On Fairness-Efficiency Tradeoffs for Multi-Resource Packet Processing

Wei Wang, Ben Liang, Baochun Li
Department of Electrical and Computer Engineering
University of Toronto
July 8, 2013

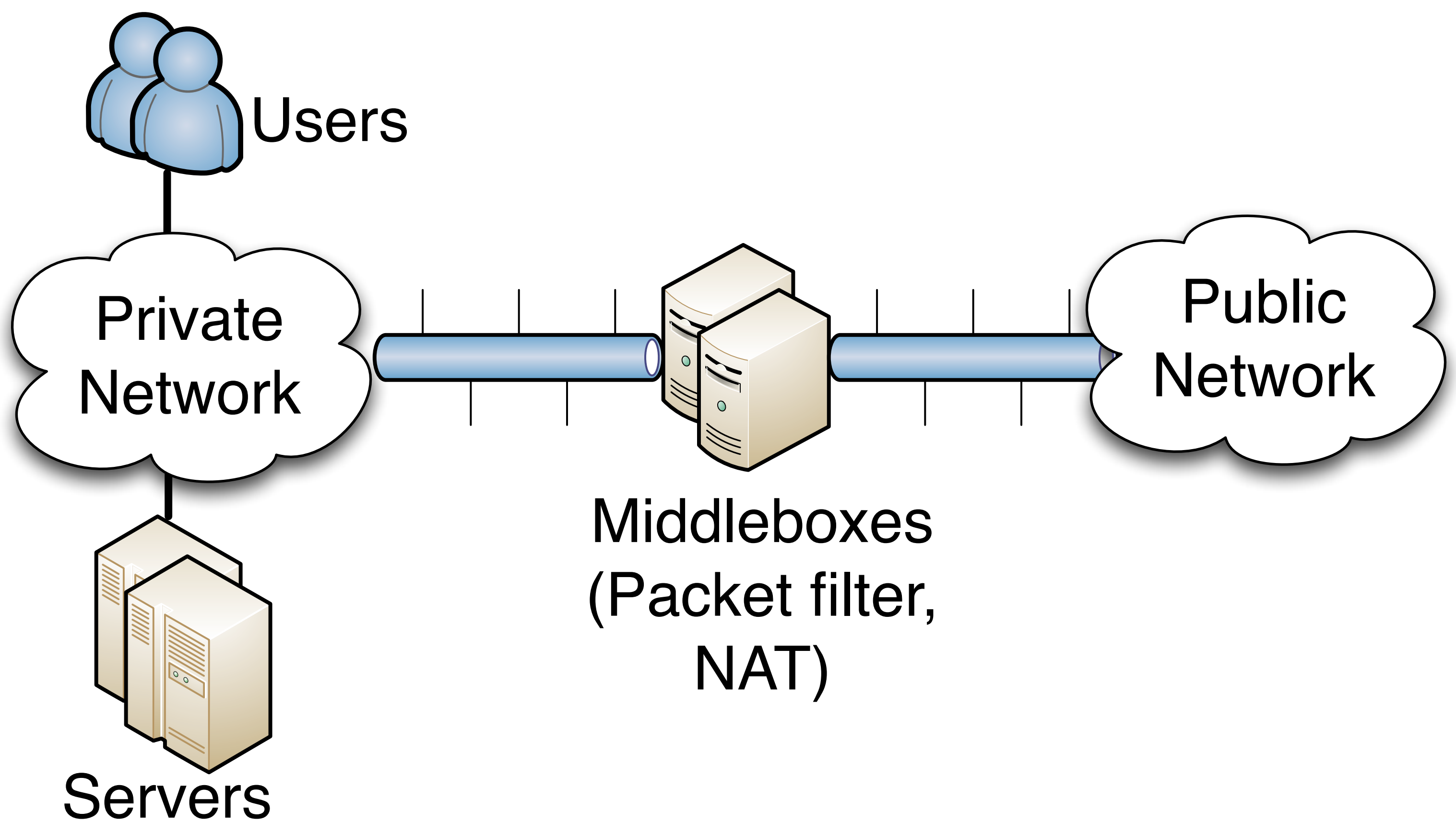
Background

Middleboxes (MBs) are ubiquitous in datacenter networks

- The sheer number is on par with the L2/L3 infrastructures

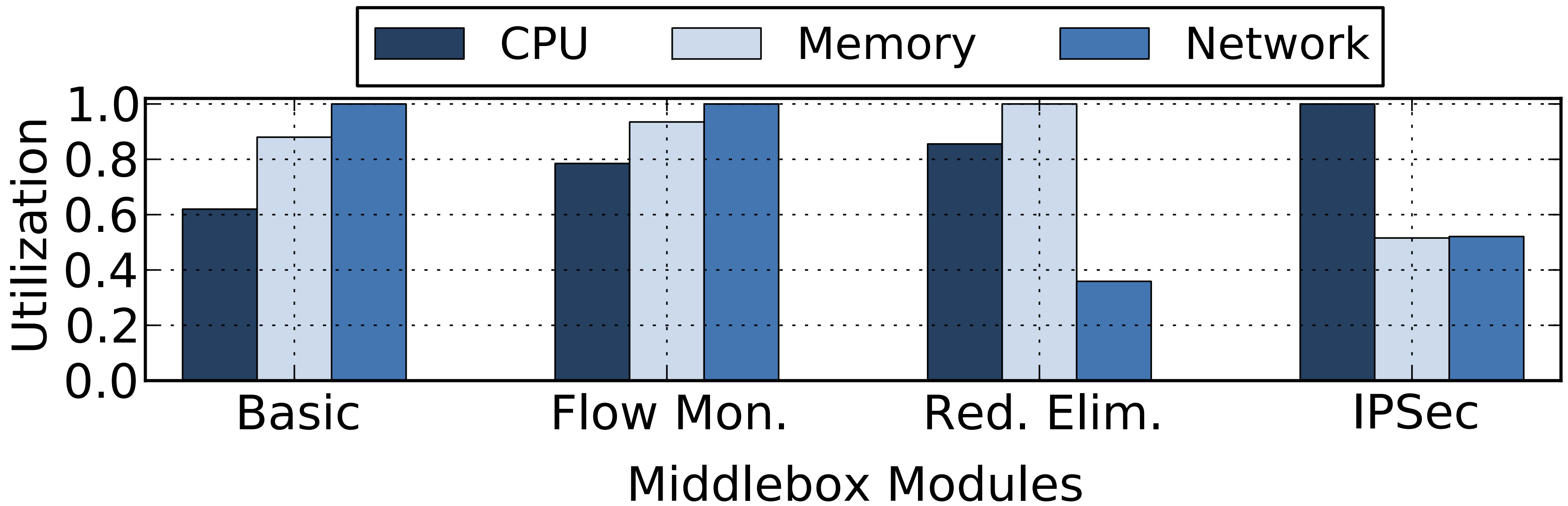
Perform a wide range of critical network functionalities

- WAN optimization, intrusion detection and prevention, etc.



Multi-resource packet processing in MBs

Performing different network functionalities requires different amounts of MB resources



Ghodsi et al SIGCOMM12

How to schedule traffic flows in a *fair* and *efficient* manner?

Fairness and efficiency

Dominant Resource Fairness

- Offer *predictable service isolation*
- Such service isolation is *independent* of other flows' behaviours

Efficiency

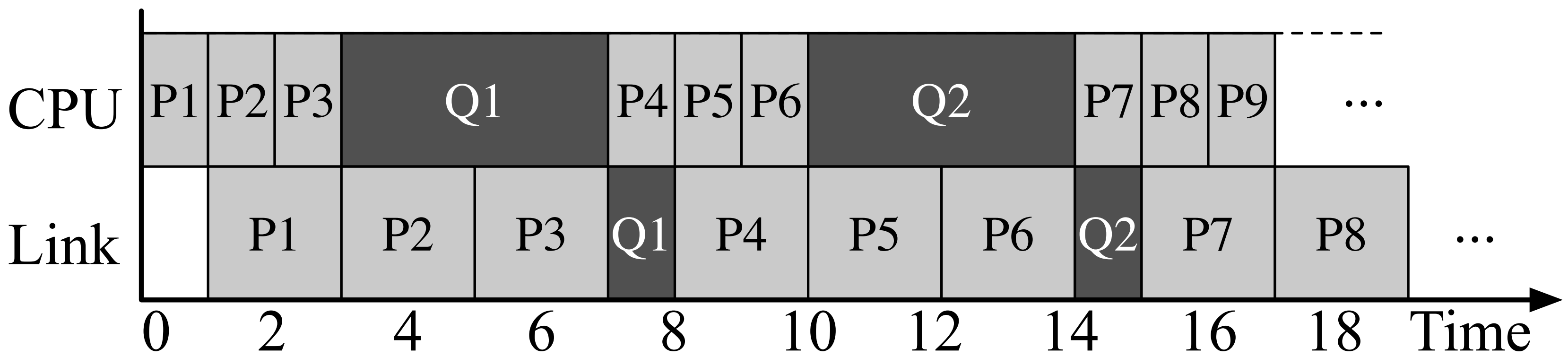
- Flows should finish their services as fast as possible
- High resource utilization

Ideally, we would like to have a scheduling algorithm that is both fair and efficient

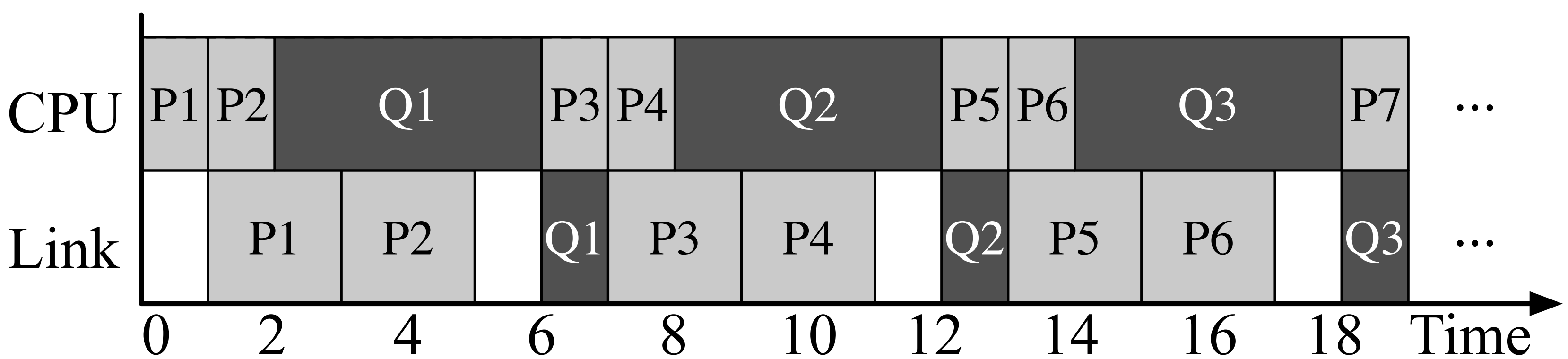
However...

Achieving both fairness and efficiency may not be possible when *multiple resources* are to be scheduled

There exists a fairness-efficiency tradeoff



(a) Packet scheduling that is efficient yet unfair.



(b) Packet scheduling that is fair yet inefficient.

A new research problem

Received little attention before in the fair queueing literature

- Traditional fair queueing has *only one resource* to schedule, i.e., output bandwidth
- Fairness-efficiency tradeoff does not exist!
- As long as the fair queueing algorithm is *work conserving*, it is the most efficient with the maximum resource utilization

Fairness-efficiency tradeoff has only been discussed in multi-resource allocations [Joe-Wong12]

- Multiple resources are concurrently shared among users *in space*
- In middleboxes, hardware resources are limited and have to be multiplexed by multiple flows *in time*

A new design concern

Fairness is not the only objective to pursue for resource scheduling

- Some applications may have a loose requirement on fairness, but emphasize more on efficiency and resource utilization
- We do not want to sacrifice efficiency for fairness in all cases

Allow a user to specify the tradeoff requirement and design a scheduling algorithm to implement it

How to express the fairness-efficiency tradeoff?

How to implement the specified tradeoff?

An initial step

We give two representations of the fairness-efficiency tradeoffs

- The first unifies both fairness and efficiency concerns into a unifying framework
- The second optimizes efficiency under the fairness constraints

We discuss their implementation issues and share some of our insights

The Fairness and Efficiency Measures

Dominant Resource Fairness (DRF)

Dominant resource

the one that requires the most processing time

A packet requires 1ms for CPU processing and 4 ms for link transmission

Link bandwidth is the dominant resource

DRF

Allocate equal processing time on the *dominant resources* of all backlogged flows

Fairness measure

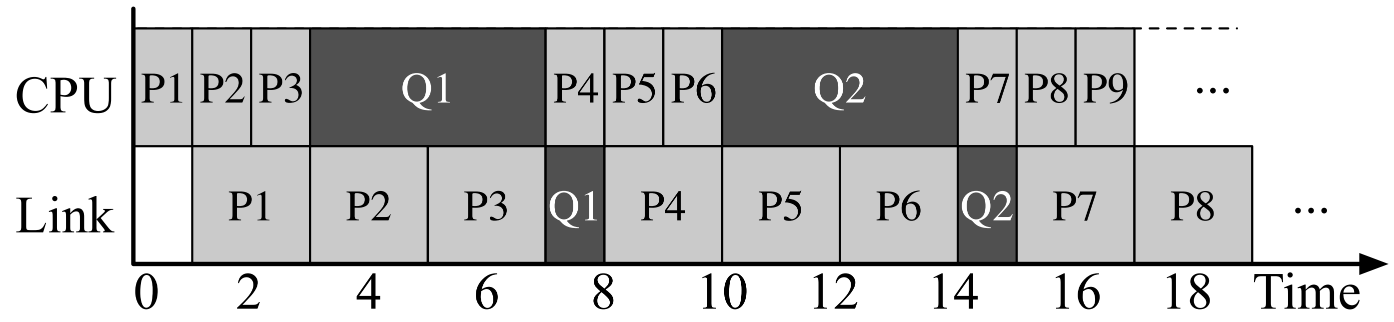
Relative fairness bound (RFB)

The gap of the processing time two flows receive on their dominant resources

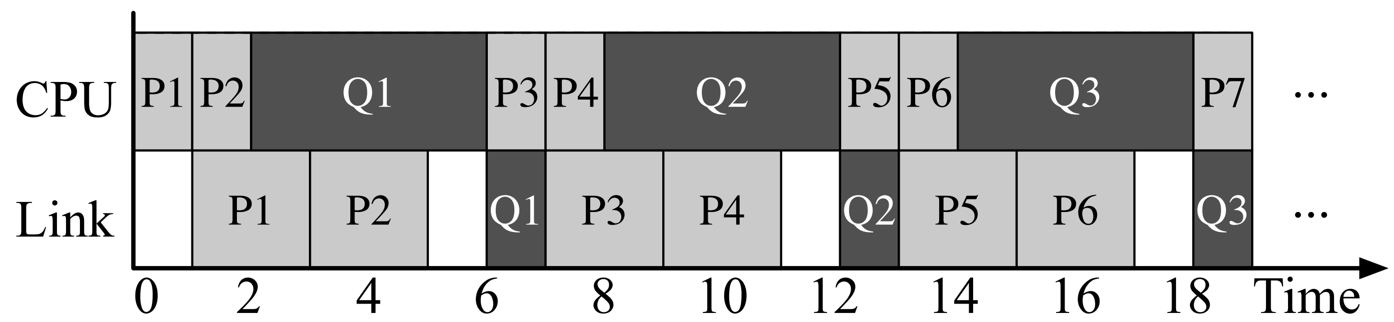
$$R = \sup_{t_1, t_2; i, j \in \mathcal{B}(t_1, t_2)} \left| \frac{T_i(t_1, t_2)}{w_i} - \frac{T_j(t_1, t_2)}{w_j} \right| .$$

The less RFB, the fairer the scheduler

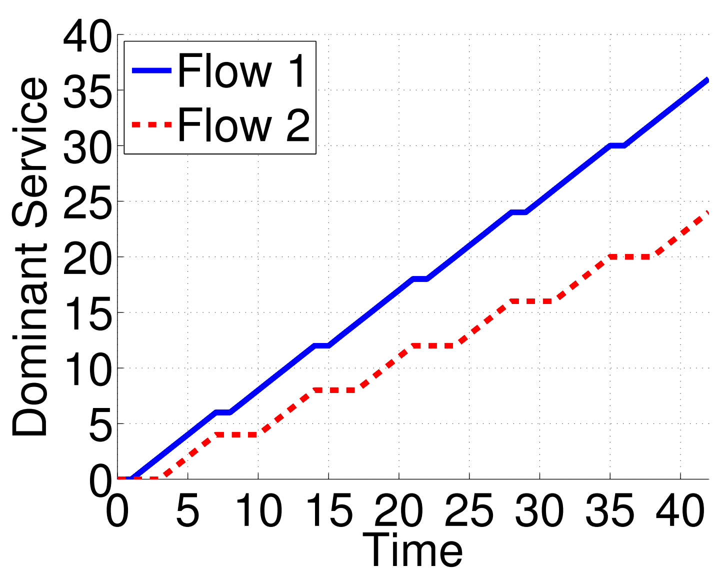
Fairness measure (cont'd)



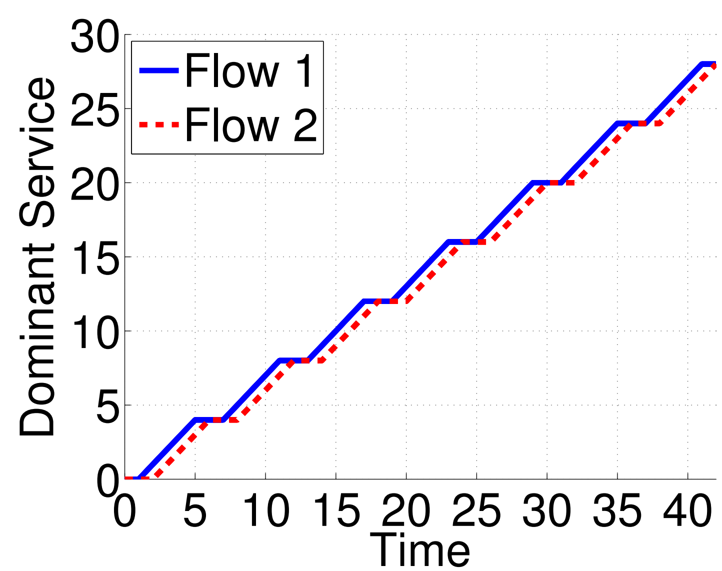
(a) Packet scheduling that is efficient yet unfair.



(b) Packet scheduling that is fair yet inefficient.



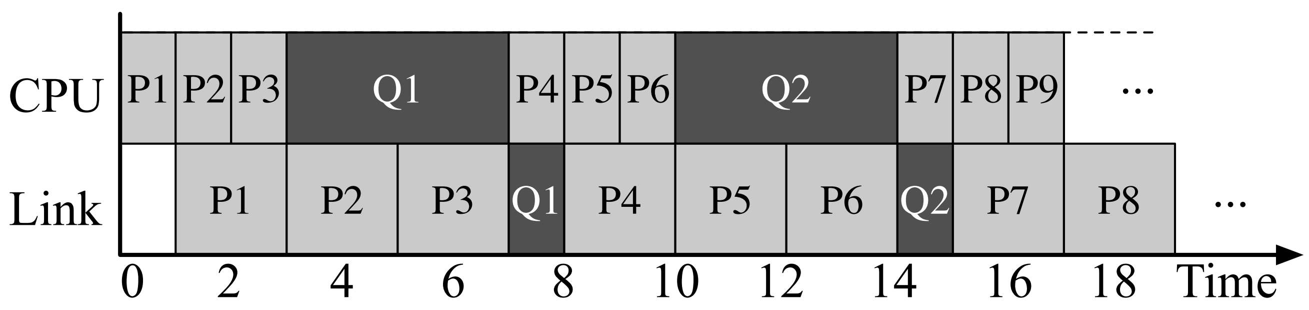
(a) Services recevied in Fig. 1a.



(b) Services recevied in Fig. 1b.

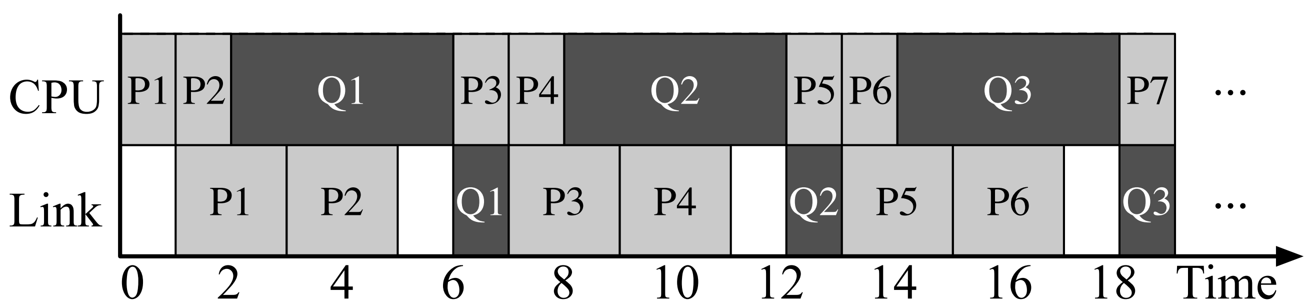
Efficiency as aggregate dominant services

The efficiency is measured as the *aggregate dominant services* that all flows receive



(a) Packet scheduling that is efficient yet unfair.

Aggregate Dominant Services: 10/7 t



(b) Packet scheduling that is fair yet inefficient.

Aggregate Dominant Services: 4/3 t

Efficiency gap: $(10/7 - 4/3)t$

Efficiency as the makespan

Makespan

The total time that is required to finish processing all packets

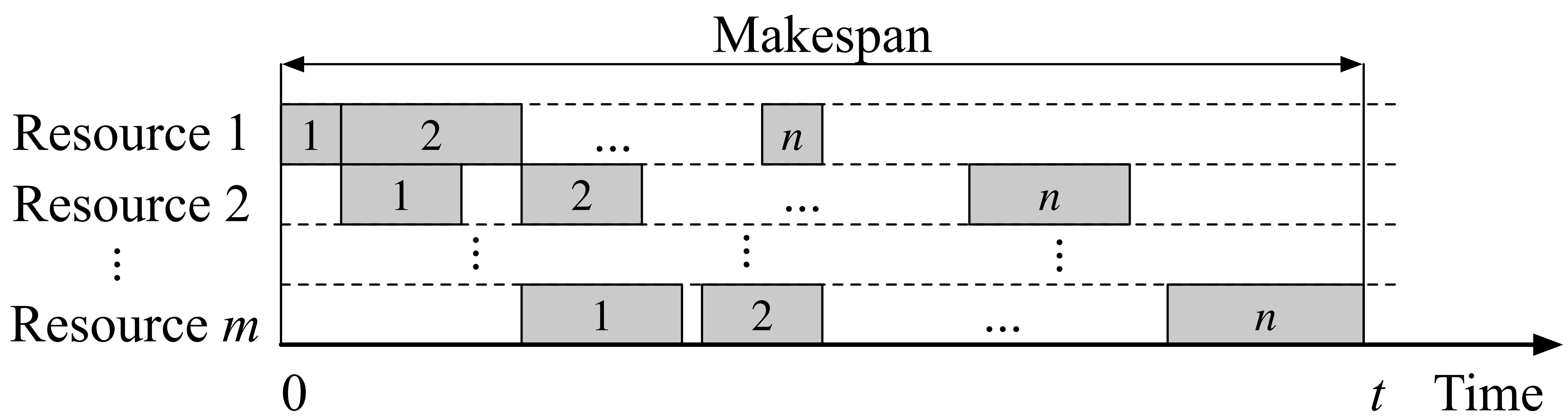


Fig. 4. Illustration of a makespan of a schedule serving n packets.

The Fairness-Efficiency Tradeoffs

Two tradeoff representations

Representing the tradeoff using a unifying framework

- Efficiency is measured as the aggregate dominant service
- More rigorous, but hard to implement

Representing the tradeoff as efficiency optimization under fairness constraints

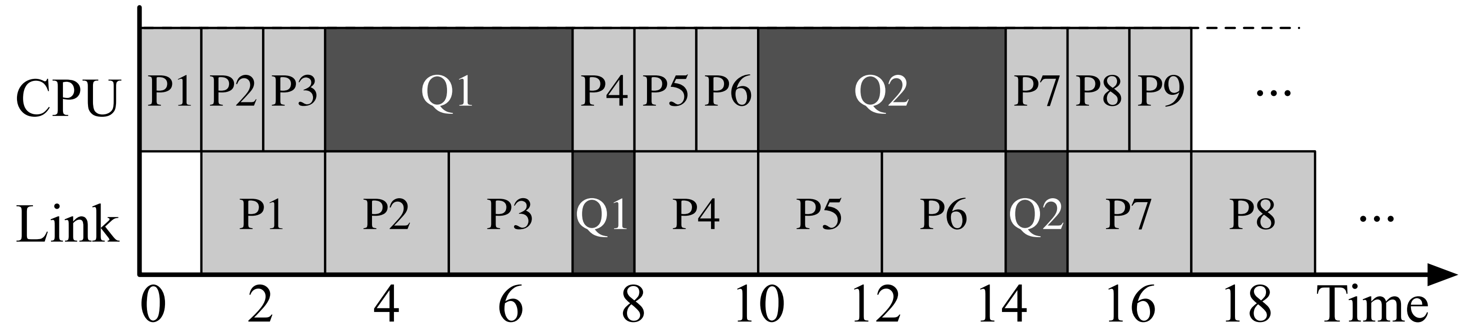
- Heuristic, but amenable to implement

Tradeoff using a unifying framework

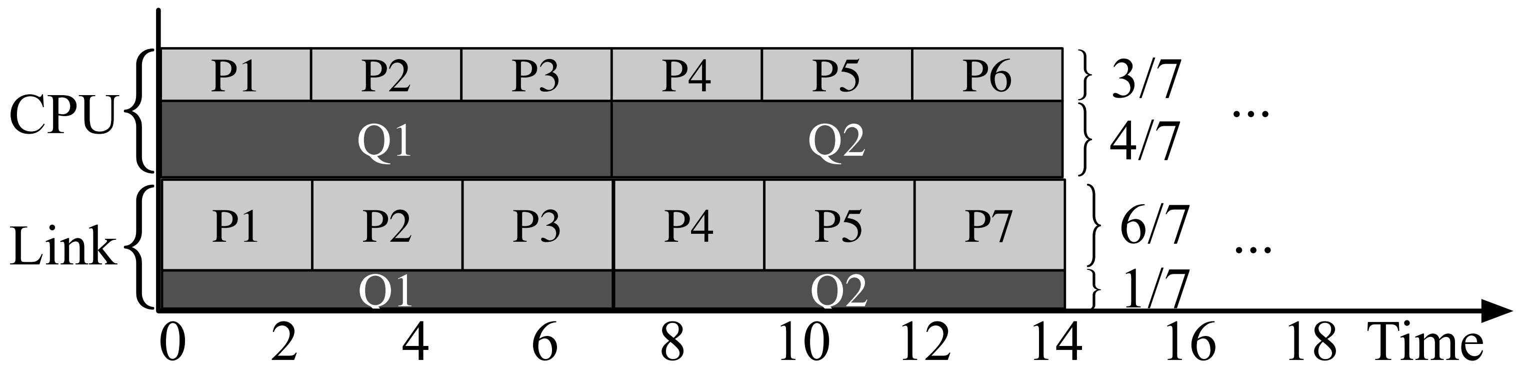
The idealized fluid flow model

Multi-Resource Fluid Model

- Flows are assumed to be served in *arbitrarily small increments*
- Multiple flows can be served *in parallel*



(a) Packet scheduling that is efficient yet unfair.



(a) The fluid version of the schedule shown in Fig. 1a.

Tradeoff using a unifying framework

Fairness-Efficiency Tradeoffs [Joe-Wong12]

x_i^t: the fraction of dominant resource allocated to flow *i* at time *t*

$\mathbf{x}^t = \langle x_1^t, \ldots, x_n^t \rangle$: the allocation vector

$$f_{\beta,\lambda}(\mathbf{x}^t) = \operatorname{sgn}(1-\beta) \left(\sum_{i \in \mathcal{B}(t)} \left(\frac{x_i^t}{\sum_{j \in \mathcal{B}(t)} x_j^t} \right)^{1-\beta} \right)^{\frac{1}{\beta}} \left(\sum_{i \in \mathcal{B}(t)} x_i^t \right)^{\lambda} .$$

Fairness — Efficiency

β: fairness parameter

λ: efficiency parameter

Tradeoff using a unifying framework (cont'd)

Idealized fluid model implements the specified tradeoff at all times

$$\begin{aligned} \max_{\mathbf{x}^t} \quad & f_{\beta,\lambda}(\mathbf{x}^t) \\ \text{s.t.} \quad & \sum_{i \in \mathcal{B}(t)} x_i^t \bar{\tau}_{i,r} \leq 1, \quad r = 1, 2, \ldots, m, \end{aligned}$$

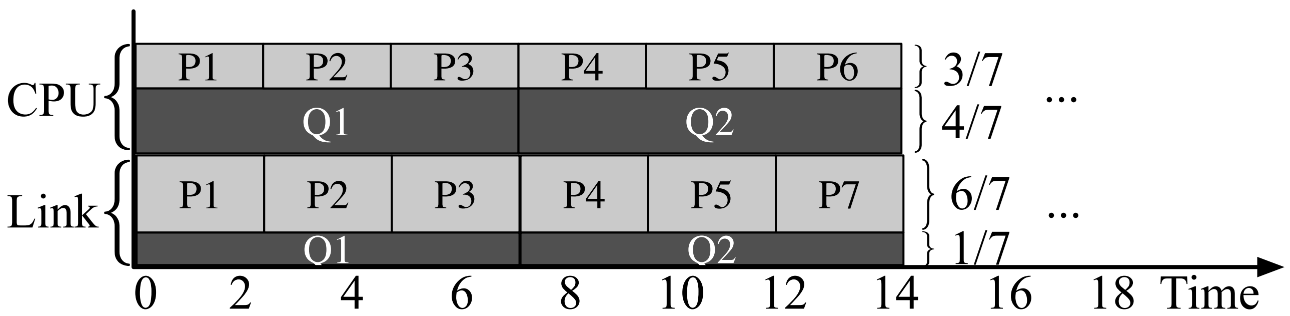
Packet-by-packet scheduling

Just like how GPS is approximated by WFQ

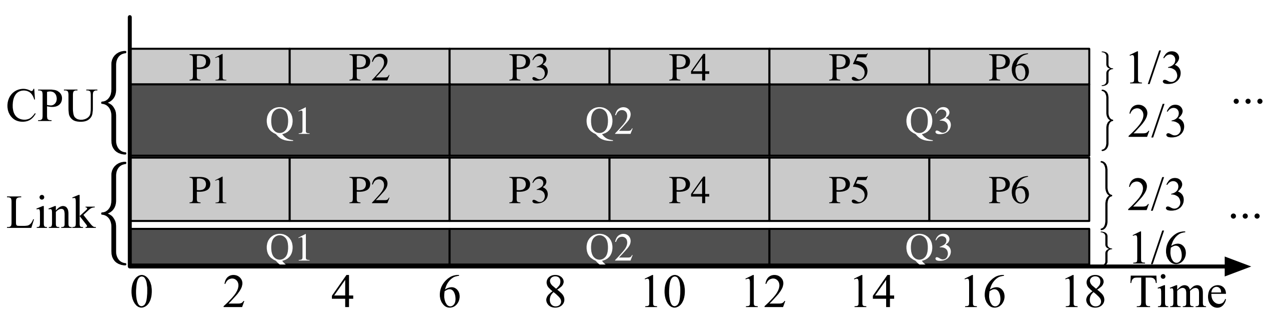
- Maintain an idealized fluid scheduling as a referencing system in background
- Whenever there is a scheduling opportunity, the packet that finishes its service the earliest in the referencing system is scheduled

Packet-by-packet scheduling (cont'd)

Referencing fluid system

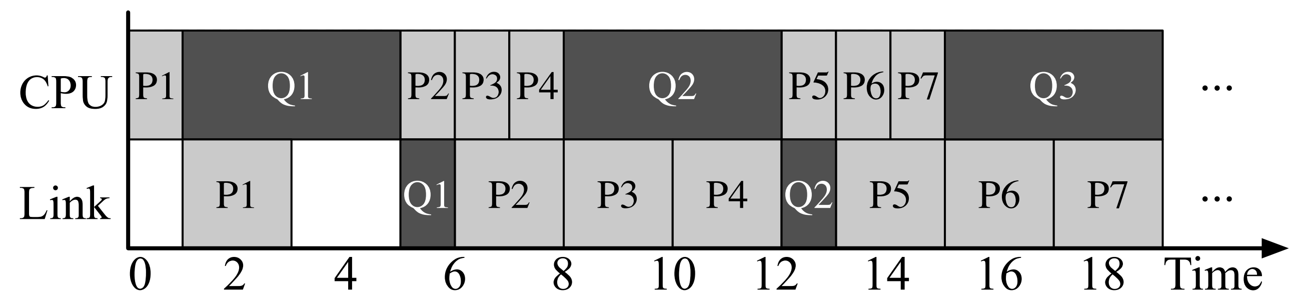


(a) The fluid version of the schedule shown in Fig. 1a.

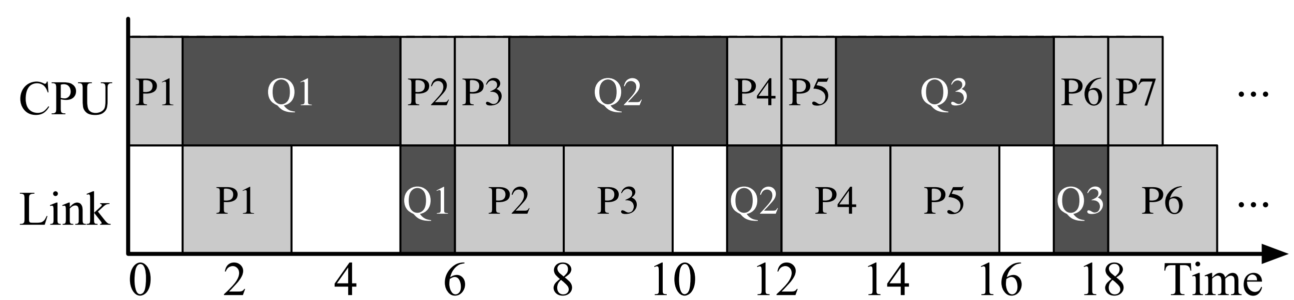


(b) The fluid version of the schedule shown in Fig. 1b.

Practical system



(a) Packet-by-packet approximation to the schedule of Fig. 7a.



(b) Packet-by-packet approximation to the schedule of Fig. 7b.

Open problem

Maintaining the fluid model requires high computational complexity

The optimization problem is generally non-convex

$$\max_{\mathbf{x}^t} \quad f_{\beta,\lambda}(\mathbf{x}^t)$$
$$\text{s.t.} \quad \sum_{i \in \mathcal{B}(t)} x_i^t \bar{\tau}_{i,r} \leq 1, \quad r = 1, 2, \ldots, m,$$

Tradeoff as a constrained optimization problem

Tradeoff as a constrained optimization Problem

Intuition

Schedule packets as quickly as possible, as long as the specified fairness requirement (RFB) is not violated

How can packets be scheduled "as quickly as possible"?

Minimize the makespan

Schedule packets as quickly as possible

Consider an extreme case where the efficiency (makespan) is the only objective to optimize

- Multi-stage flow shop problem

When there are two resources, e.g., CPU and bandwidth

- Johnson's algorithm is optimal

When there are more than two resources, the problem is NP-hard!

- Extending Johnson's algorithm offers a good heuristic

Implement fairness-efficiency tradeoff

The scheduler keeps track of the dominant services allocated to every traffic flow

As long as the service gap does not exceed the specified fairness requirement (RFB), we schedule packets using Johnson's heuristic for higher efficiency

Once the gap exceeds some threshold, the flow that receives the least dominant services will have the highest priority to be served, until the gap falls below the threshold

After that, the efficiency will become the primary concern

Conclusions

Unlike single-resource packet scheduling, there exists a tradeoff between fairness and efficiency

We raise attention to two important research problems

- How can fairness-efficiency tradeoff be expressed?
- How can a queueing scheme be designed to implement the specified tradeoff?

As an initial step

- We present two tradeoff representations for multi-resource packet scheduling
- We discuss their implementation issues

Thanks!

http://iqua.ece.toronto.edu/~weiwang/