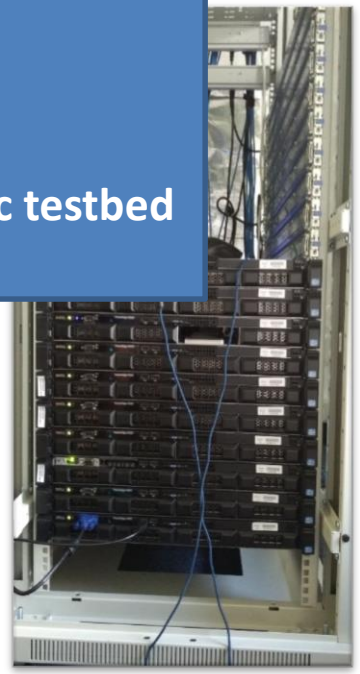
Optical Circuit Switch (x5)

All connected, can form three architectures:

1. Pure electrical (traditional)
2. Pure optical
3. Hybrid electrical/optical

Email me if you want to play on it: a really fantastic testbed to translate your ideas into practice!!





Commodity rack servers, Dell PowerEdge R320 (x100)

Research: A Bottom-up Approach

- **Infrastructure/Architecture**
  - OSA, the *first-ever* all optical datacenter architecture (ToN'14, NSDI'12)
  - BigSwitch, a massive-port (6336 ports) datacenter switch
- **Network Layer**
  - XPath, intra-datacenter routing control (NSDI'15, ToN'15)
  - Amoeba, inter-datacenter data transfers (EuroSys'15)
- **Transport Layer**
  - PIAS, practical flow scheduling (NSDI'15, HotNets'14)
  - CODA, centralized/decentralized coflow scheduling
- **Computing Platform for Applications**
  - BigComputing, network-enabled large scale distributed computing platforms for big data analytics and machine learning

Goal: simple, practical, readily-implementable solutions for real applications!

PIAS: Practical Information Agnostic Flow Scheduling

[USENIX NSDI'15]

http://sing.cse.ust.hk/projects/PIAS

- Motivation: cloud apps desire low latency for short flows/messages

- Design goal: minimize Flow Completion Time (FCT)

Not feasible for some real applications

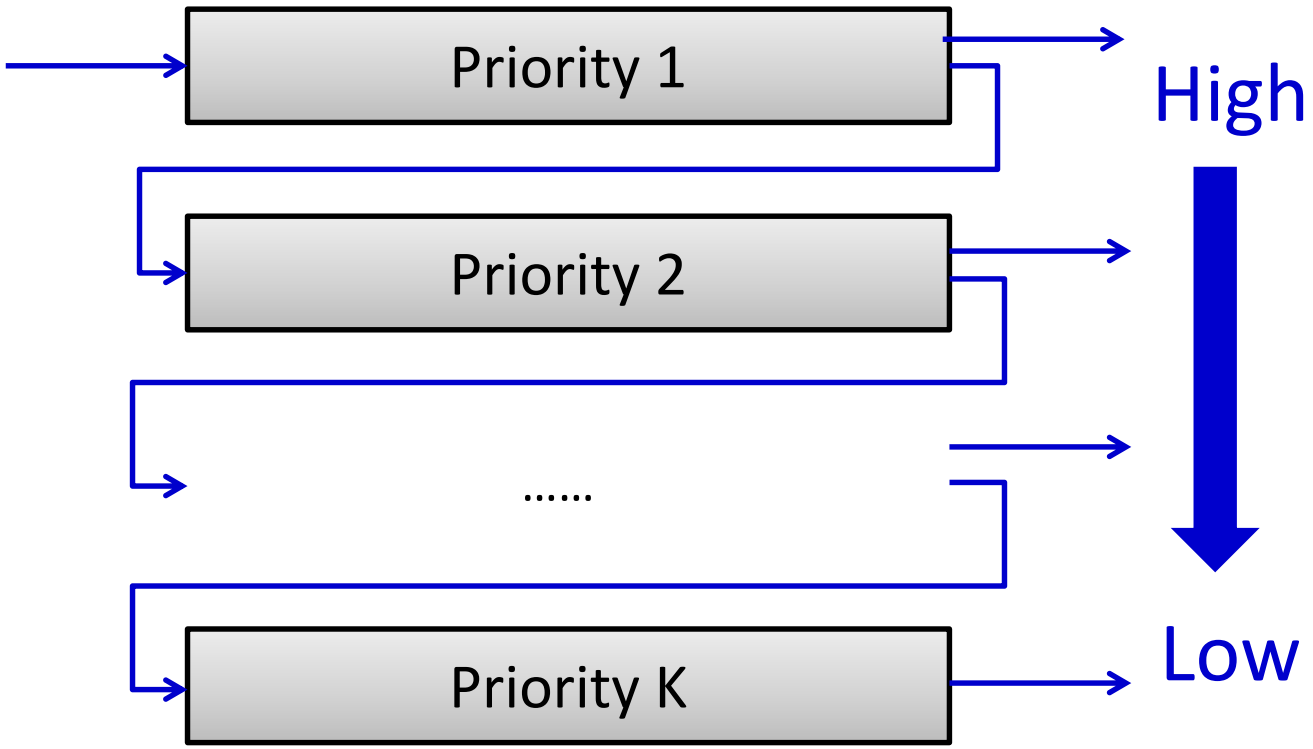
- Existing solutions: almost all ~~assume prior knowledge of flow size~~ information to approximate ideal Shortest Job First (SJF) to minimize average FCT with ~~customized network elements~~.

Hard to implement in practice

- **PIAS** makes no assumption on prior knowledge of flow size, while still emulating SJF to minimize average FCT with existing commodity switches.

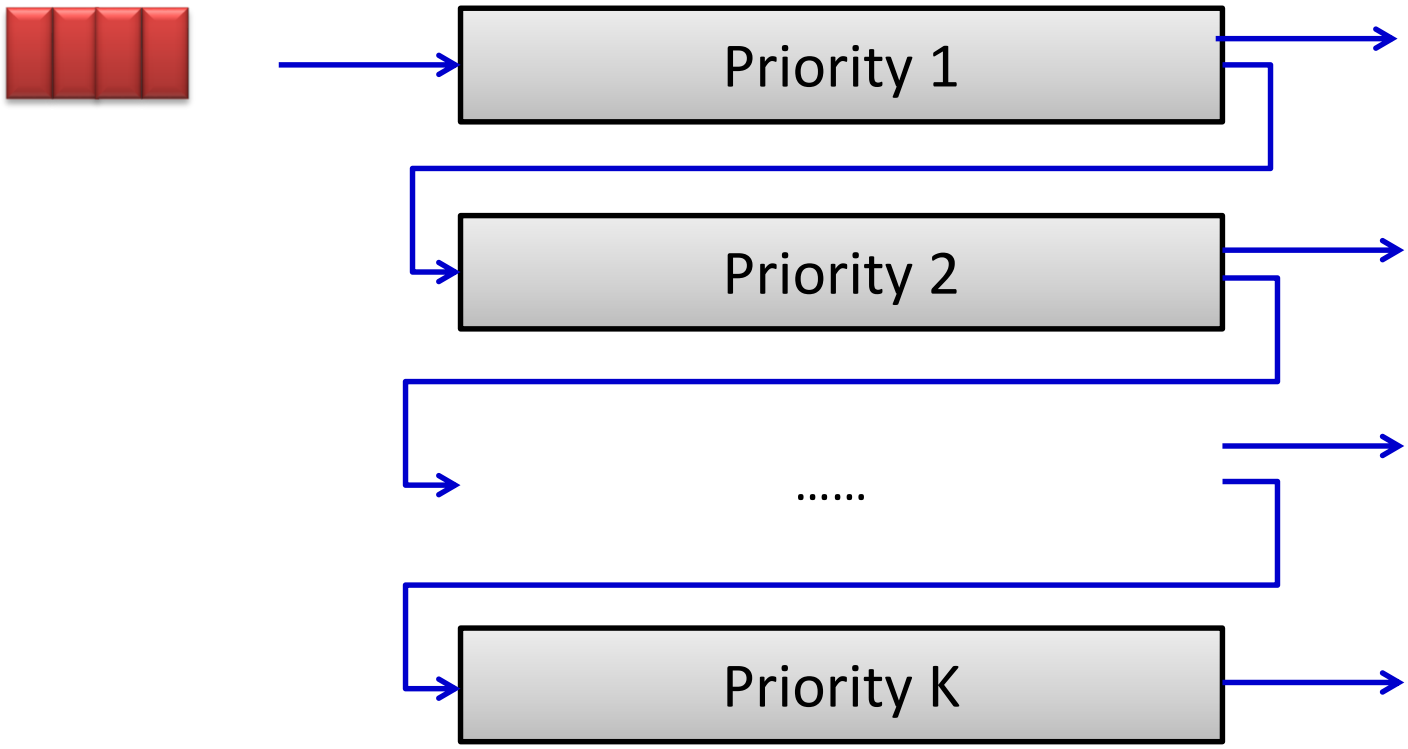
PIAS: Enabling Technique

- Today's commodity switching chips already support priority queues

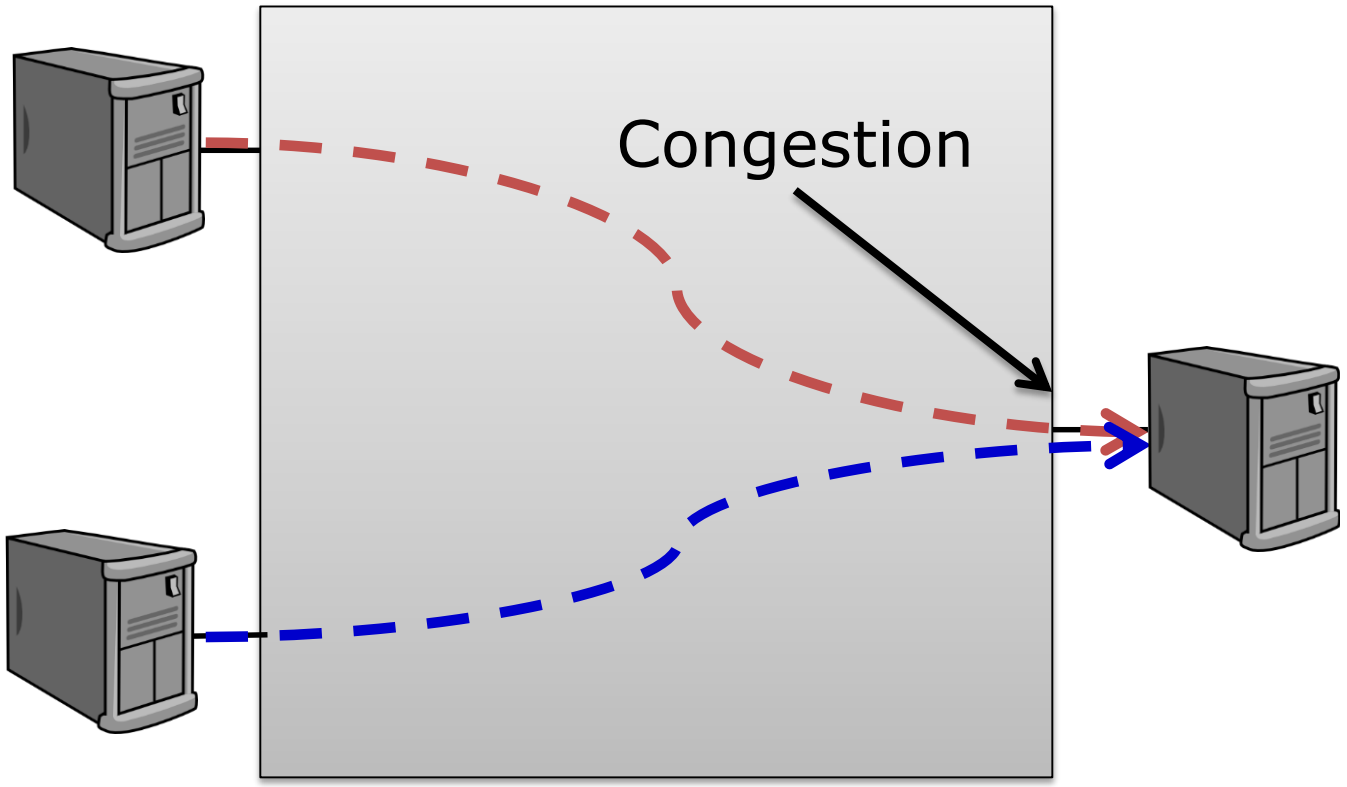


PIAS: Core Idea

- PIAS leverages the priority queues to perform Multi-Level Feedback Queue (MLFQ) scheduling to emulate SJF

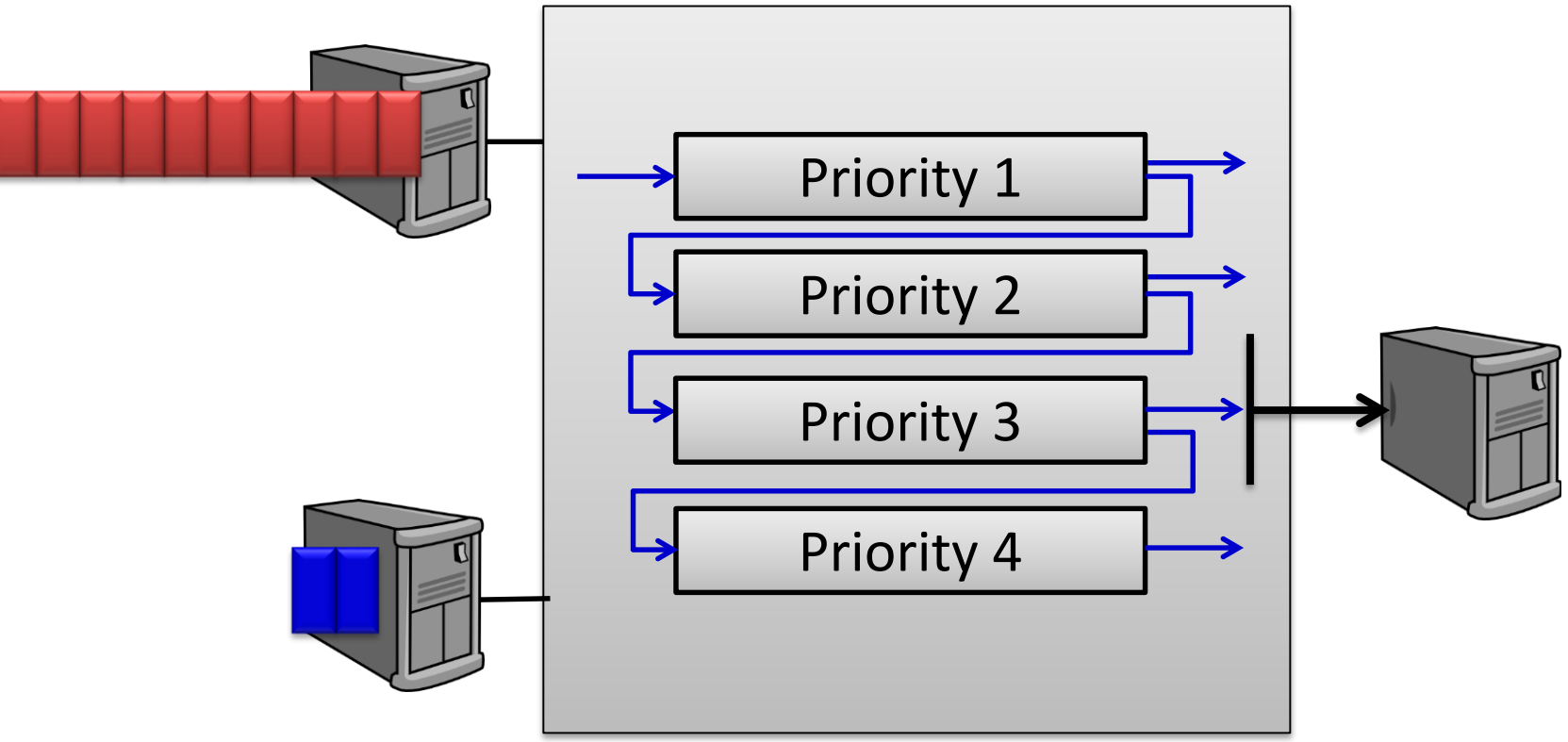


PIAS in One Animation



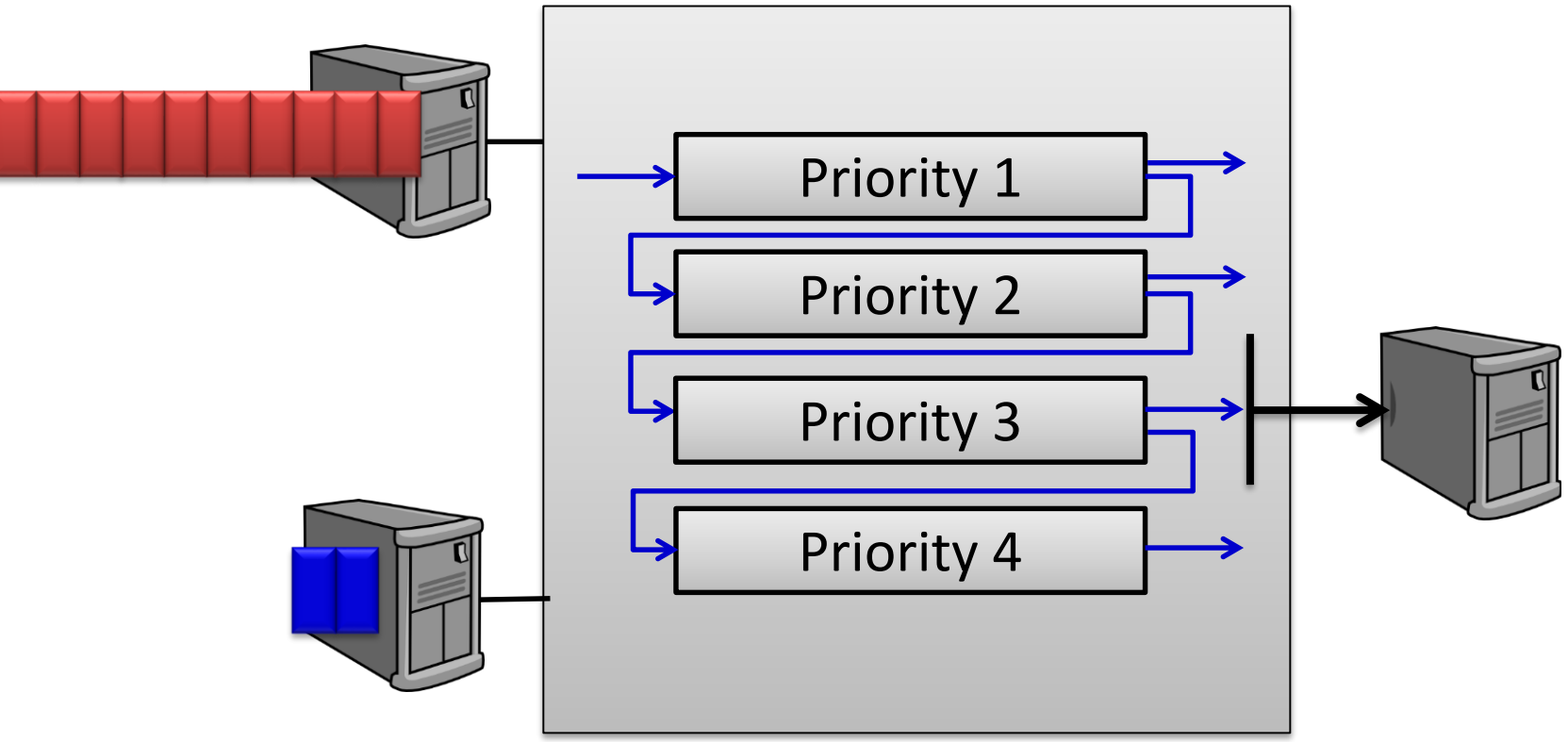
PIAS in One Animation

Flow 1 with 10 packets and flow 2 with 2 packets arrive



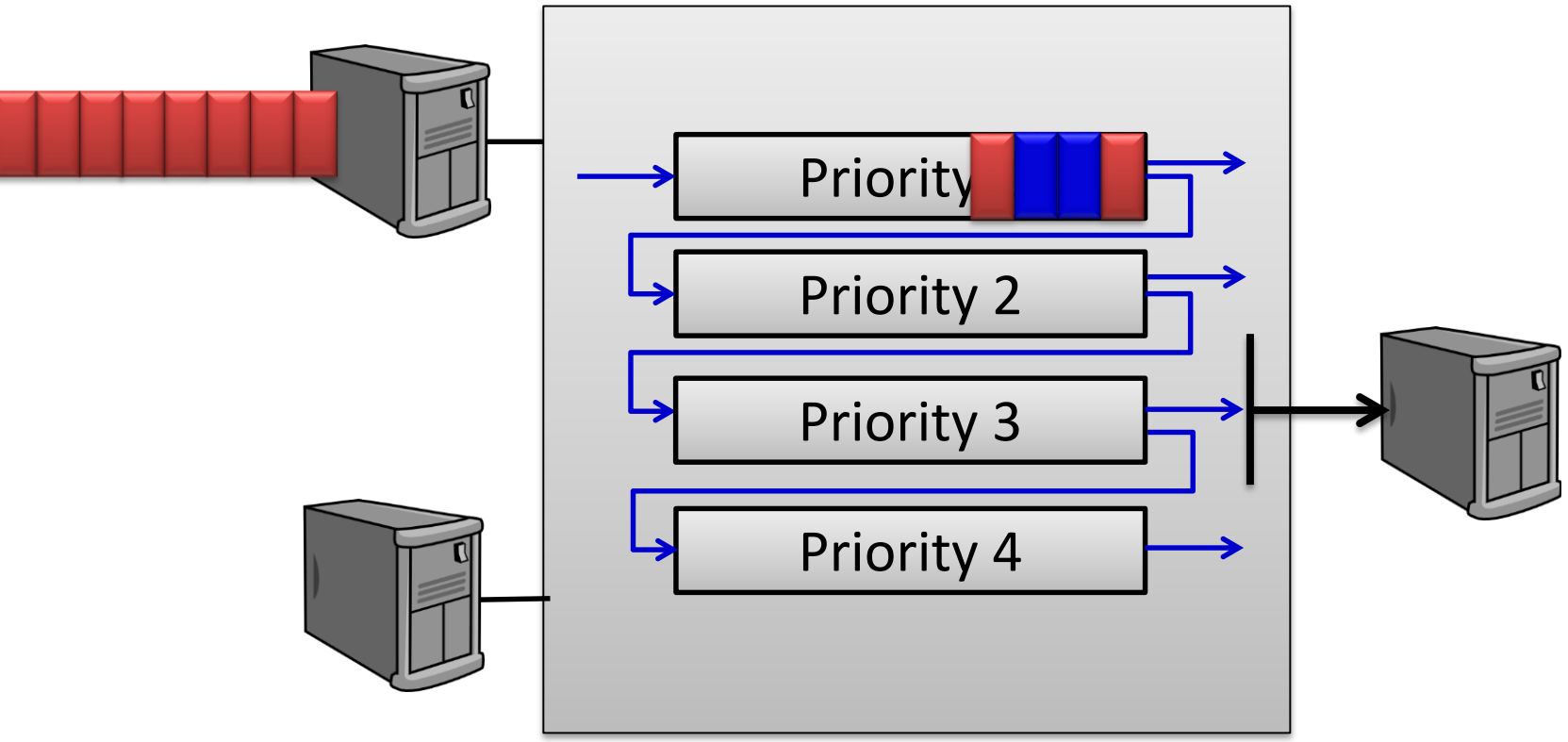
PIAS in One Animation

Flow 1 and 2 transmit simultaneously



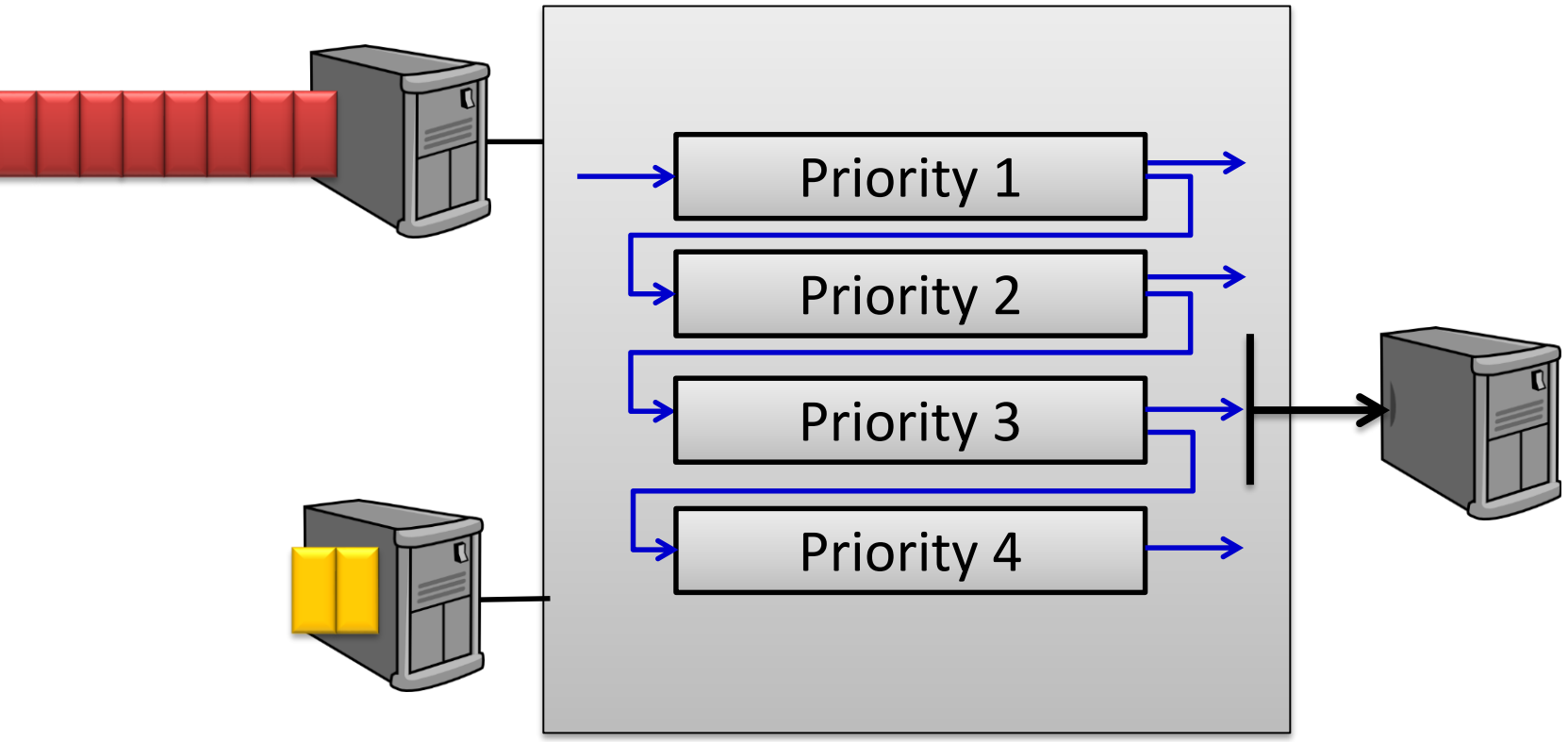
PIAS in One Animation

Flow 2 finishes while flow 1 is demoted to priority 2



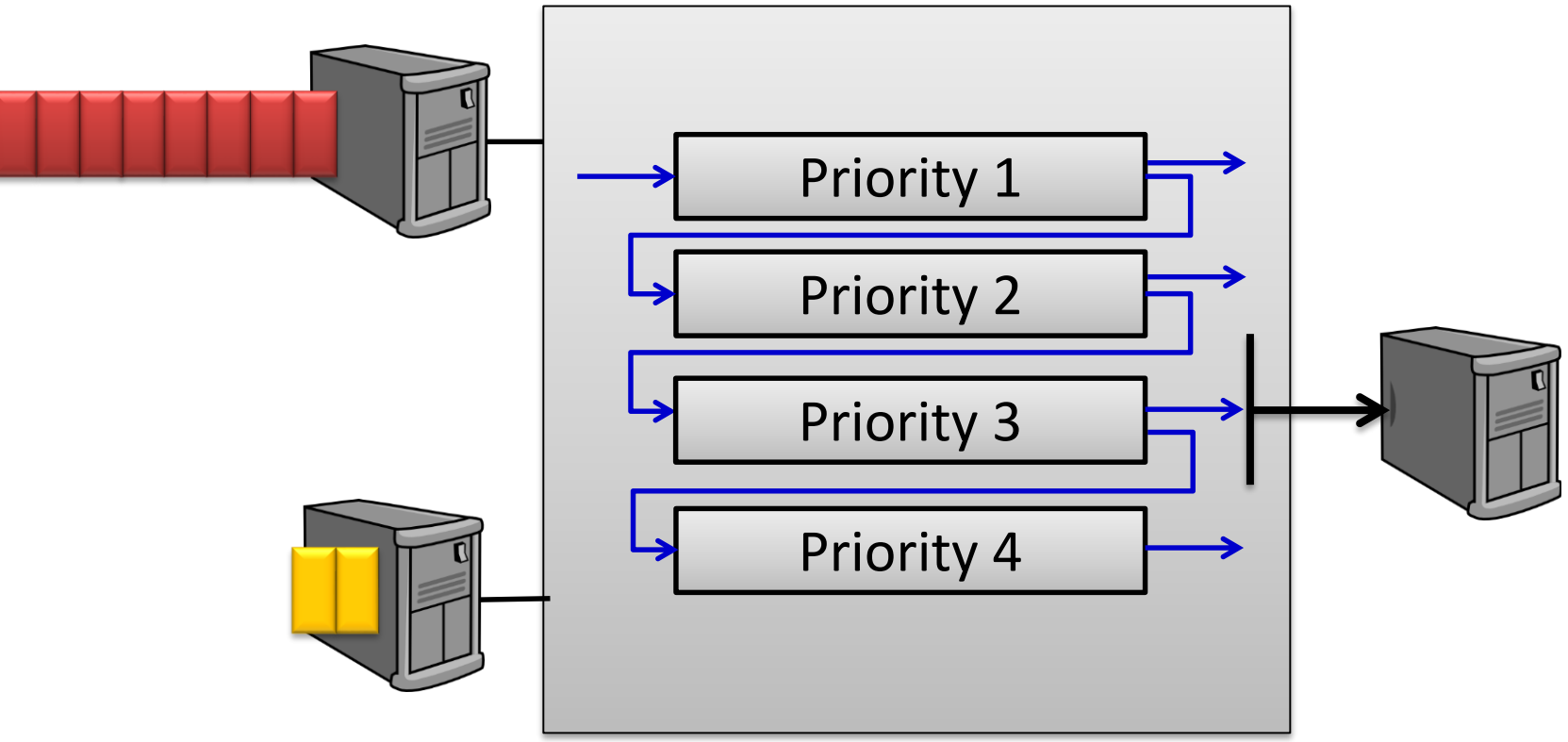
PIAS in One Animation

Flow 3 with 2 packets arrives



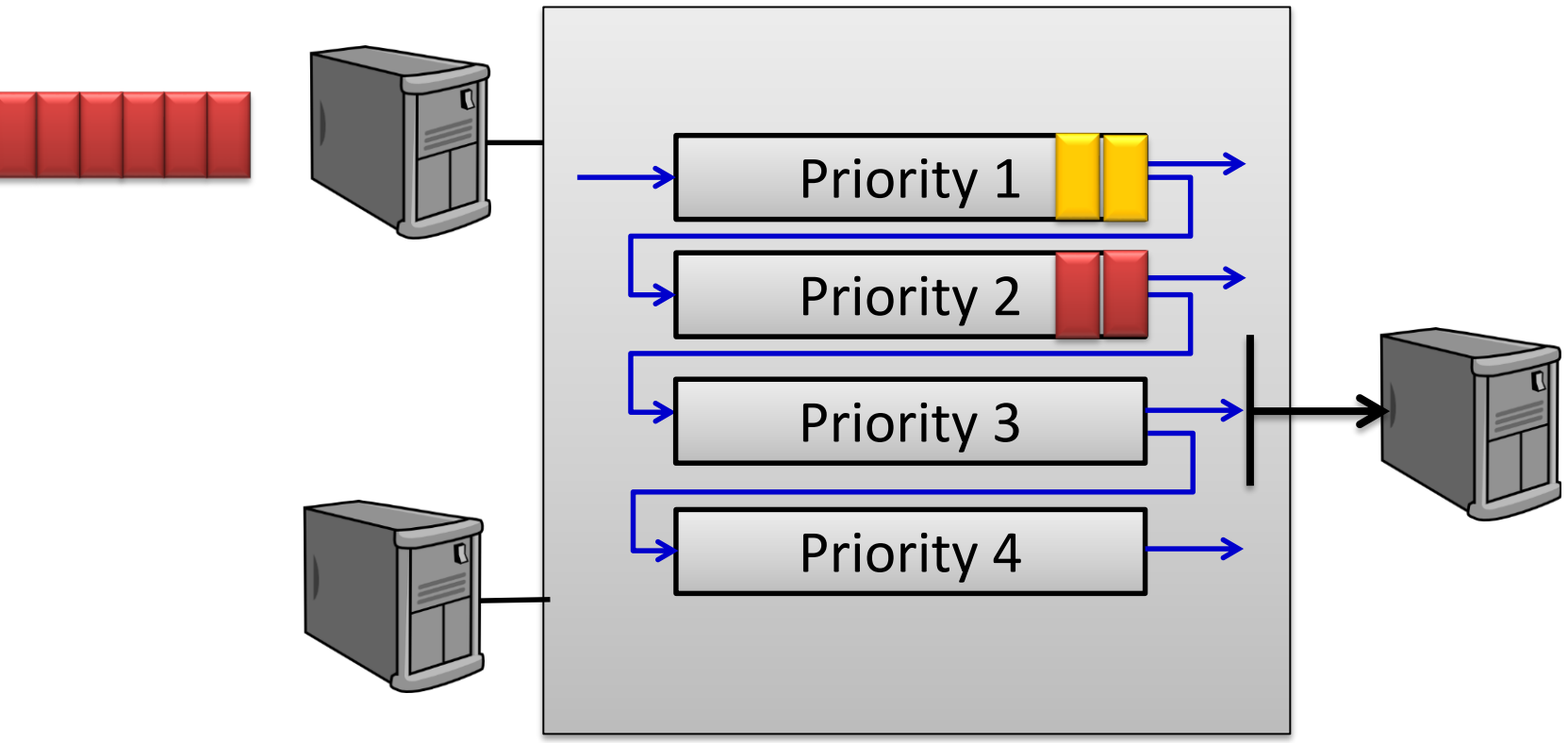
PIAS in One Animation

Flow 3 and 1 transmit simultaneously



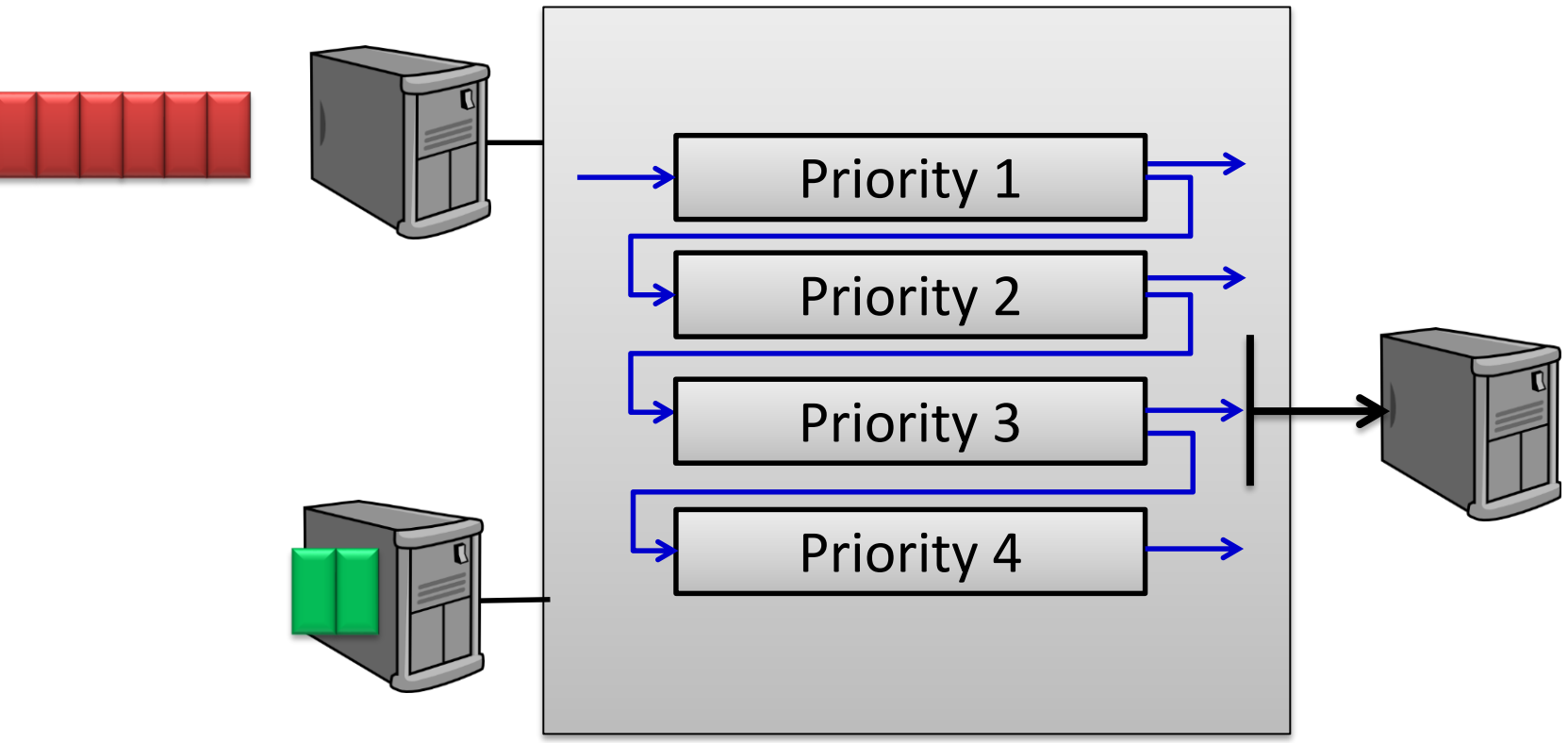
PIAS in One Animation

Flow 3 finishes while flow 1 is demoted to priority 3



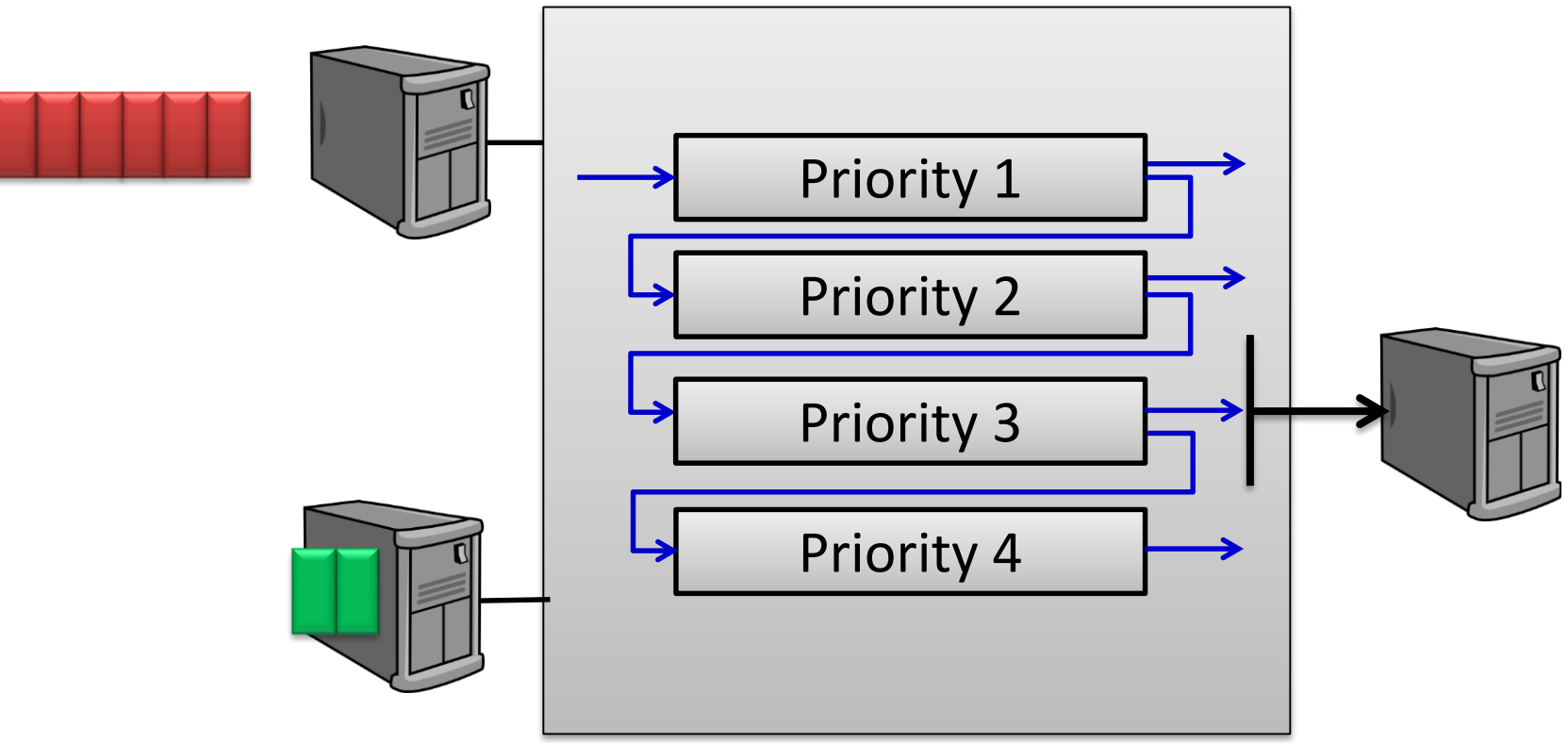
PIAS in One Animation

Flow 4 with 2 packets arrives



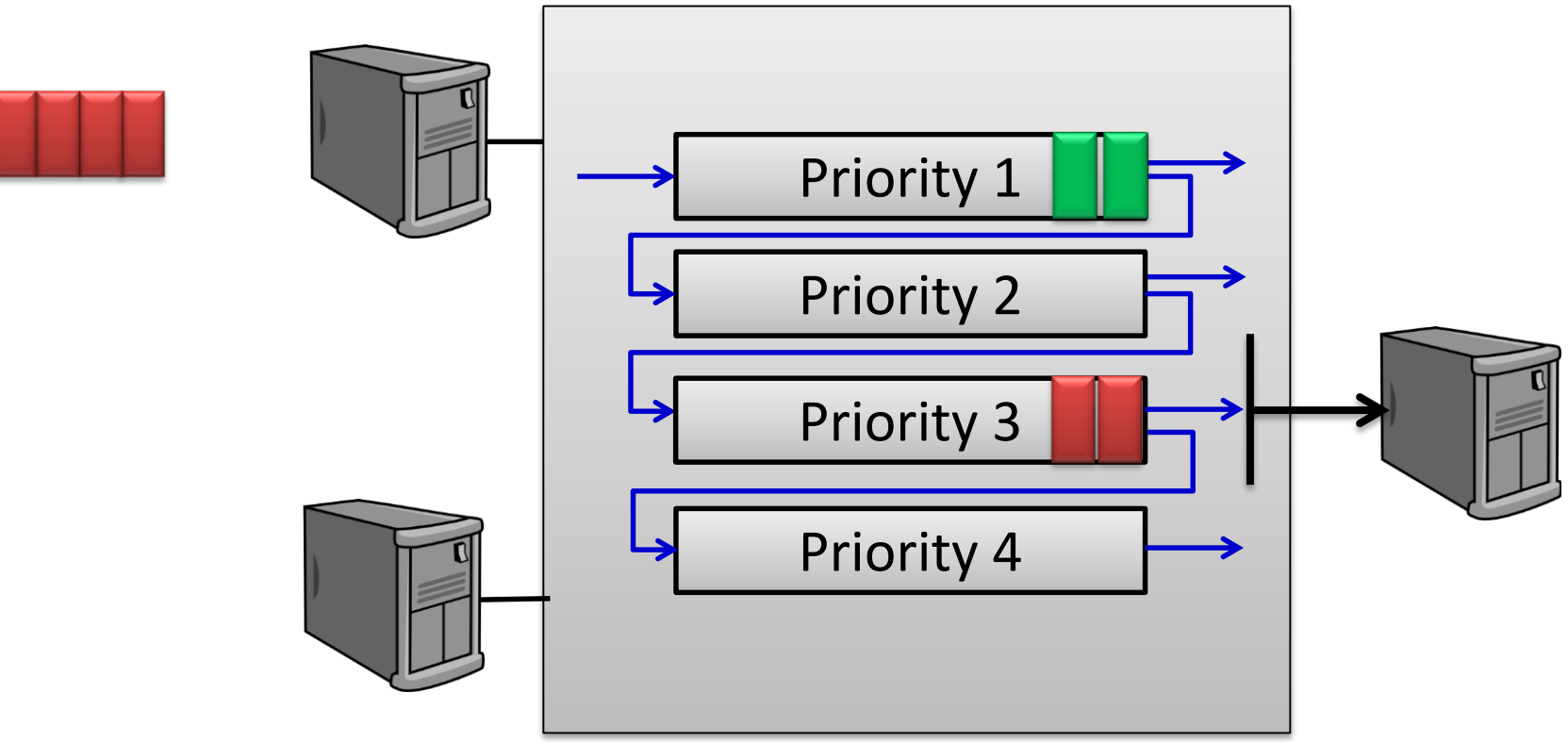
PIAS in One Animation

Flow 4 and 1 transmit simultaneously



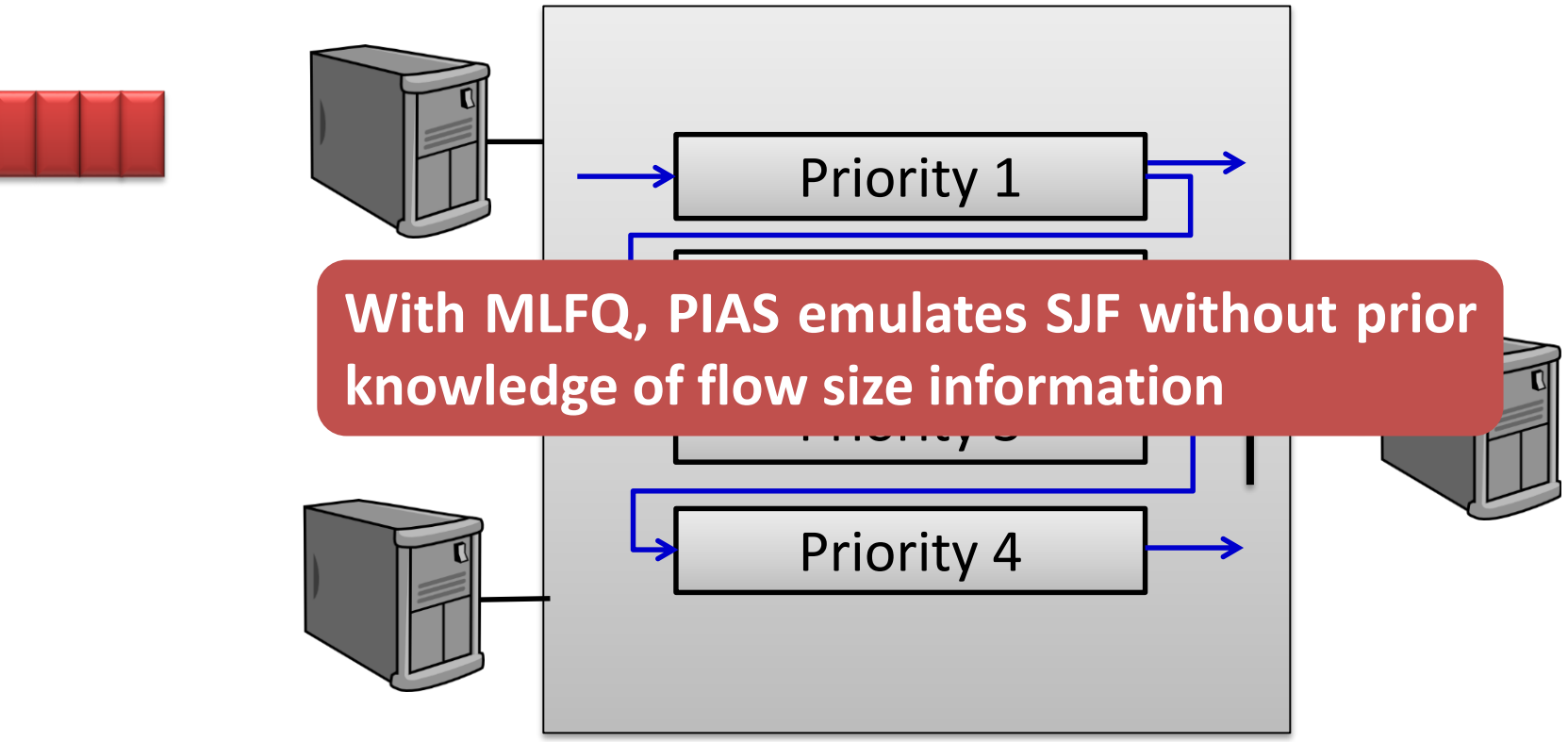
PIAS in One Animation

Flow 4 finishes while flow 1 is demoted to priority 4



PIAS in One Animation

Eventually, flow 1 finishes in priority 4



Thanks, Q&A