Universal Packet Scheduling

Radhika Mittal, Rachit Agarwal,
Sylvia Ratnasamy, Scott Shenker

UC Berkeley

Packet Scheduling

- **Active research literature with many**
  - **Algorithms**
    - FIFO, DRR, virtual clocks, priorities…
  - **Goals**
    - fairness, small packet delay, small FCT…
  - **Contexts**
    - WAN, datacenters, cellular…

We are asking a new question…..

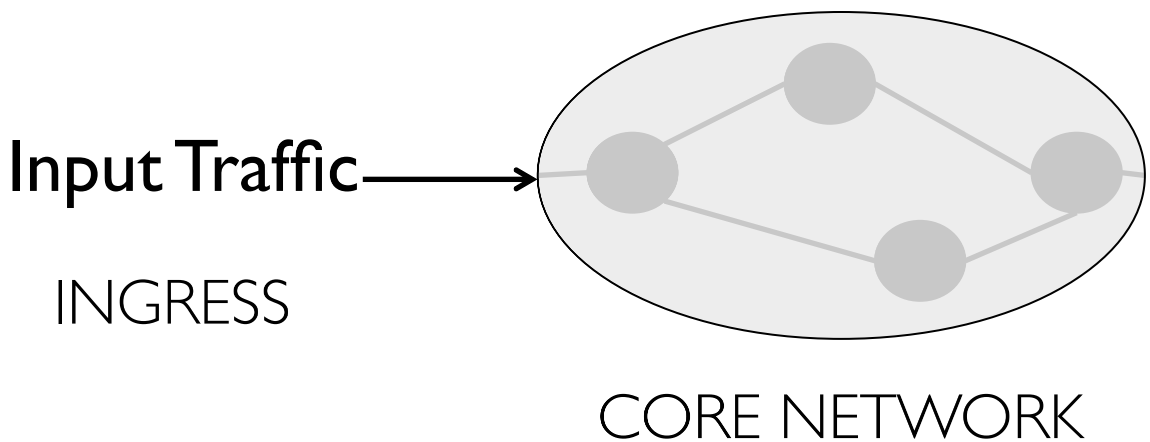
Is there a *universal* packet scheduling algorithm?

UPS: Universal Packet Scheduling Algorithm

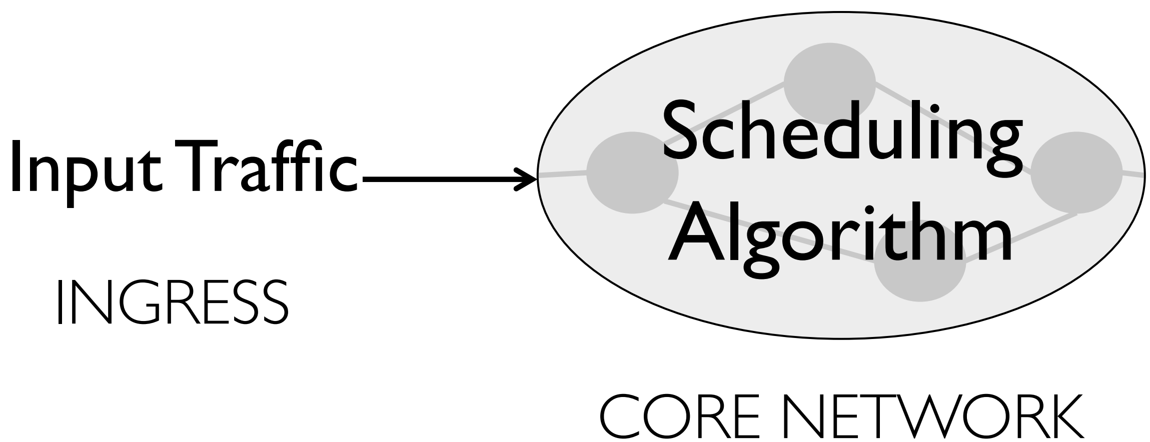
A *single scheduling algorithm* that can imitate the network-wide output produced by ***any*** other algorithm.

How can a single algorithm imitate all others?

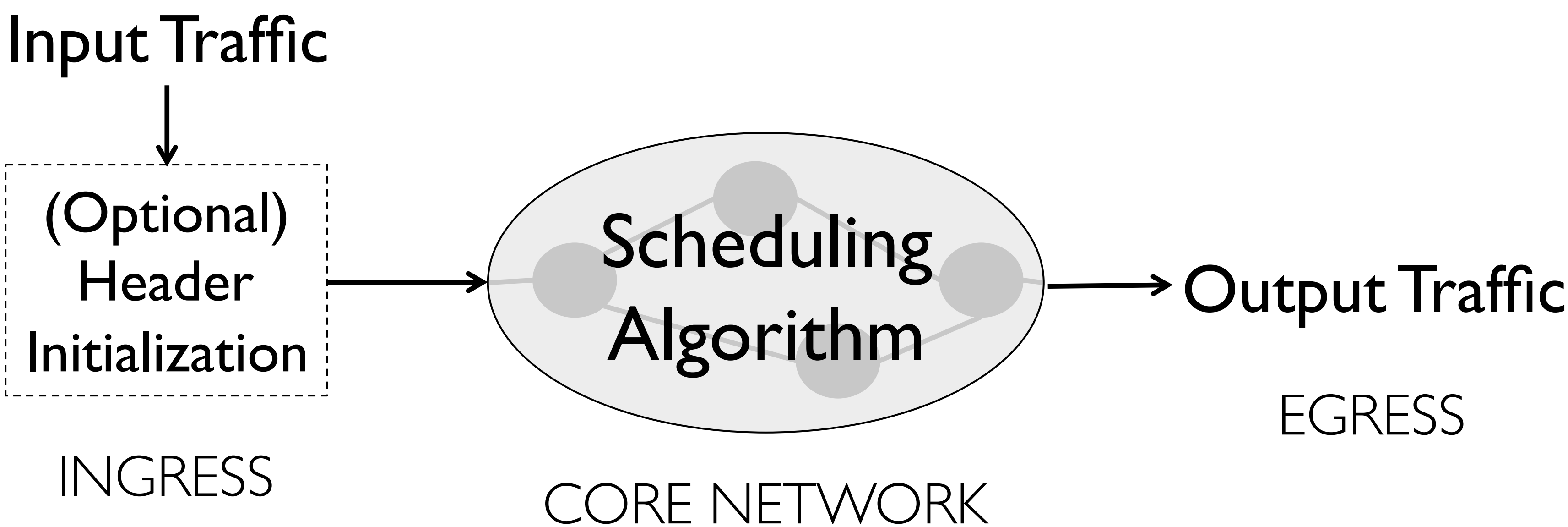
Network-wide Model



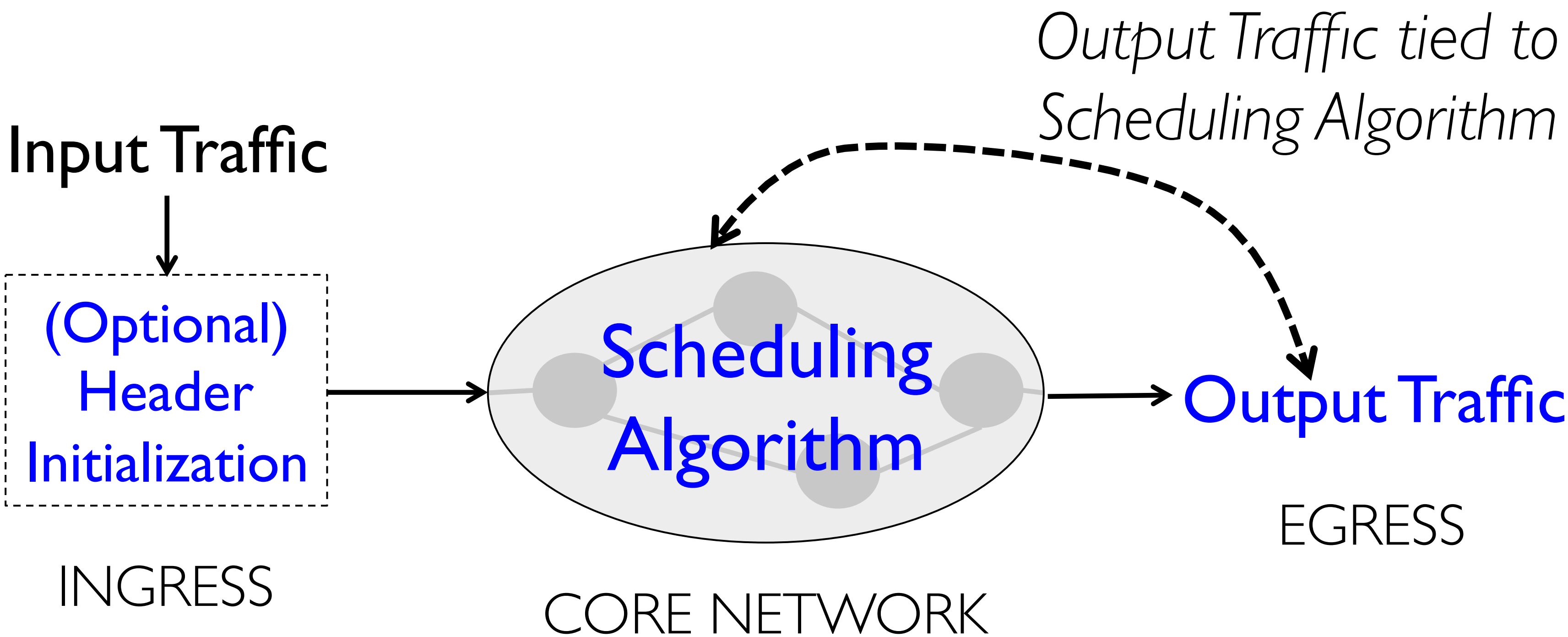
Network-wide Model



Network-wide Model

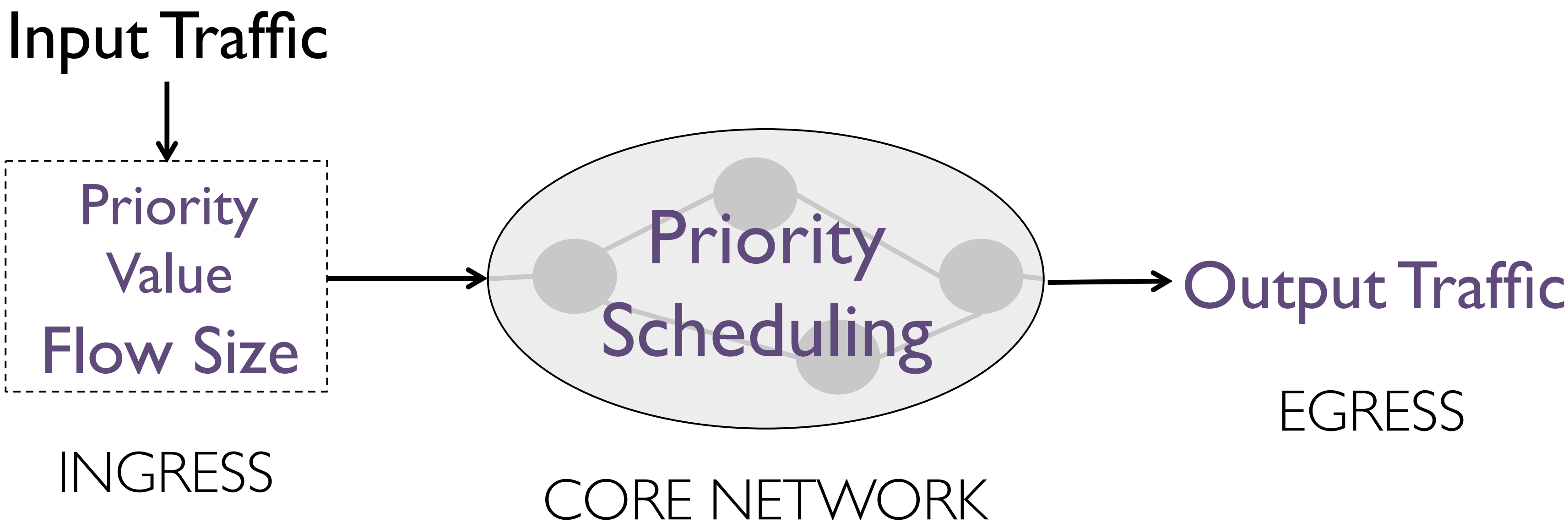


Network-wide Model



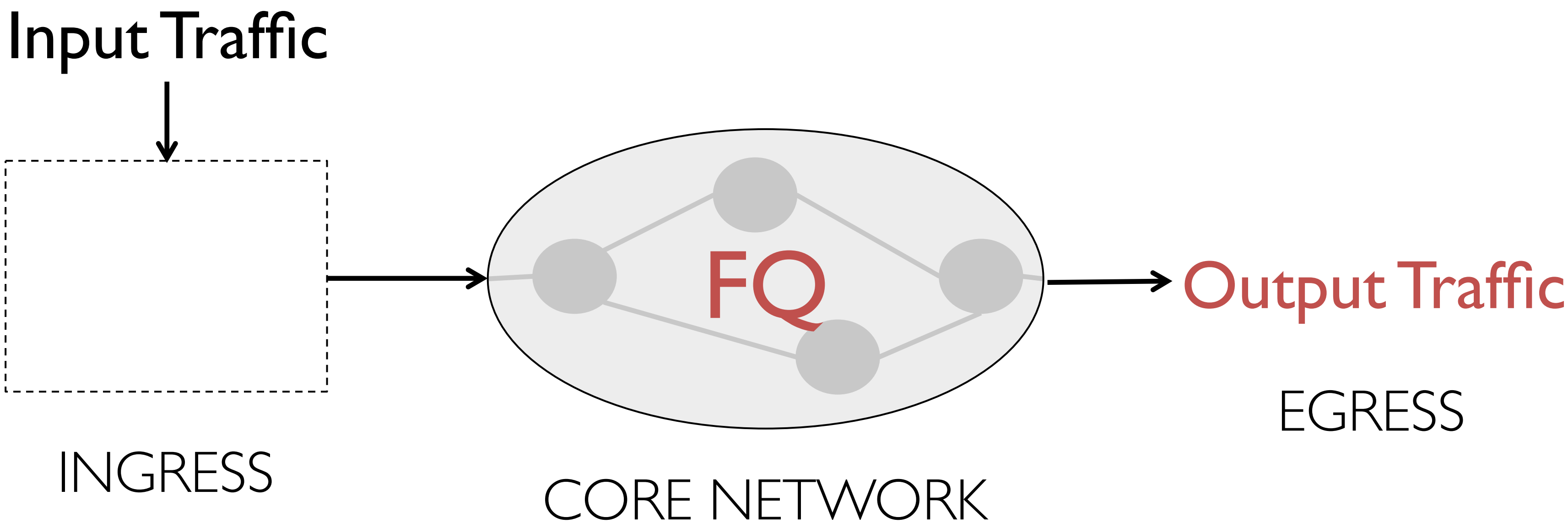
Network-wide Model

Goal: Minimize Mean FCT



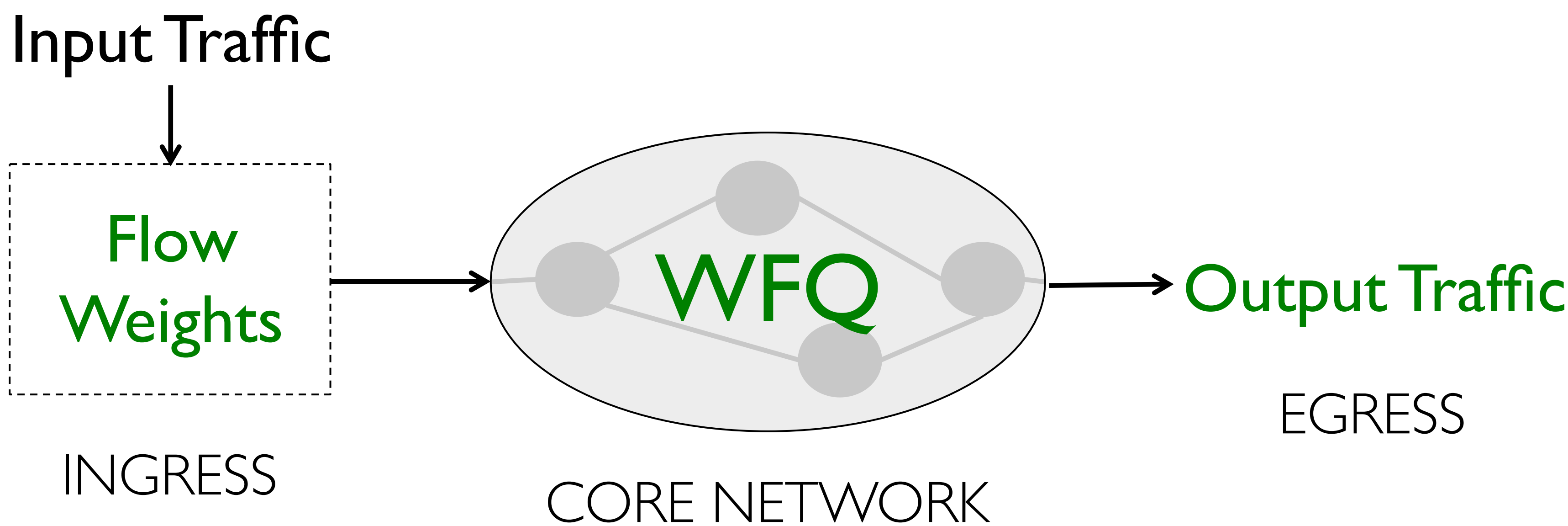
Network-wide Model

Goal: Fairness

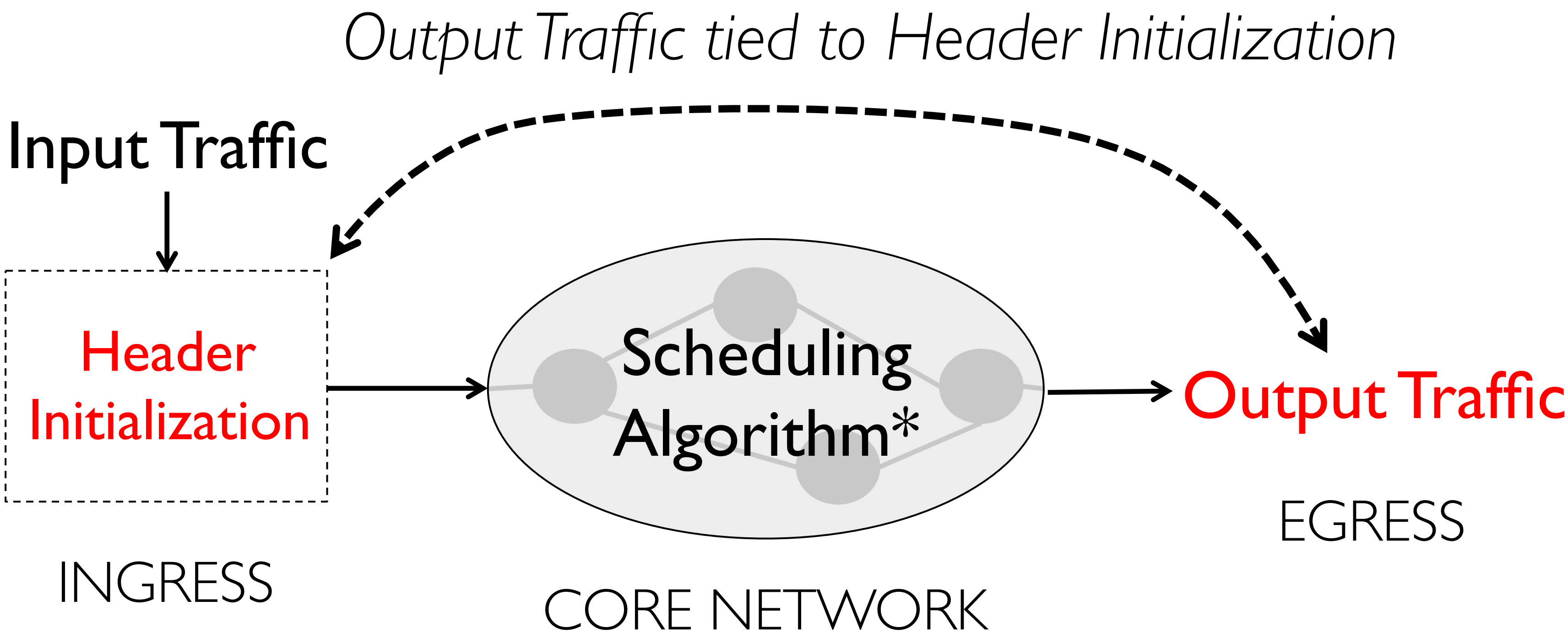


Network-wide Model

Goal: Weighted Fairness

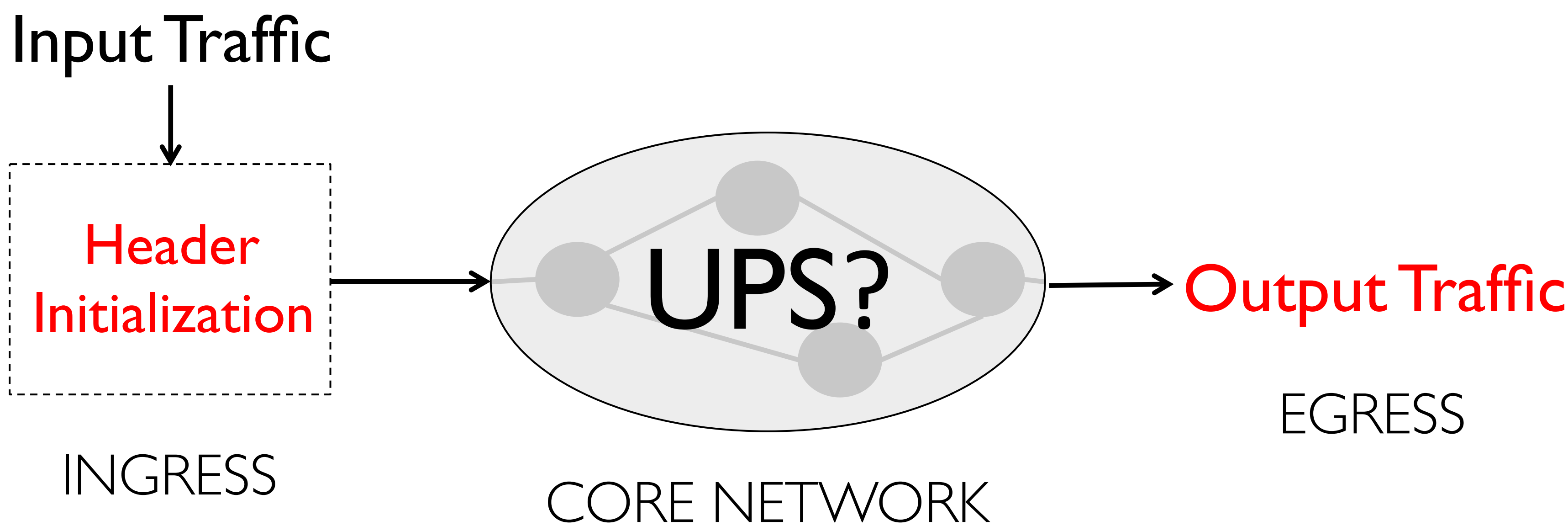


Network-wide Model



* Uses packet header state to make scheduling decisions

Network-wide Model



Universality

vs

Programmability

Programmability:

- **Single hardware** for multiple algorithms to meet multiple goals

Universality:

- **Single algorithm** with varying header initializations for multiple goals

How do we formally define and evaluate a UPS?

Defining a UPS

Theoretical Viewpoint:

Can it replay a given schedule?

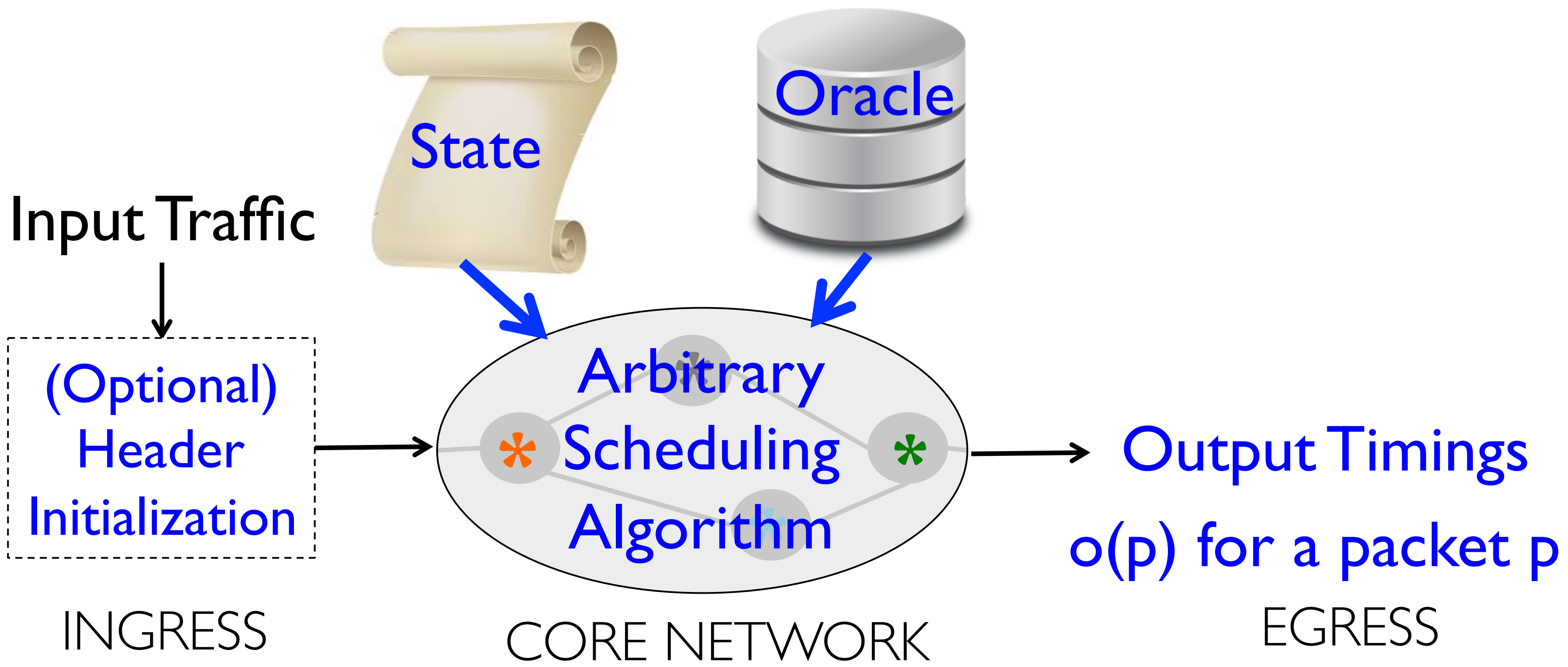
Practical Viewpoint:

Can it achieve a given objective?

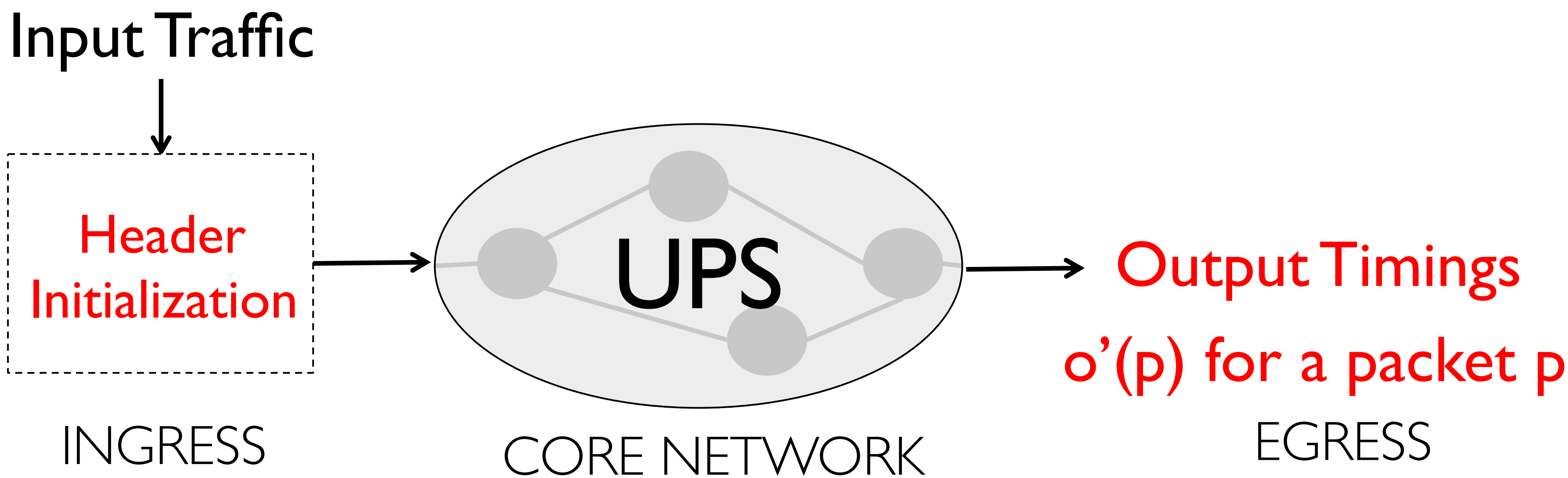
Theoretical Viewpoint

Can it replay a given schedule?

Original Schedule

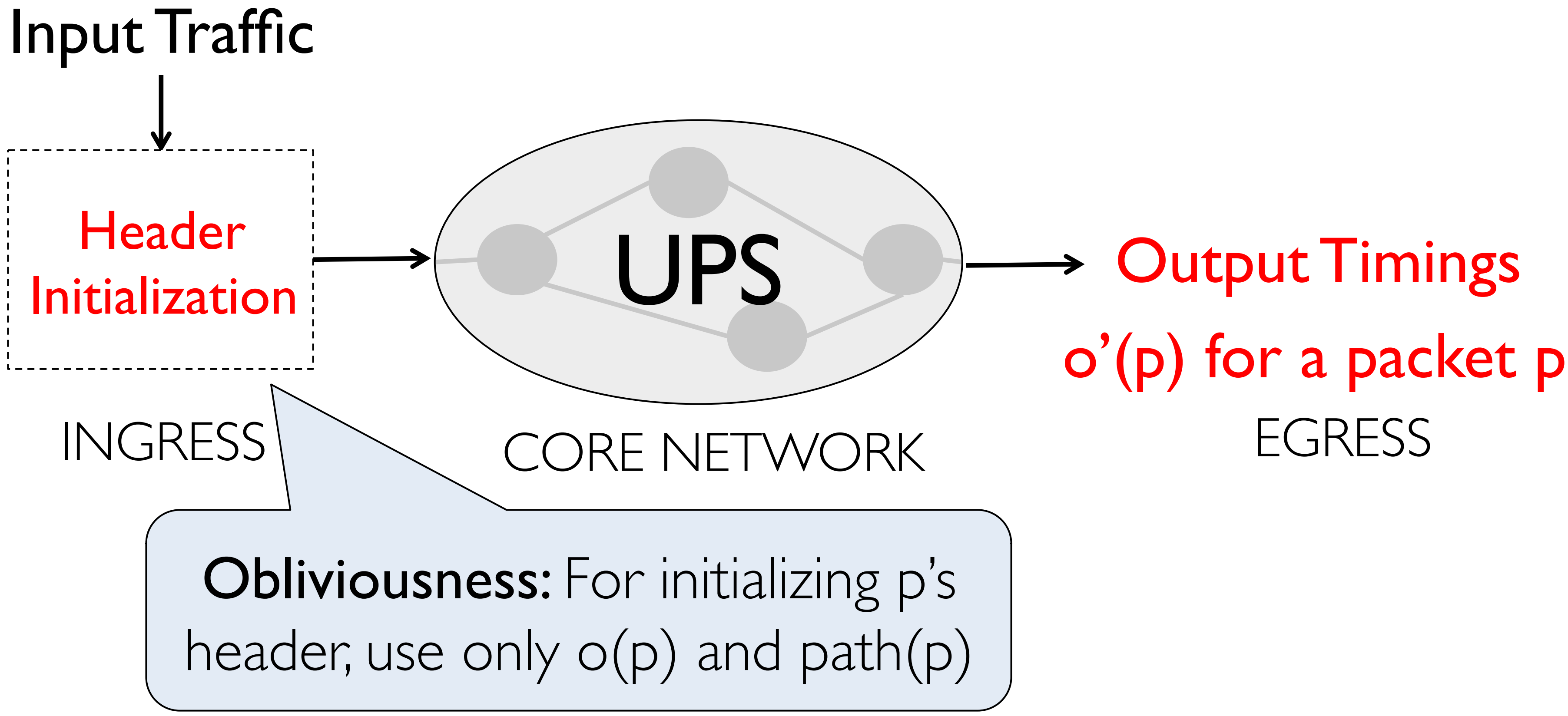


Replaying the Schedule, given o(p)

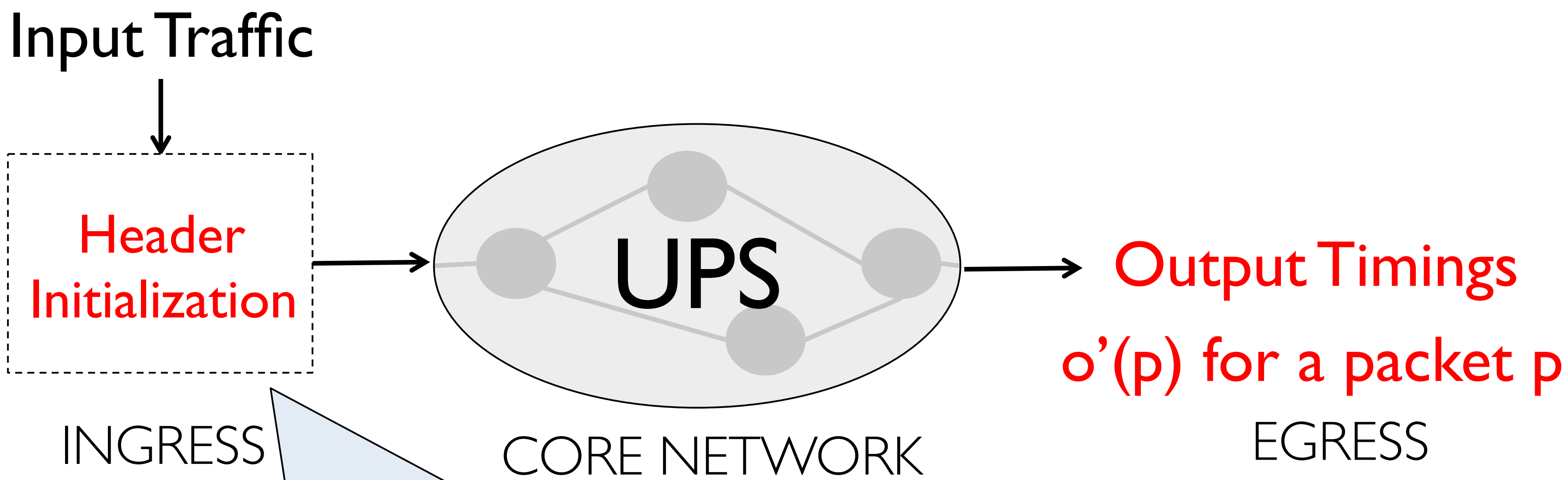


For every packet p, o'(p) $\leq$ o(p)

Pragmatic Constraints on a UPS

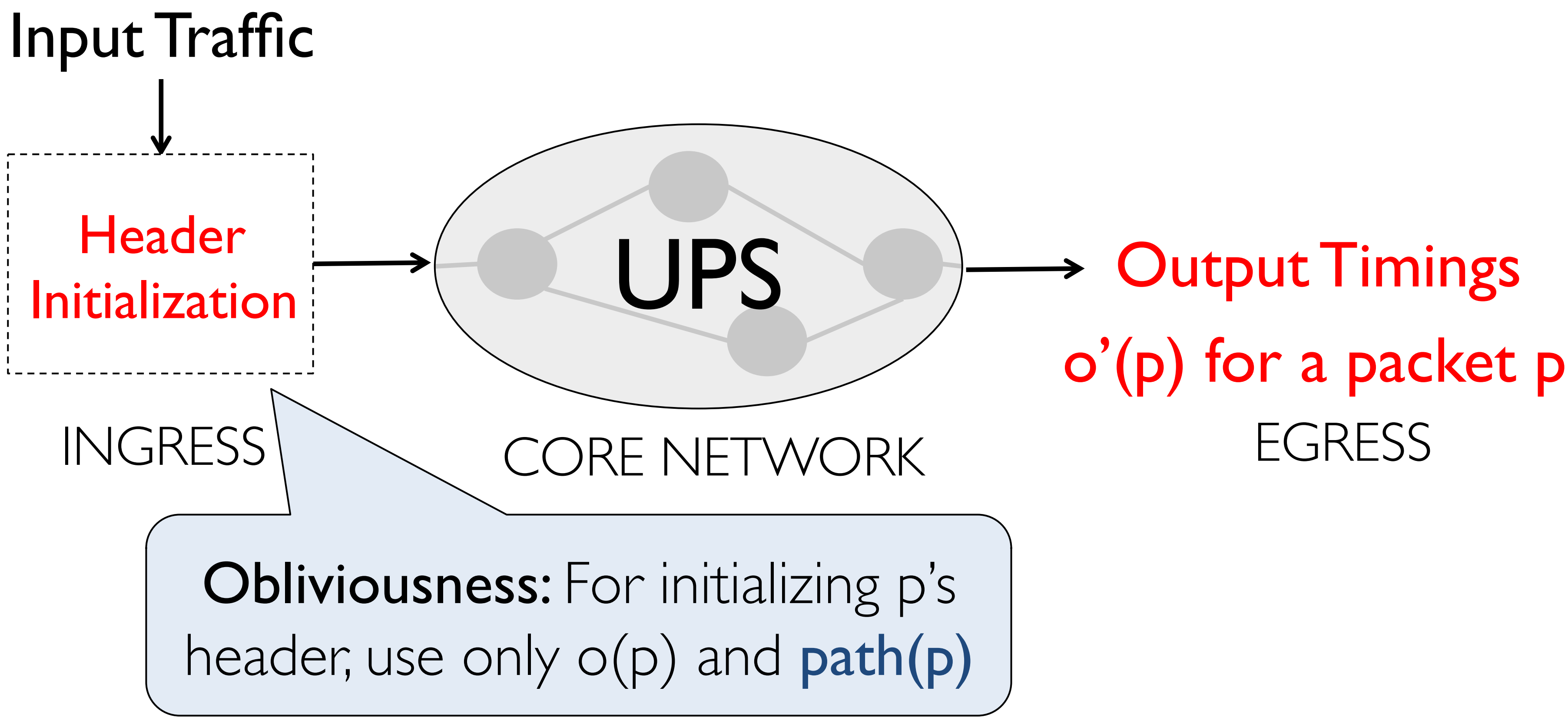


Pragmatic Constraints on a UPS

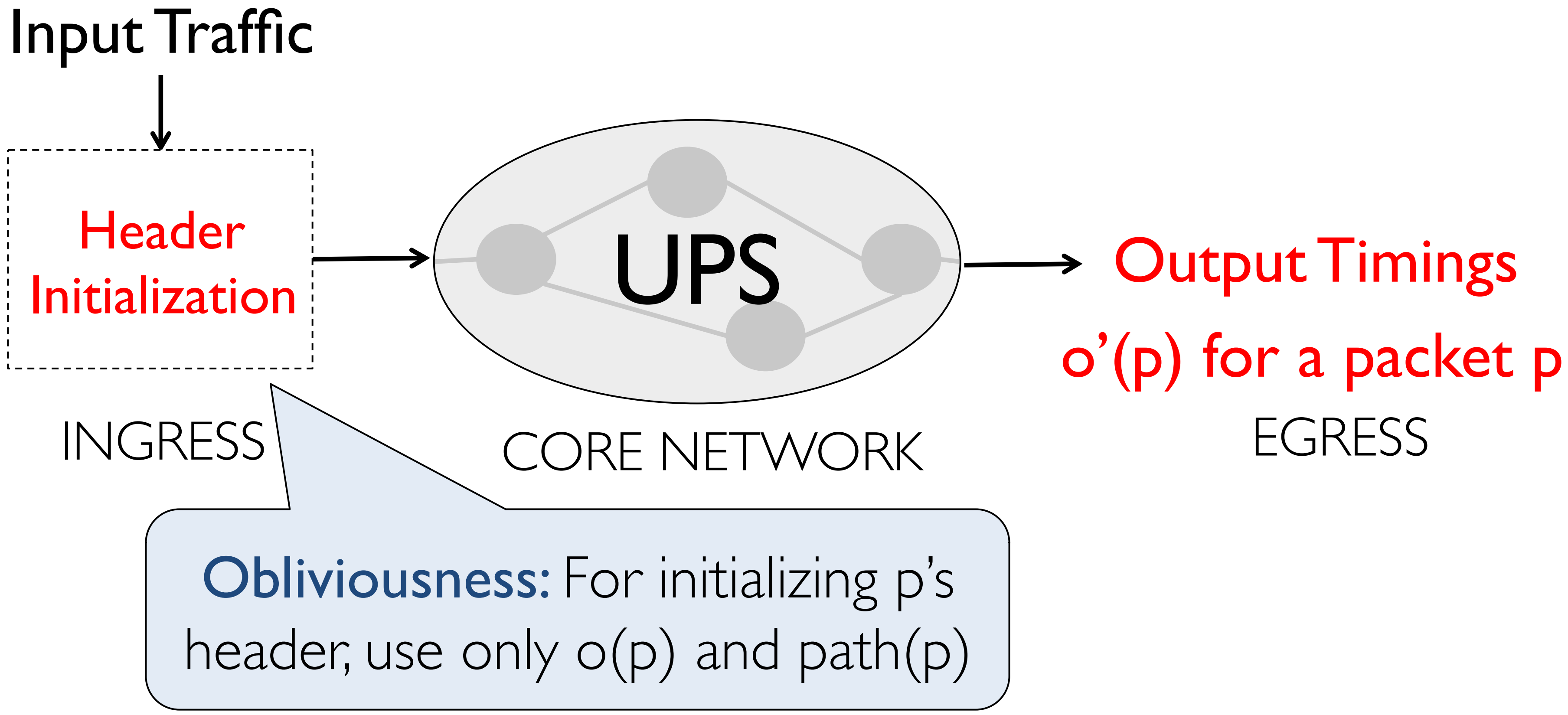


Obliviousness: For initializing p's header, use only o(p) and path(p)

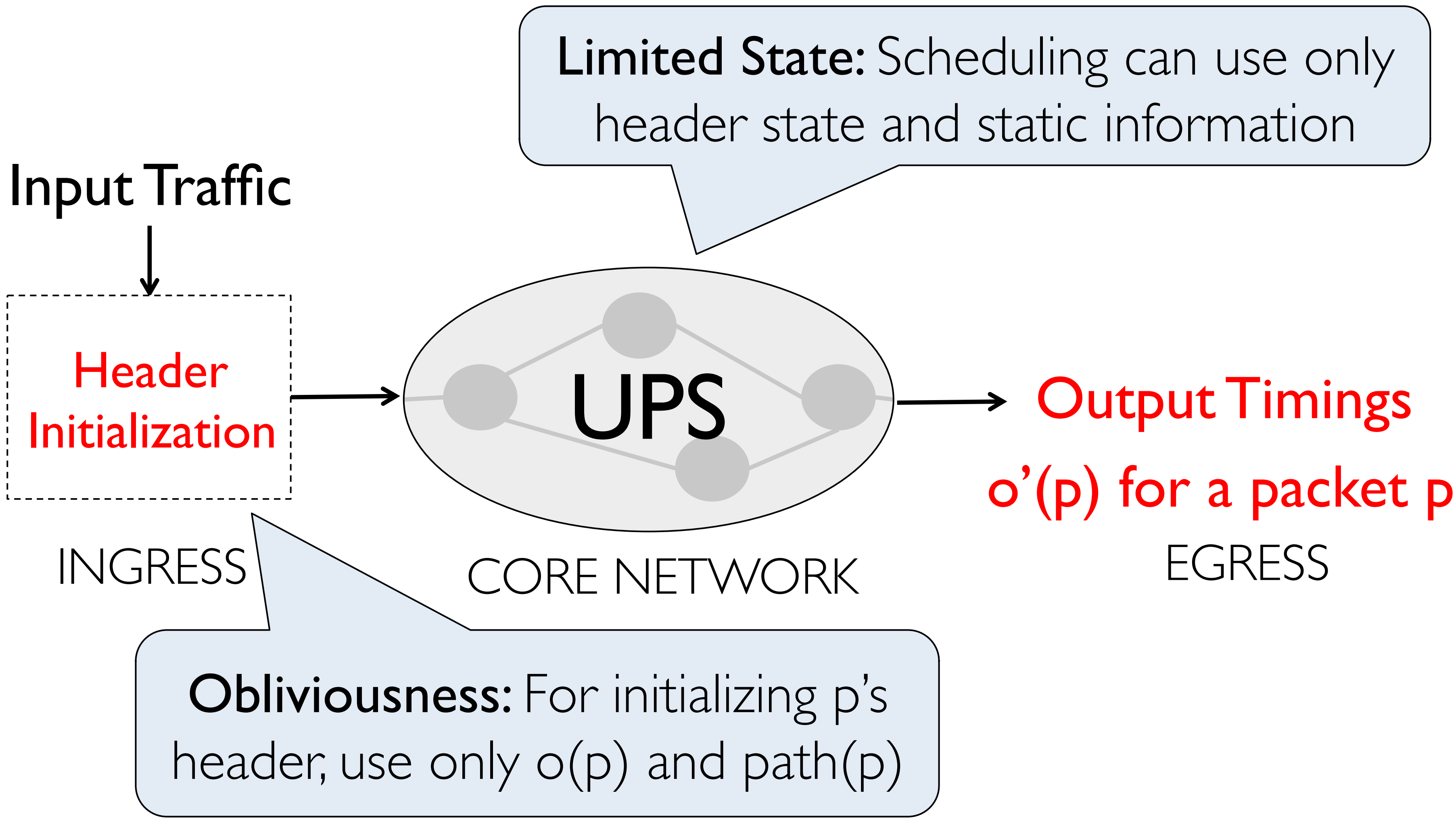
Pragmatic Constraints on a UPS



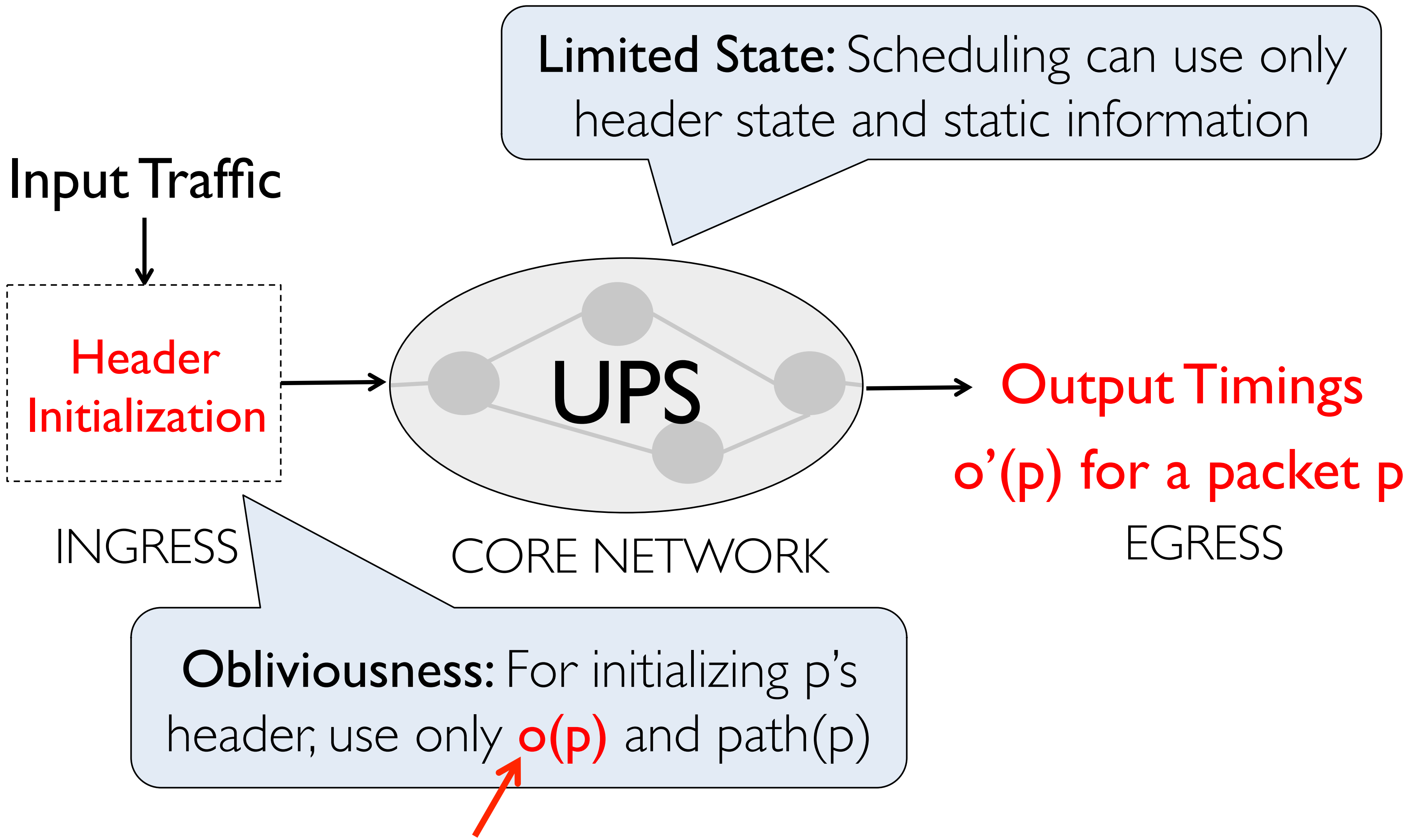
Pragmatic Constraints on a UPS



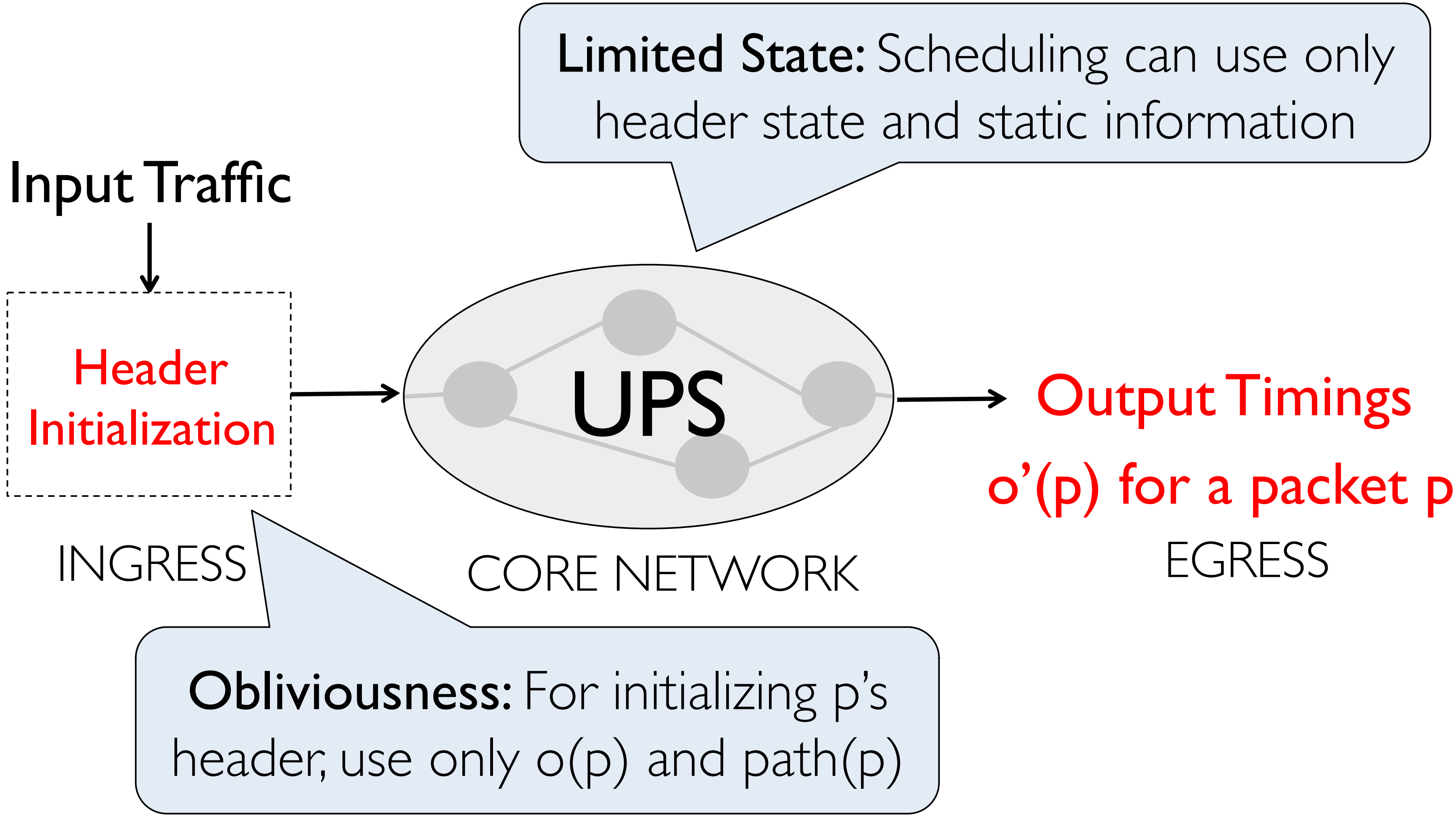
Pragmatic Constraints on a UPS



Pragmatic Constraints on a UPS



We call this Blackbox Initialization



Basic Existence and Non-existence Results

There exists a UPS under *Omniscient Initialization*

when scheduling time at every hop is known

No UPS exists under *Blackbox Initialization*

when only the final output time is known

How close can we get to a UPS?

Key Result: Depends on congestion points

No. of Congestion Points per Packet	General
1	✓
2	✓
3	✗

Can we achieve this upper bound?

Can we achieve this upper bound?

Yes, LSTF!

Least Slack Time First

- Packet header initialized with a slack value
  - slack = maximum tolerable queuing delay
- At the routers
  - Schedule packet with least slack time first
  - Update the slack by subtracting the wait time

Alternate EDF-based implementation:
Static deadlines in packet headers with additional state in the routers.

Key Results

No. of Congestion Points per Packet	General	LSTF
1	✓	✓
2	✓	✓
3	✗	✗

Not all algorithms achieve upper bound

No. of Congestion Points per Packet	General	LSTF	Priorities
1	✓	✓	✓
2	✓	✓	✗
3	✗	✗	✗


How well does LSTF perform empirically?

Empirically, LSTF is (almost) universal

- ns-2 simulation results on realistic network settings
  - Less than 3% packets missed their output times
  - Less than 0.1% packets are late by more than one transmission time

Summarizing the theoretical viewpoint

- Evaluate the ability to replay a schedule, given its final output times

- Analytical Results:
  - No UPS exists
  - LSTF comes as close to a UPS as possible

- Empirical Results: LSTF is *almost* universal!

Practical Viewpoint

Can it achieve a given objective?

Achieving various network objectives

- Slack assignment based on heuristics

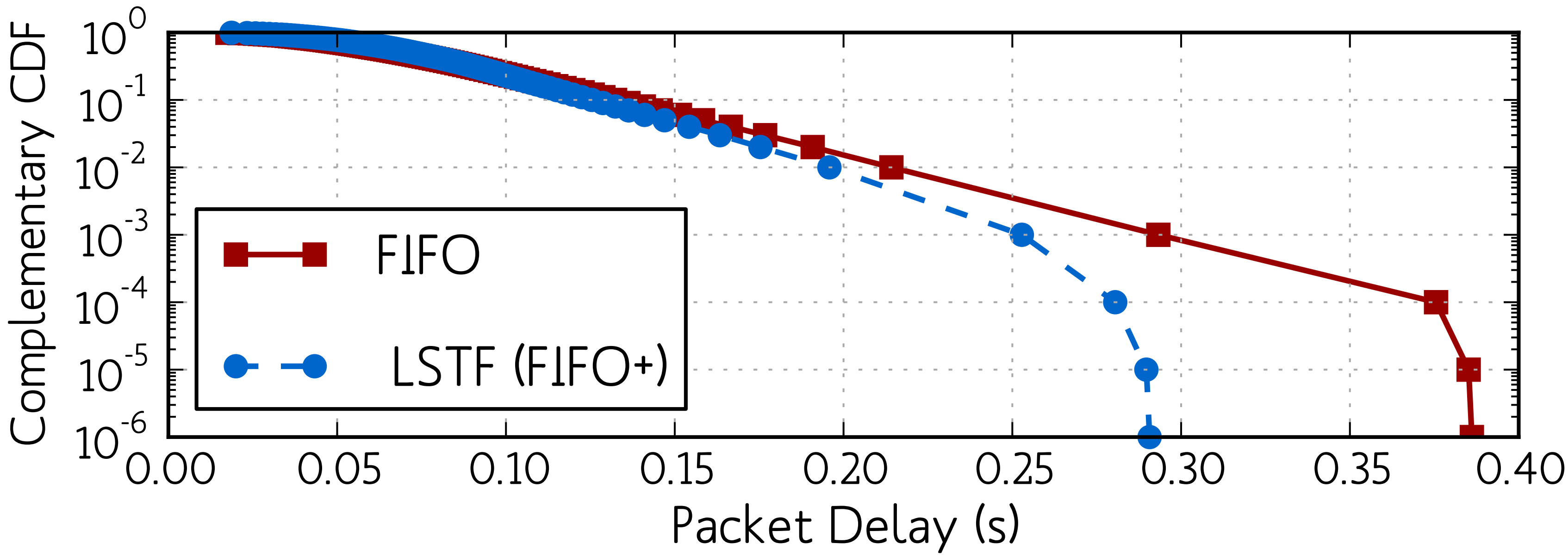
- Comparison with state-of-the-art

- Three objective functions
  - Tail packet delays
  - Mean Flow Completion Time
  - Fairness

Tail Packet Delays

Slack Assignment: Same slack for all packets

State-of-the-art: FIFO, FIFO+

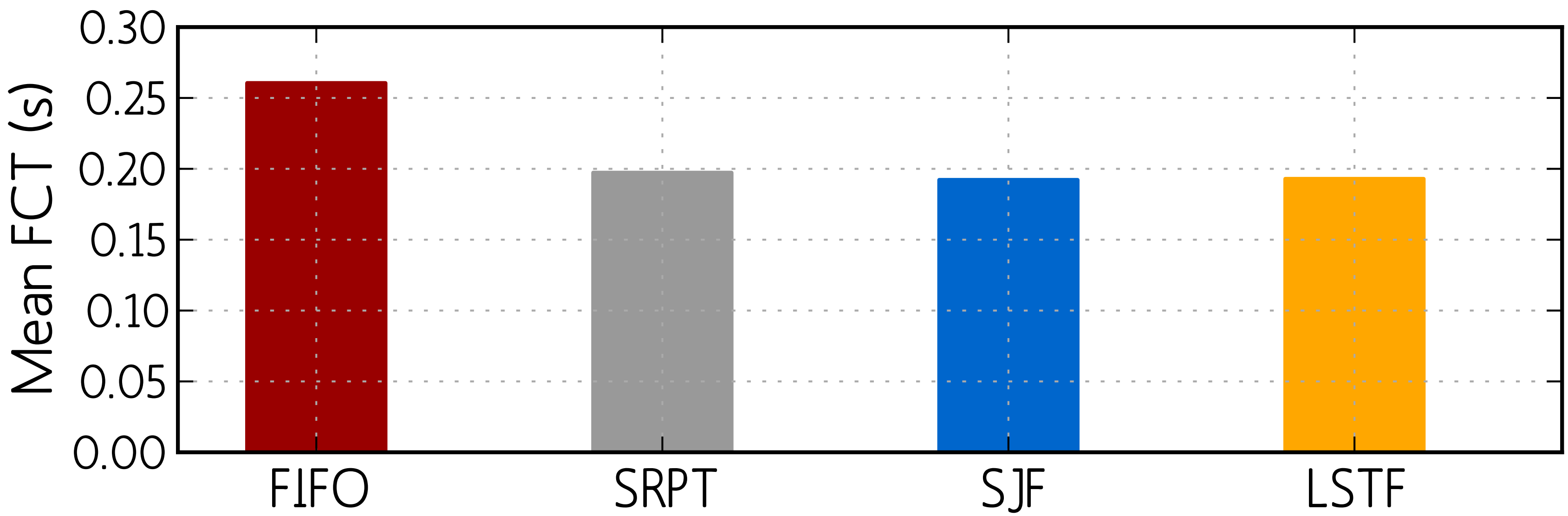


Smaller Tail Packet Delays with LSTF (FIFO+)

Mean Flow Completion Time

Slack Assignment: Proportional to flow size

State-of-the-art: SJF, SRPT



Mean FCT with LSTF comparable to SRPT and SJF

Fairness

Slack Assignment: Based on Virtual Clocks

State-of-the-art: Fair Queuing

Result: Eventual convergence to fairness

Results Summary

- Theoretical results show that
  - There is no UPS under blackbox initialization
  - LSTF comes as close to a UPS as possible
  - Empirically, LSTF is very close
- LSTF can be used in practice to achieve a variety of network-wide objectives

Implication

- Less need for many different scheduling algorithms.
- Can just use LSTF, with varying initializations.

There are still a bunch of open questions!

Open Questions

- What is the least amount of information needed to achieve universality?
- Are there tractable bounds for the degree of lateness with LSTF?
- What is the class of objectives that can be achieved with LSTF in practice?

Summary

- Theoretical results show that
  - There is no UPS under blackbox initialization
  - LSTF comes as close to a UPS as possible
  - Empirically, LSTF is very close
- LSTF can be used in practice to achieve a variety of network-wide objectives