ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Applications to Transport, Network QoS, and In-Network Computing

Khanh Nam Chu

ONE LOVE. ONE FUTURE.

Contents

I. Transport and Network QoS

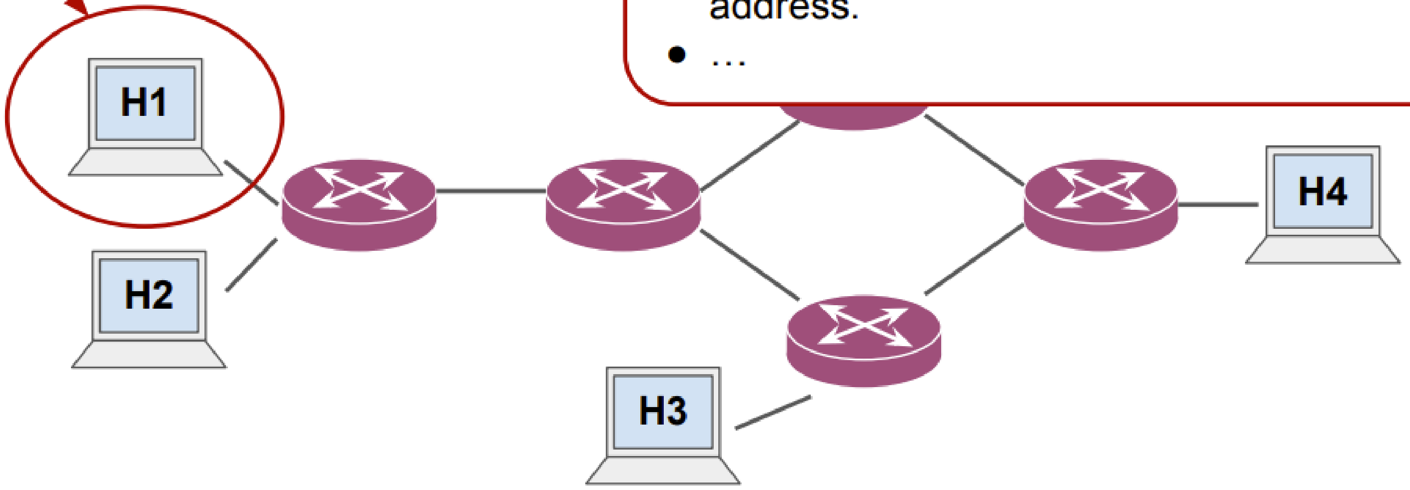
1. Networks are shared infrastructure

Traffic from multiple "flows" share

- The link to the network
- The memory and computational resources of the host

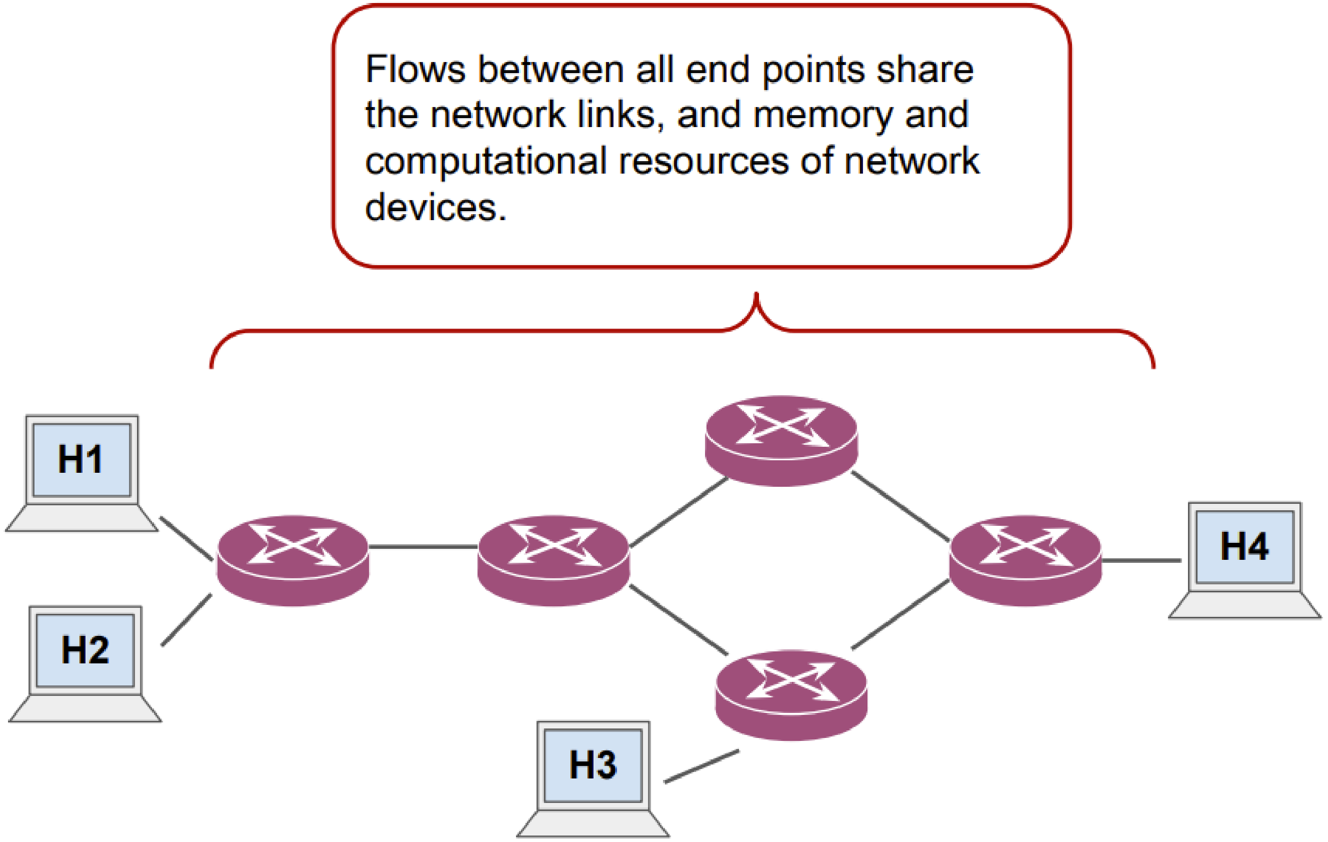
You can think of a flow in the broad sense of the term – a set of packets that are treated together as a group for network processing purposes:

- A TCP flow
- Packets originating from the same VM
- All the DNS packets from the same IP address.
- …



I. Transport and Network QoS

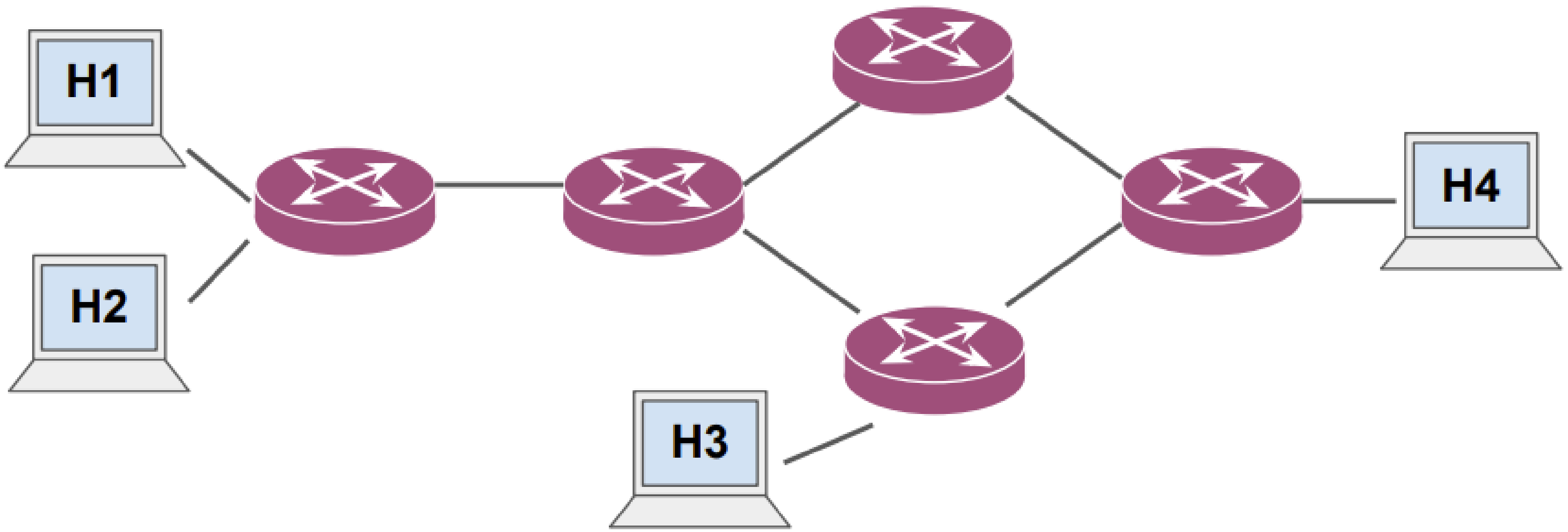
1. Networks are shared infrastructure



I. Transport and Network QoS

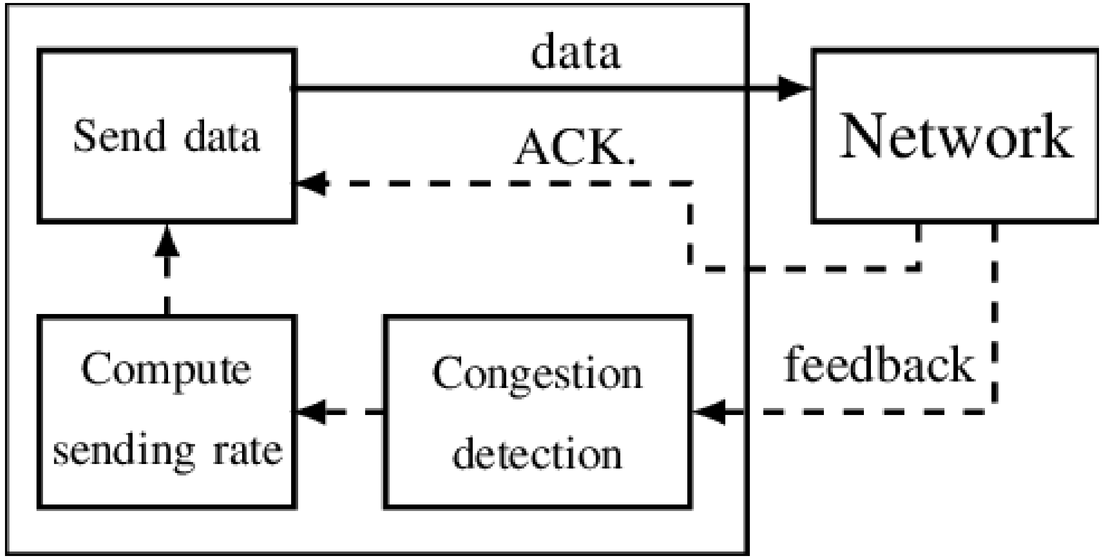
1. Networks are shared infrastructure

- IP only provides best-effort packet delivery.
- There are other mechanisms to control/customize how different flows share network resources: end-to-end congestion control, packet scheduling, active queue management,…

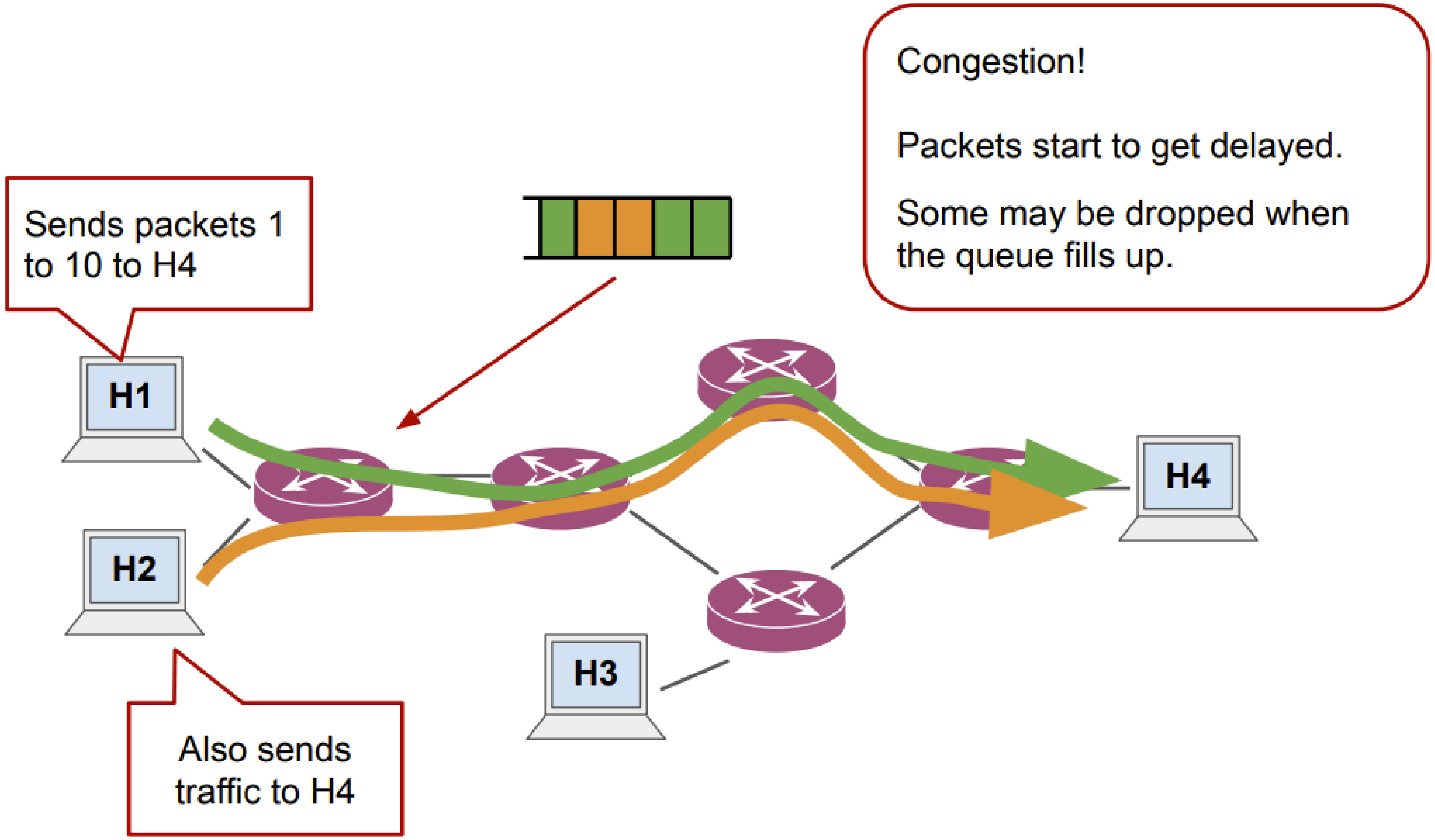


2. End-to-end Congestion Control

- For every flow, the sender sends some packets out.
- Use some signals to detect congestion in the network: packets getting lost, packets taking longer to get to the receiver, network/receiver telling you it is congested,…
- Adjust sending rate accordingly.

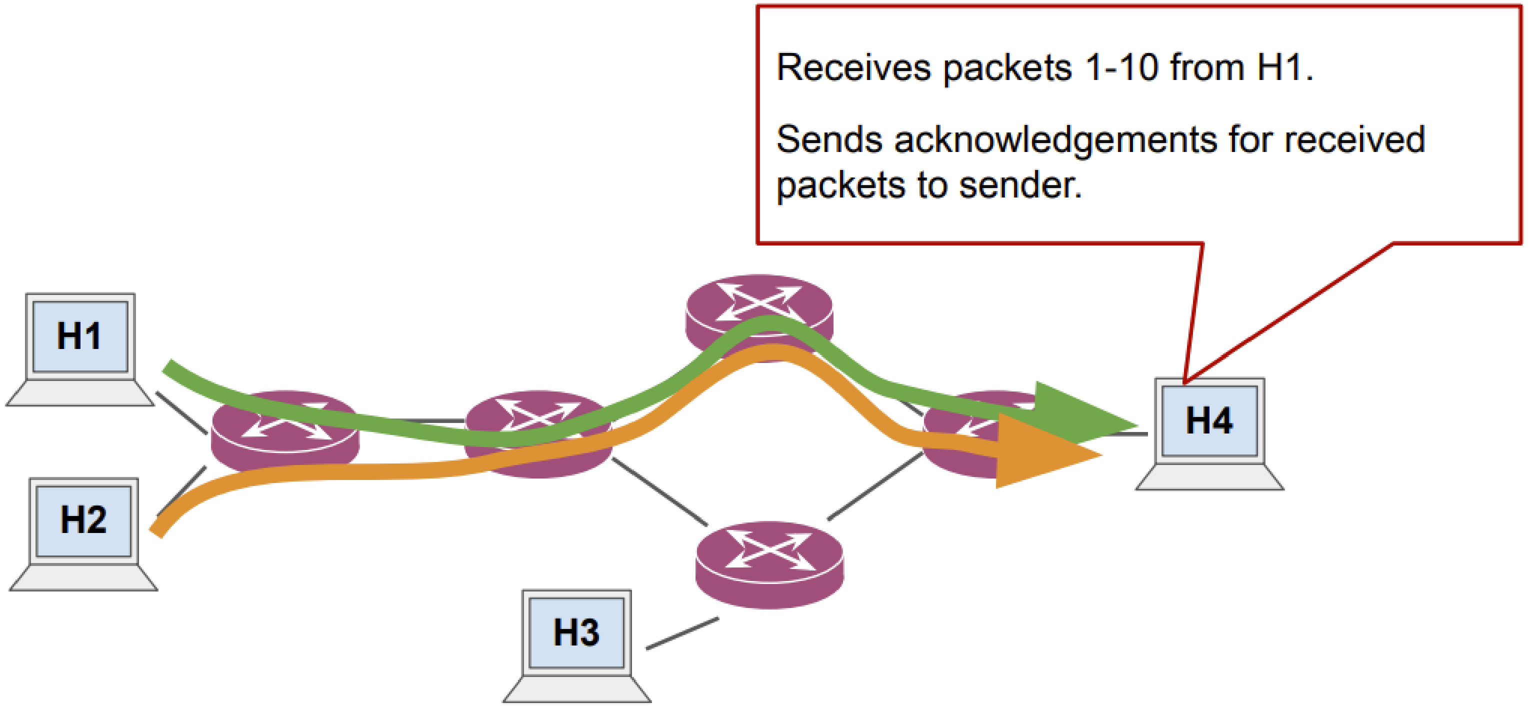


2. End-to-end Congestion Control

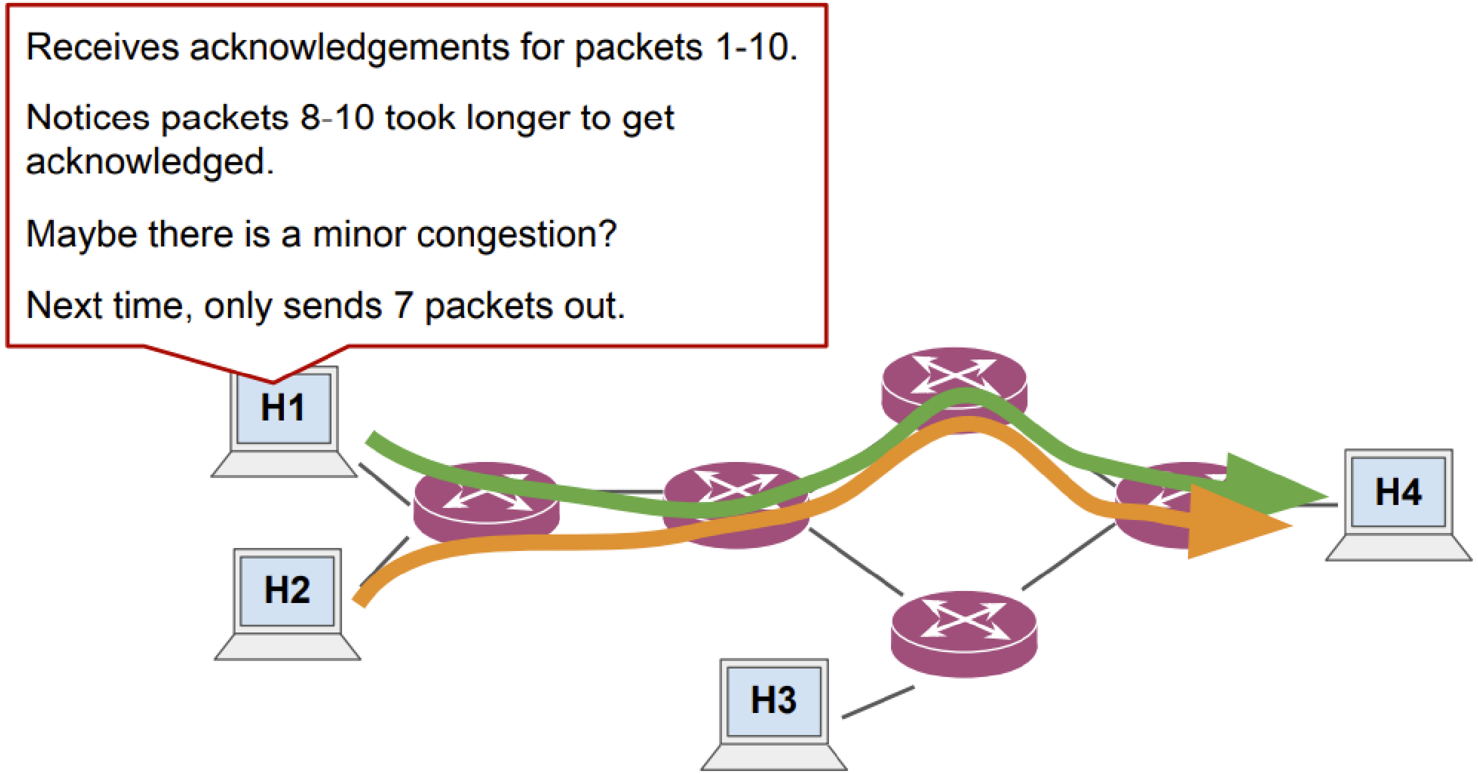


I. Transport and Network QoS

2. End-to-end Congestion Control



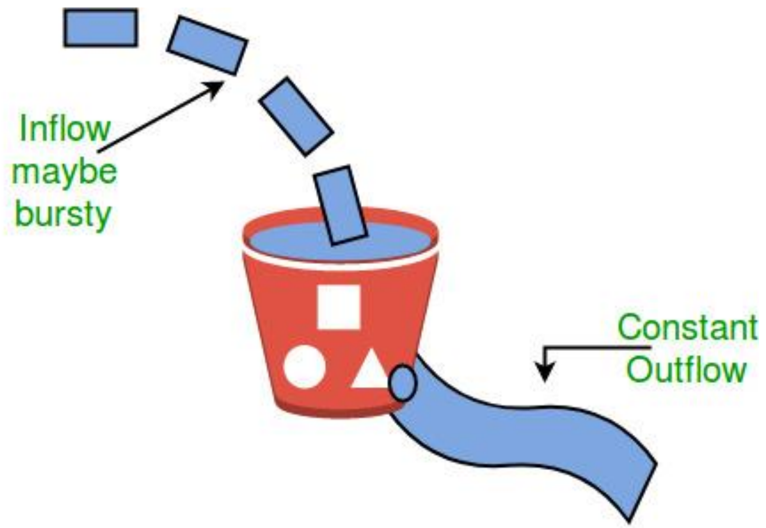
2. End-to-end Congestion Control



I. Transport and Network QoS

2. End-to-end Congestion Control

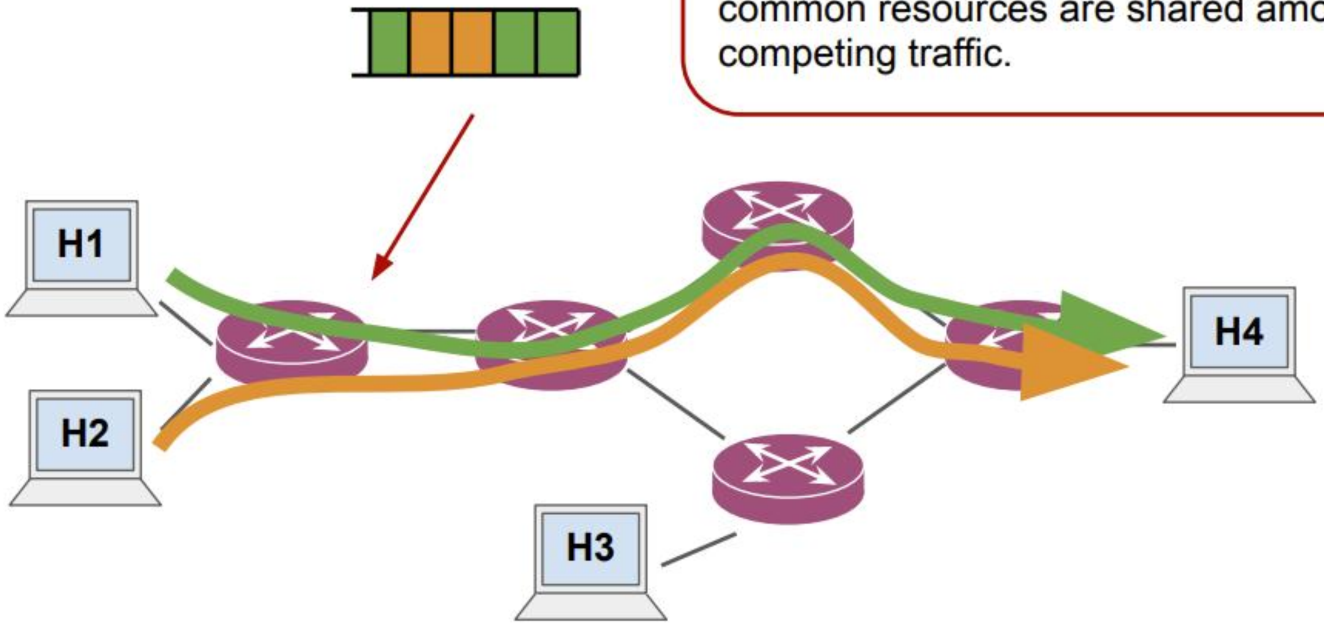
- The congestion-control algorithm decides when and how much to change the pace for each flow
- It affects how different flows in the network interact with each other at bottlenecks.
- Fairness in Bandwidth Sharing: Multiple flows sharing the same bottleneck link adjust their rates depending on congestion signals.
- Efficient Resource Utilization: End-to-end congestion control prevents persistent queue buildup and collapse due to uncontrolled packet injection.



3. Packet Scheduling

Packet schedulers decide how to line up packets in queues and who gets to go next.

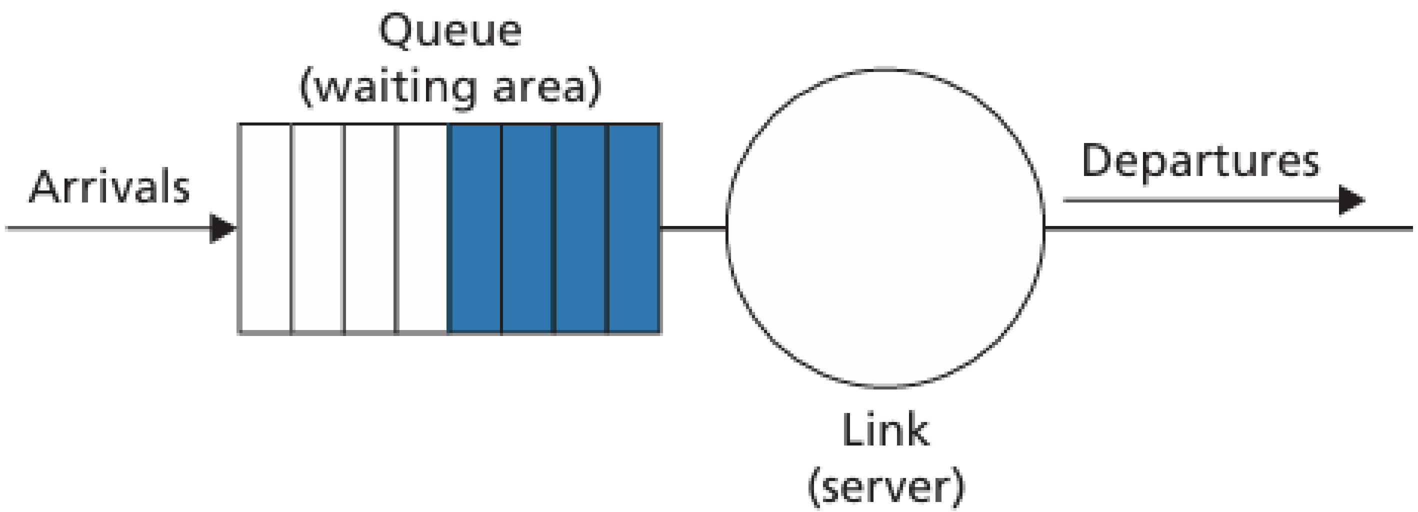
When there is contention, they can affect how common resources are shared among competing traffic.



3. Packet Scheduling

First-in-First-out (FIFO)

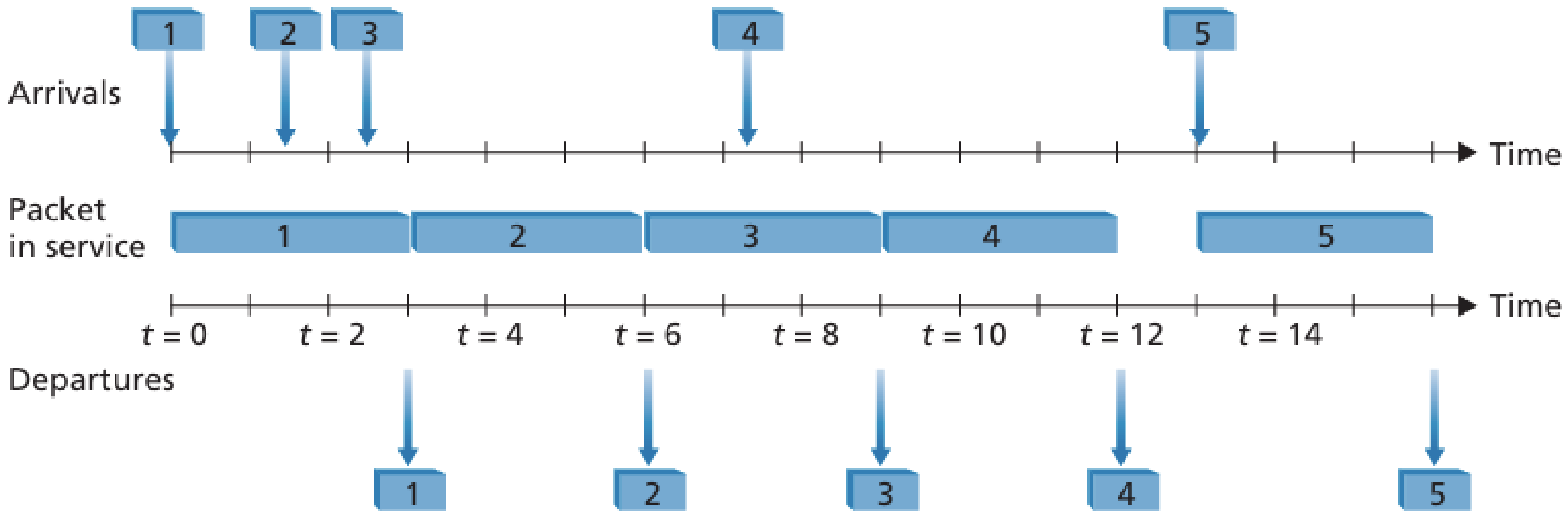
- Packets are queued up in the order they arrive and exit in the same order.



3. Packet Scheduling

First-in-First-out (FIFO)

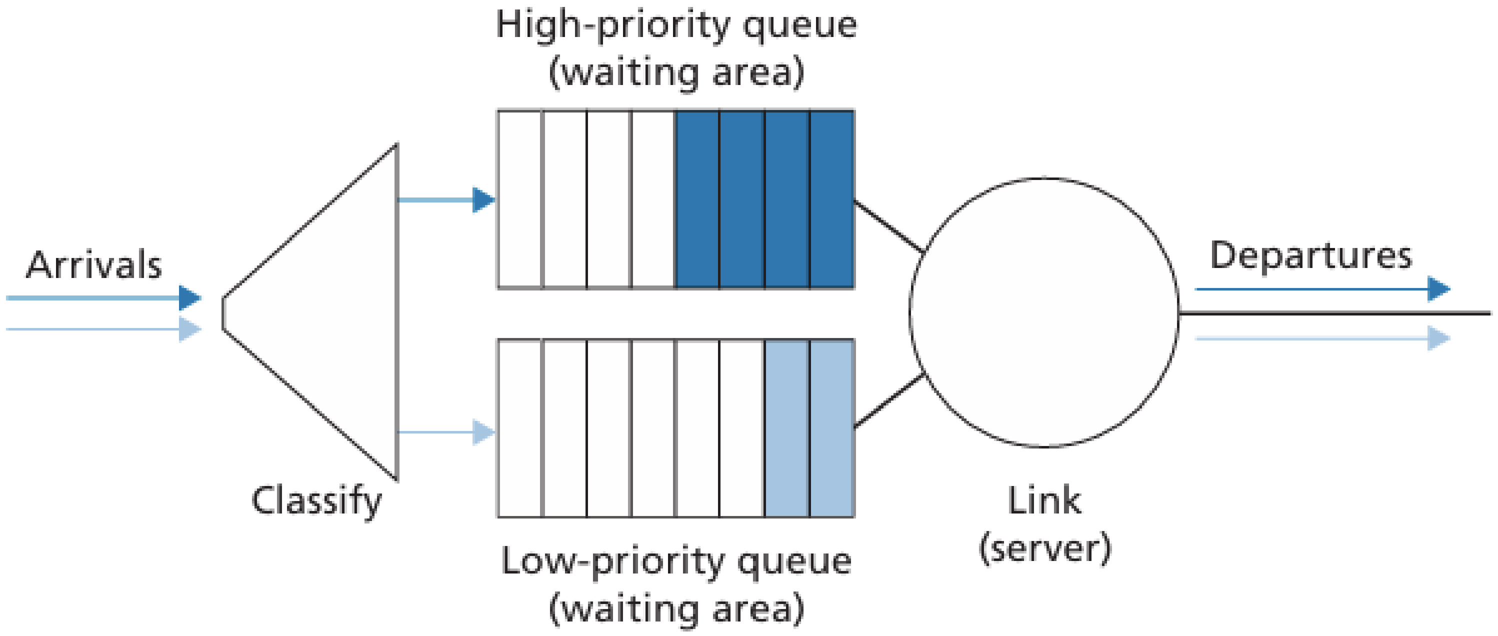
- Packets are queued up in the order they arrive and exit in the same order.



3. Packet Scheduling

Priority Queuing

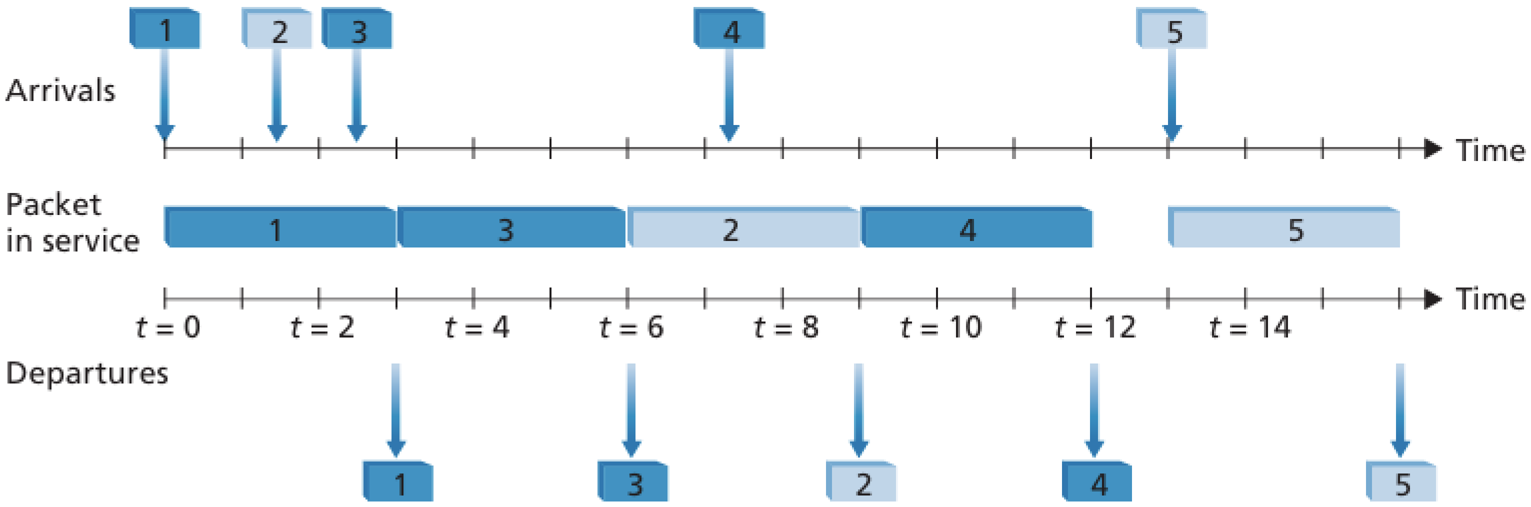
- Under priority queuing, packets arriving at the output link are classified into prior ity classes upon arrival at the queue.



3. Packet Scheduling

Priority Queuing

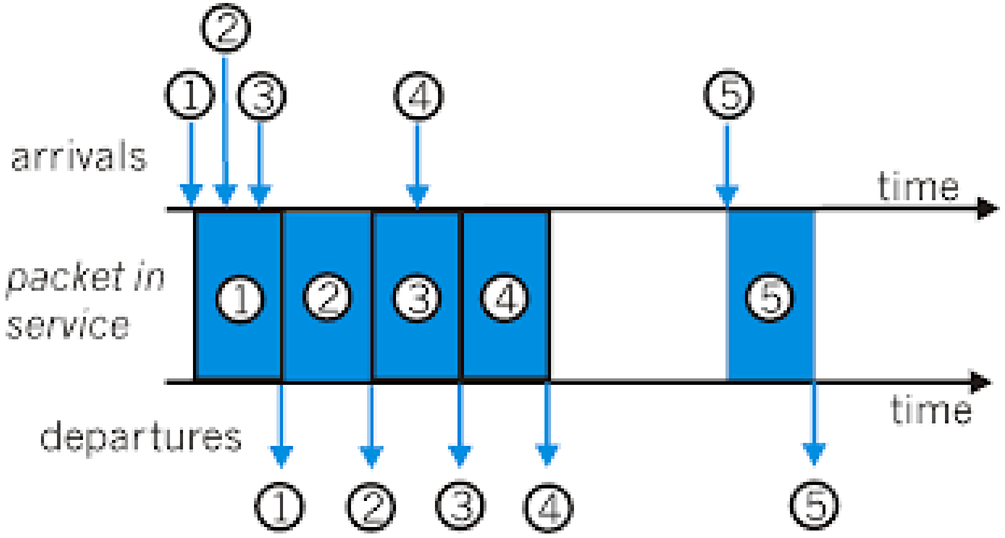
- Under priority queuing, packets arriving at the output link are classified into prior ity classes upon arrival at the queue.



3. Packet Scheduling

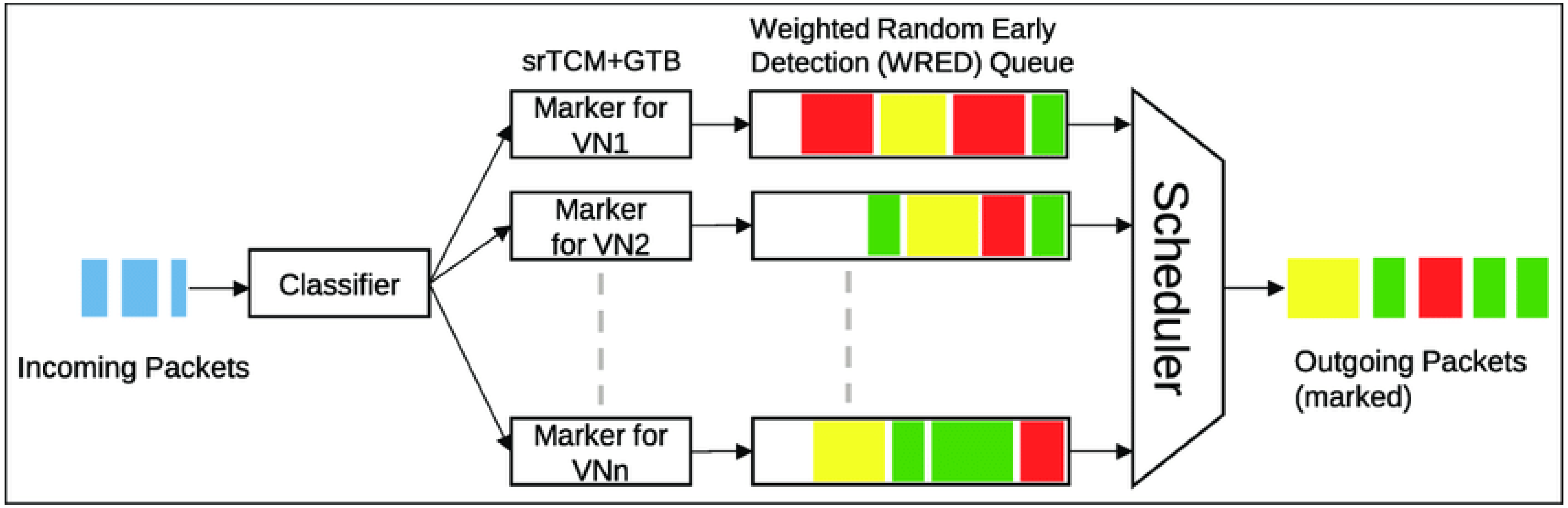
There are many other, more complex, schedulers:

- A set of FIFOs, each with its own priority.
- A set of FIFOs, serviced in round robin fashion.
- A FIFO, but packets can only be dequeued at a specific rate.
- A hierarchy of schedulers,…



4. Active Queue Management (AQM)

- AQM algorithms manage the occupancy of a single queue.
- They try to drop/mark packets before the queue is full: To keep the queue occupancy, and therefore, latency, within desirable bounds.
- Different algorithms have different ways of deciding when to start dropping/marking, whether to drop or mark, and which packets to drop/mark.



I. Transport and Network QoS

5. How has Network Programmability helped?

- Traditional networks mostly rely on end-to-end congestion control to keep network devices simple and fast.
- In traditional networks, the sender has to infer what is happening at the switch from indirect signals (delays, loss, marked packets).
- In traditional networks, the sender has to infer what is happening at the switch from indirect signals (delays, loss, marked packets).
- Why not have the switch play a more active role in handling contention with more sophisticated scheduling and AQM algorithms?

→ Network Programmability comes into play.

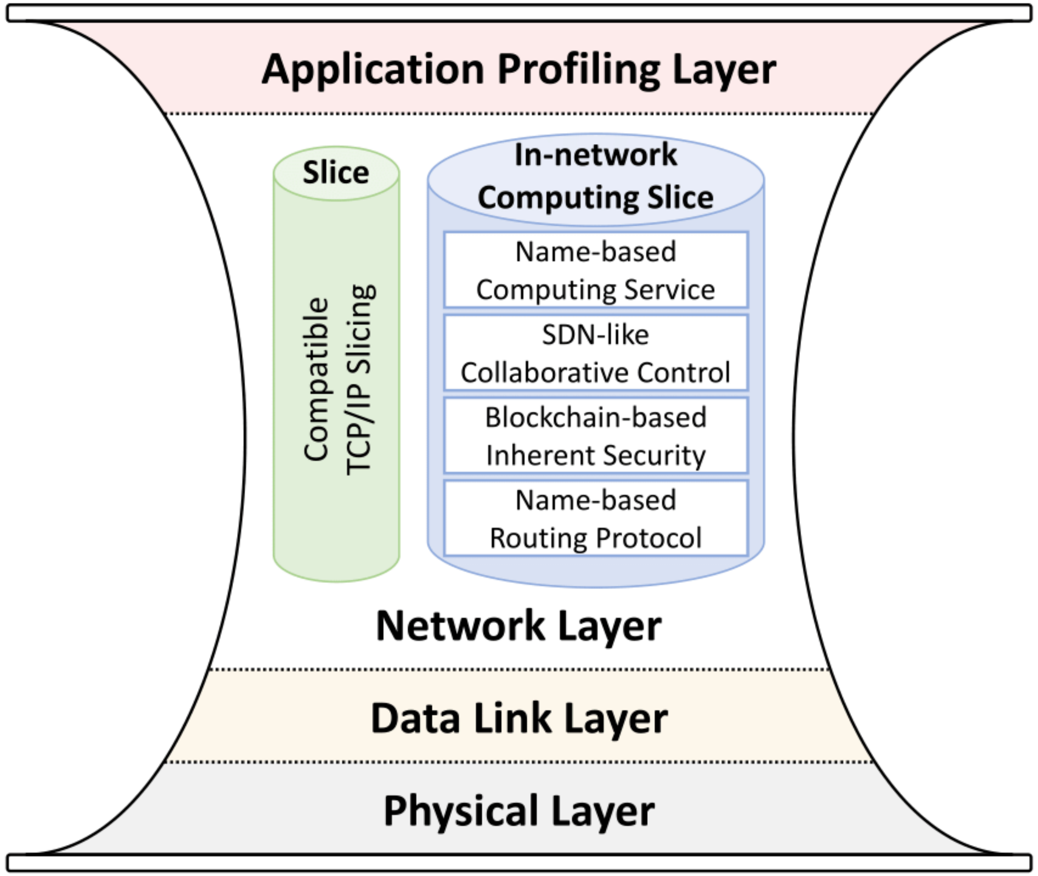
I. Transport and Network QoS

5. How has Network Programmability helped?

- Customizing the signals to e2e congestion control, scheduling, AQM, etc. to the each network and the requirements of its applications.
- Motivating new signaling, scheduling, AQM, etc. techniques.
- Better signals for congestion control algorithms.
- More complex (and flexible) packet scheduling.
- Targeted fine-grained measurements.

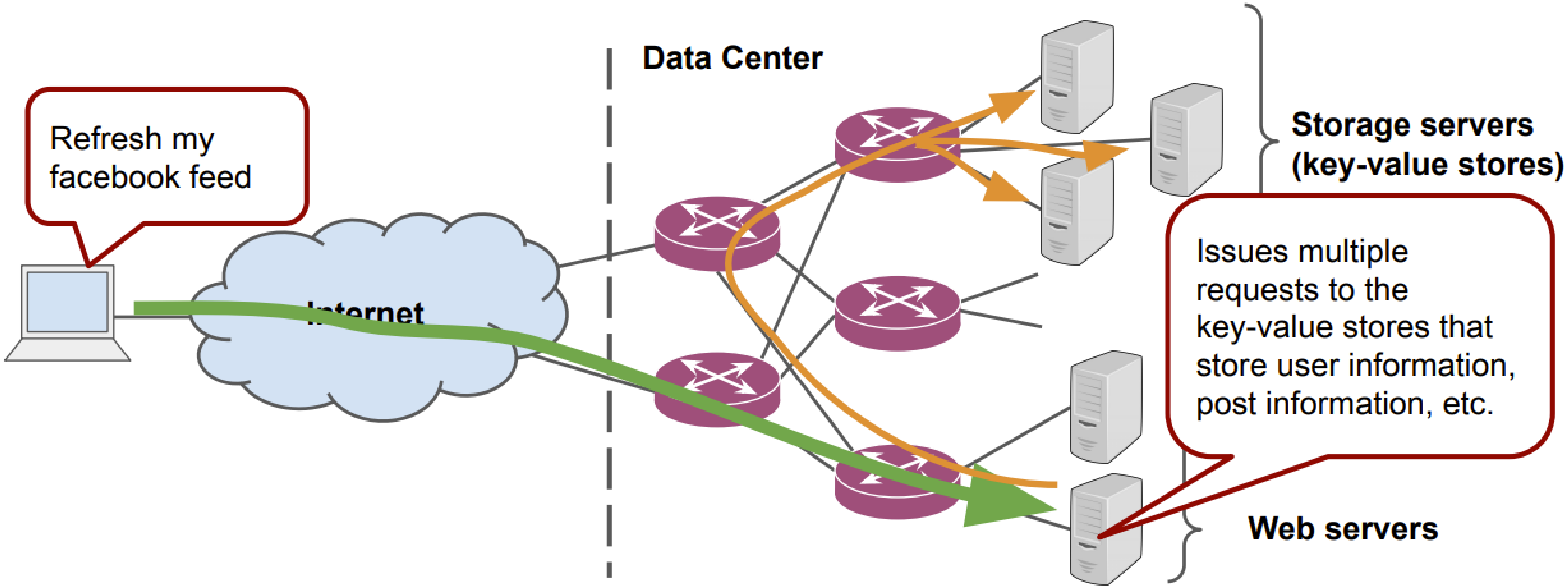
II. In-Network Computing

In-network computing means pushing some computation *inside the network devices themselves* (switches, routers, NICs), instead of doing everything at the end-hosts.



II. In-Network Computing

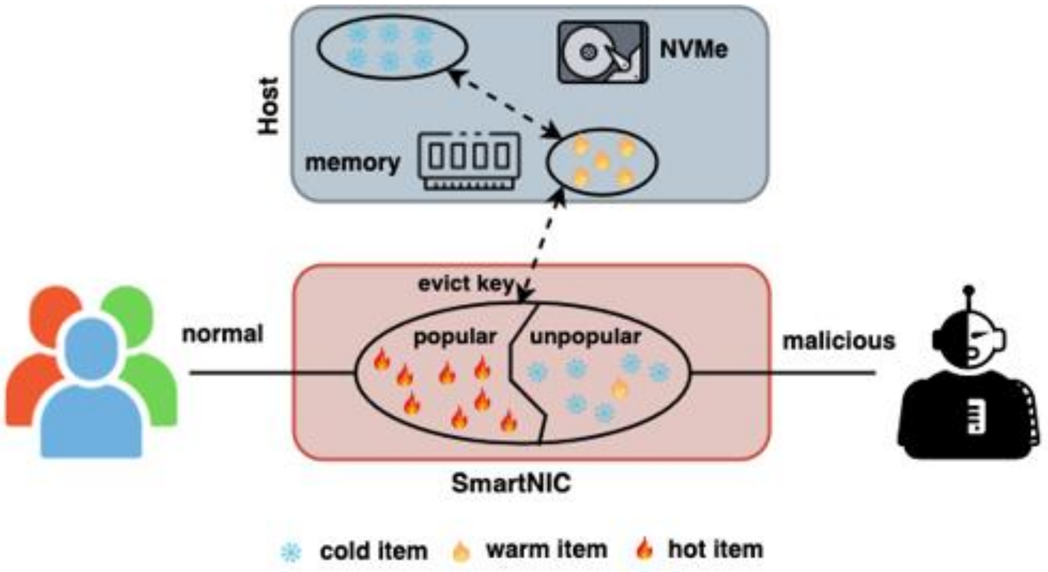
1. Example 1: In-Network Caching



II. In-Network Computing

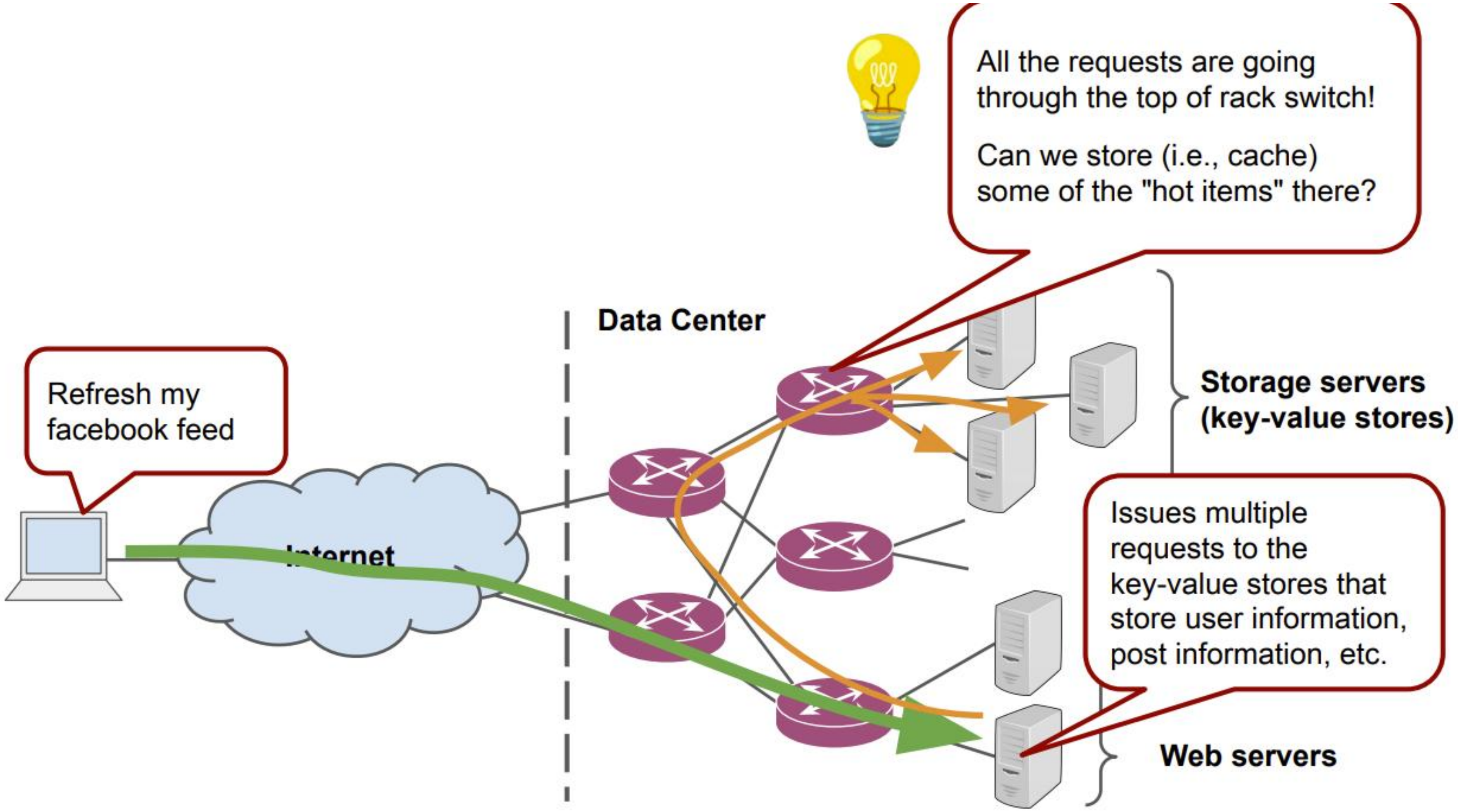
1. Example 1: In-Network Caching

- Key-value stores can get millions if not billions of requests every second.
- To handle such load, there are usually several storage servers, each taking care of part of the key-value store.
- Requests are load-balanced across storage servers.
- Problem: One server (or a subset of them) can get overwhelmed and not be able to answer queries fast enough for good user quality of experience.



II. In-Network Computing

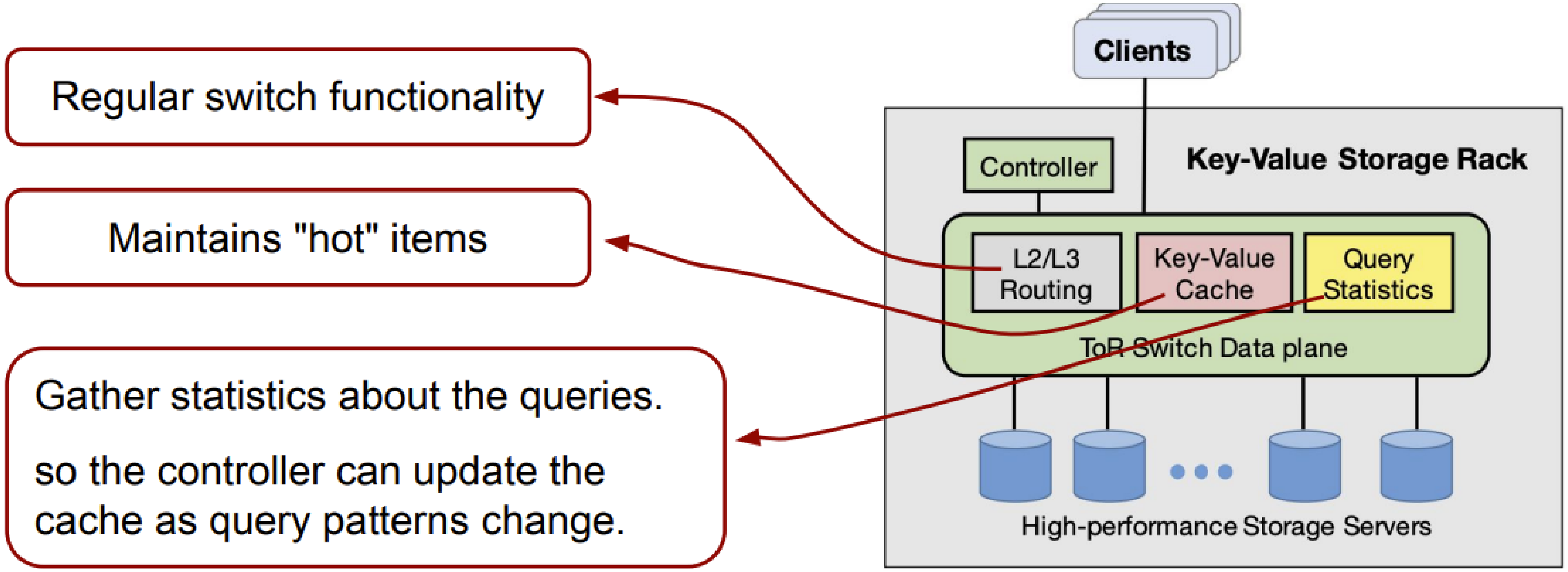
1. Example 1: In-Network Caching



II. In-Network Computing

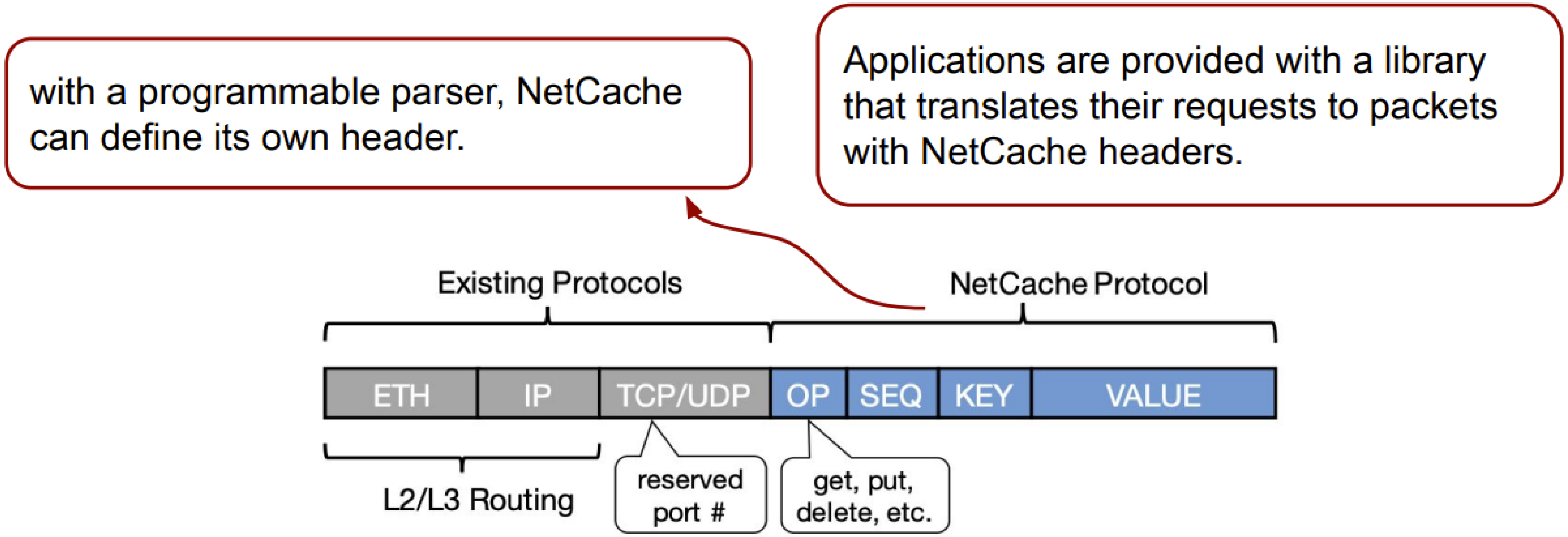
1. Example 1: In-Network Caching

NetCache proposes to do that.



1. Example 1: In-Network Caching

NetCache proposes to do that.

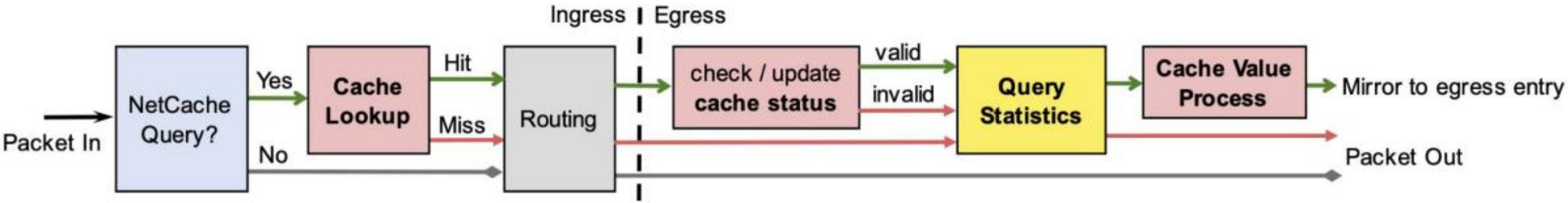


II. In-Network Computing

1. Example 1: In-Network Caching

NetCache proposes to do that.

Logical view of NetCache switch data plane



II. In-Network Computing

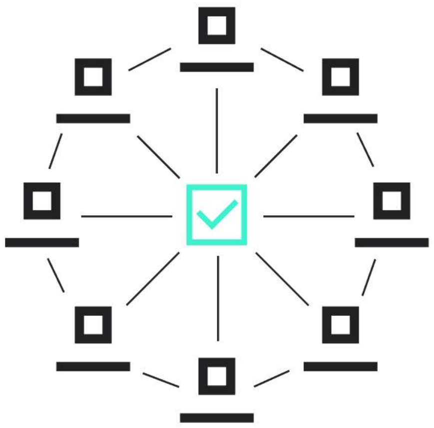
2. Example 2: In-Network consensus

- Network consensus is about getting a set of distributed nodes (machines, routers, servers, etc.) to agree on a single value or state, even when the network is unreliable, has delays, or some nodes may fail.
- In distributed systems, no single node has a full view of the system. To make progress reliably, nodes must agree on:
  - Who is the leader/primary (e.g., in a cluster)?
  - What is the next block in a blockchain?
  - What is the committed value in a replicated database?
- Without consensus, different parts of the system might diverge and behave inconsistently.

II. In-Network Computing

2. Example 2: In-Network consensus

- Consensus protocols use message exchanges between nodes to ensure:
  - Agreement → All correct nodes agree on the same value.
  - Validity → The agreed value must come from some node's input (not an invented one).
  - Termination → All correct nodes eventually decide.
  - Fault Tolerance → The system works correctly even if some nodes crash or act maliciously.



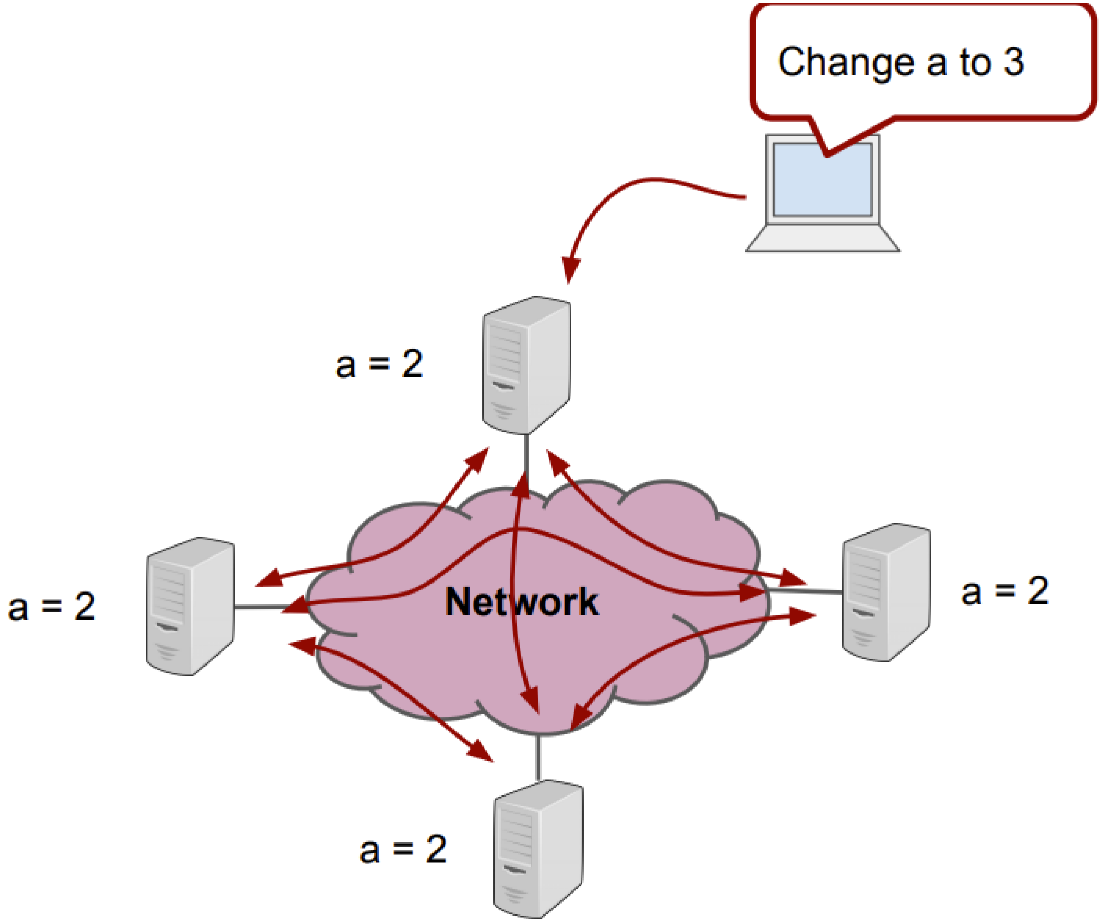
Decentralized Consensus

Centralized Consensus

2. Example 2: In-Network consensus

Each participant has its own view of the values of interest

Before any changes, participants communicate to make sure everyone is aware of the change.



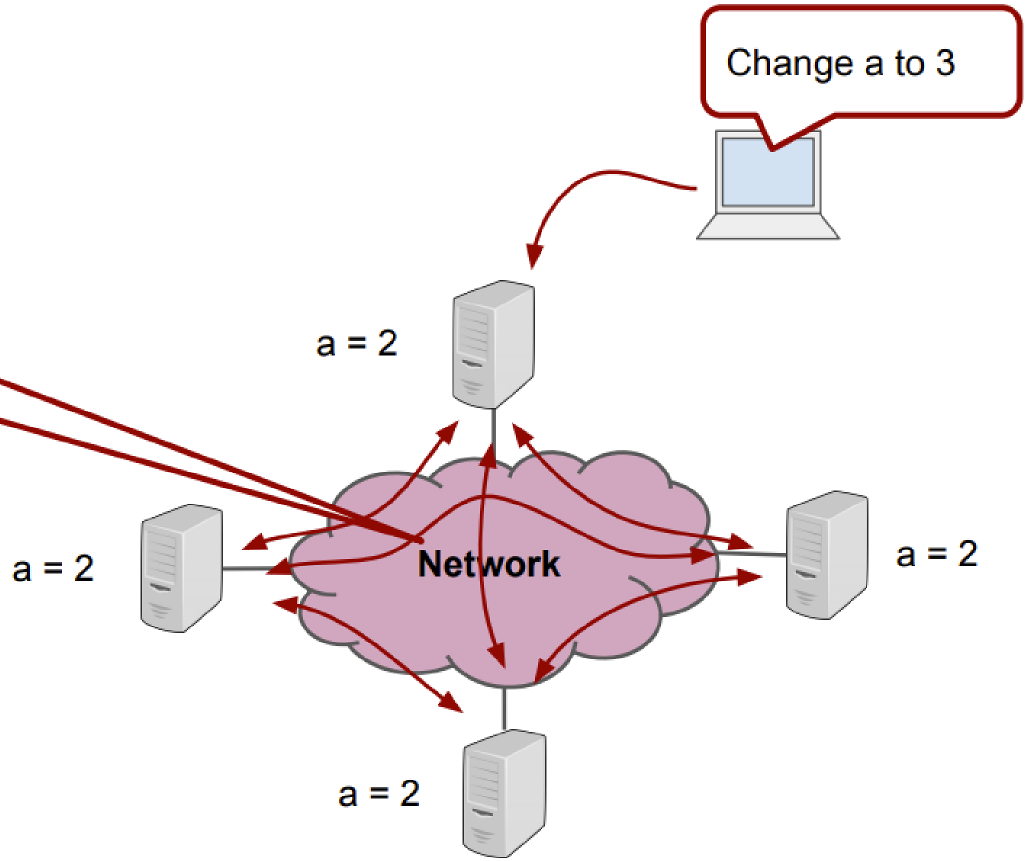
II. In-Network Computing

2. Example 2: In-Network consensus

Can we implement it in the network?

Consensus is communication heavy

the actual computations done on each participant is quite simple.



II. In-Network Computing

2. Example 2: In-Network consensus

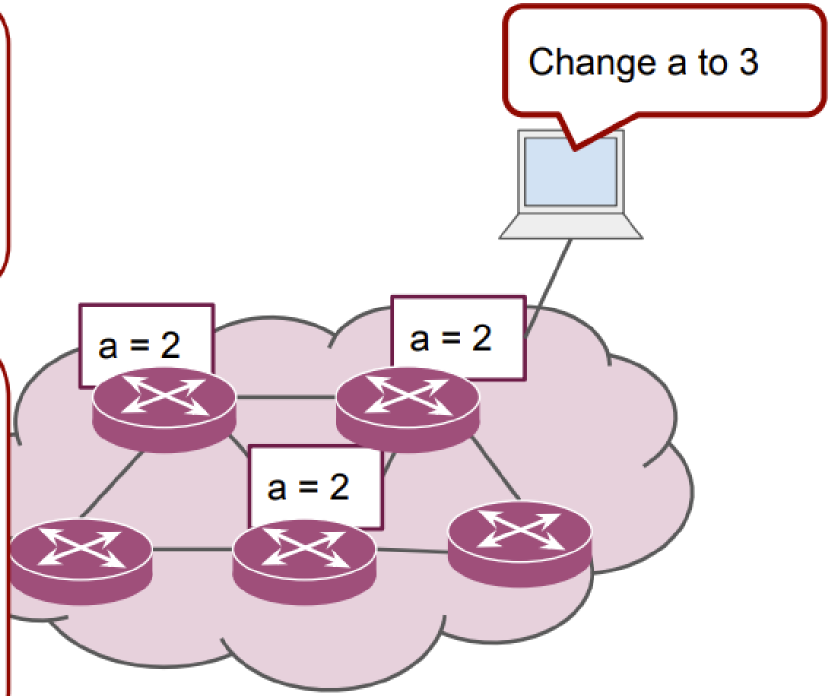
Switches keep all copies of the values.

Switches serve read and write requests.

Switches run the consensus (or coordination, or agreement) protocol.

Benefits?

- Switches are faster than servers
- Communication between each pair of servers requires the traversal of multiple switches (multiple RTTs)
- Switches are "closer" to each other, so this can be done even in sub-RTT



II. In-Network Computing

3. Challenges of In-Network computing

- Limited programmability: Switches and routers are designed for line-rate forwarding, not general-purpose computing.
- Resource constraints: Switch chips (ASICs) have tiny amounts of fast memory (SRAM/TCAM) compared to servers.
- Security and trust: In-network functions may have access to sensitive data.
- Standardization and portability: Different vendors provide different levels of programmability (Barefoot Tofino, Intel, NVIDIA BlueField DPUs).

HUST

hust.edu.vn fb.com/dhbkhn

THANK YOU !