Sharing Large Bags of Tasks in Heterogeneous NOWs: Greedier Is Not Better

Arnold L. Rosenberg
Department of Computer Science
University of Massachusetts
Amherst, Massachusetts, USA
`rsnbrg@cs.umass.edu`

The $\mathcal{TAPADS}$ Group
"where technology meets mathematics"

The Case for Mathematical Models

A well-crafted (= faithful and tractable) mathematical model of a real-life computing environment can help one:

1. hone one's intuition
   - Which features of the environment impact one's computational efficiency?
   - Precisely how do they impact the computation?

2. verify — or refute — one's intuition
   - *"Sometimes I even think that what I know's not so!"*

     The King and I

3. put items 1 and 2 together to craft efficient algorithms.

Two Work-Sharing Problems

The Computational Environment

- A "master" workstation P_0
- A NOW $\mathcal{N}$ of n *heterogeneous* workstations

  $$P_1,\ P_2,\ \ldots,\ P_n$$

  that are available for "rental"
- a large bag of equally complex tasks

The NOW-Exploitation Problem

- One has access to $\mathcal{N}$ for L time units.
- One wants to accomplish as much work as possible during that time.

The NOW-Rental Problem

- One has W units of work to complete.
- One wishes to "rent" $\mathcal{N}$ for as short a period of time as is necessary to complete that work.

Our Contributions

Within a *heterogeneous, long-message* analogue of the LogP model, we offer

A Generic Work-Sharing Protocol

- The Protocol is ***robust:***
  - It works predictably for many variants of our model.
- The Protocol is ***self-scheduling:*** It determines:
  - all work-allocations to the "rented" workstations;
  - all communication times.

A Myth-Dispelling Analysis of Work-Sharing Protocols

- We present two provably efficient solutions to the NOW-Exploitation and -Rental problems.
  - *The nongreedy solution outperforms the greedy one.*

The Model, 1

Calibration

- All units — time and packet size — are calibrated to workstation P_0's computation rate:
  - *P_0 does one "unit" of work in one "unit" of time.*
- Each unit of work produces δ units of results (for simplicity).

Computation Rates

$\rho_i \stackrel{\text{def}}{=}$ per-unit work time for workstation P_i

- $\rho_0 = 1$ (by our calibration)
- $\rho_1 \ \leq \ \rho_2 \ \leq \ \cdots \ \leq \ \rho_n$ (by convention)

$\boxed{\textbf{The Model, 2}}$

The Costs of Communication, 1

Message Processing time for P_i:

Transmission setup:	σ_{ij} time units *per message* to P_j
Transmission packaging:	π_i time units *per packet*
Reception unpackaging:	$\overline{\pi}_i$ time units *per packet*

- Subscripts reflect *workstations' heterogeneity*.

Message Transmission Time:

Latency:	λ time units for *first packet*
Bandwidth limitation:	$\tau \stackrel{\text{def}}{=} 1/\beta$ time units/packet for *remaining packets*

- $\beta \stackrel{\text{def}}{=}$ network's *end-to-end bandwidth*.

The Costs of Communication, 2

The "bottom line":

For a p-packet message from P_i to P_j:

Processing by P_i:	$\sigma_{ij} + \pi_i p$ time units
Pipelined transmission:	$\lambda + (p-1)\tau$ time units
Nonpipelined transmission:	λp time units
Processing by P_j:	$\overline{\pi}_j p$ time units

An Added Constraint:

Network capacity:	$\leq \kappa$ packets can simultaneously be in transit in the network

The Communication Regimen

- The model uses a *strict* single-port communication regimen

  $\rightarrow$ No two messages can simultaneously be in transit $\leftarrow$

  but our conclusions hold for many relaxations of this regimen.

P_0 prepares work for P_i	$P_0 \leftrightarrow P_i$ setup	P_0 transmits work	P_i unpacks work	P_i does work	P_i prepares results for P_0	$P_i \leftrightarrow P_0$ setup	P_i transmits results	P_0 unpacks results
$\pi_0 \; w_i$	$\sigma_{0\,i}$	λ $\tau(w_i - 1)$	$\bar{\pi}_i \, w_i$	$\rho_i \, w_i$	$\pi_i \, \delta \, w_i$	$\sigma_{i\,0}$	λ $\tau(\delta w_i - 1)$	$\bar{\pi}_0 \, \delta \, w_i$
in P_0	in P_0 , P_i and network	in network	in P_i			in P_0 , P_i and network	in network	in P_0

A Generic Work-Sharing Protocol

Specifying a worksharing protocol

- P_0 sends work to $P_1, P_2, \ldots, P_n$ in the order

  $P_{s_1}, P_{s_2}, \ldots, P_{s_n}$
- $P_1, P_2, \ldots, P_n$ return results to P_0 in the order

  $P_{f_1}, P_{f_2}, \ldots, P_{f_n}$

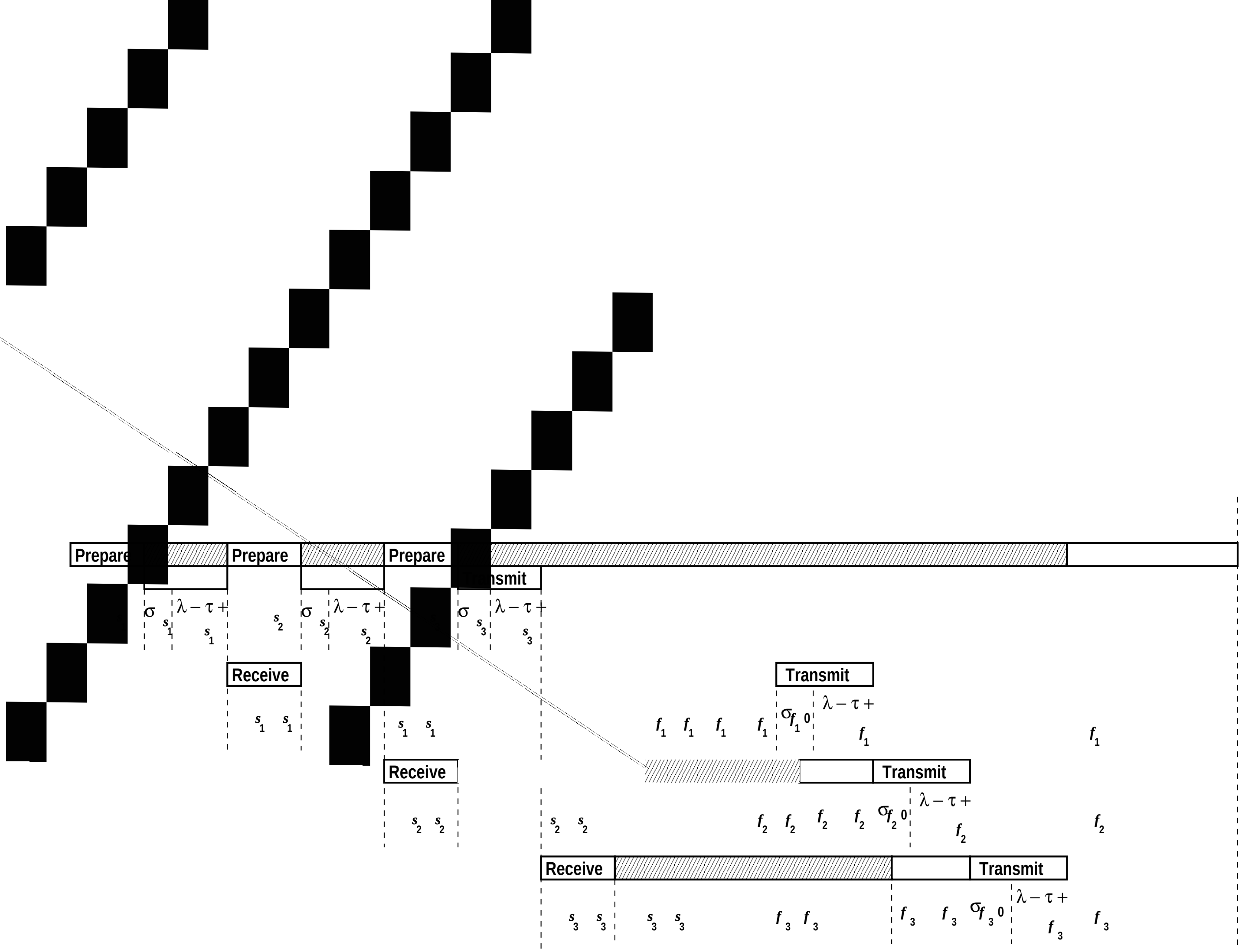
We thus have *three indexings* of the "rented" workstations:

the *power-related indexing:*	$1, 2, \ldots, n$
the *startup indexing:*	$s_1, s_2, \ldots, s_n$
the *finishing indexing:*	$f_1, f_2, \ldots, f_n$

The latter two specify each specific worksharing protocol.

Some useful abbreviations:

	Quantity	Meaning
$\tilde{\tau}$	$\tau(1+\delta)$	2-way network transmission rate
$\widetilde{\pi}_i$	$\overline{\pi}_i + \pi_i\delta$	P_i's 2-way message-packaging rate
FC_i	$(\sigma_{0i} + \sigma_{i0}) + 2(\lambda - \tau)$	P_i's *fixed* comm. overhead
VC_i	$\pi_0 + \tilde{\tau} + \widetilde{\pi}_i$	P_i's *variable* comm. overhead rate



Prepare
Prepare
Prepare
Receive
Receive
Receive
Transmit
Transmit
Transmit

$\boxed{\textbf{The Generic Protocol's Work-Allocations}}$

$$\begin{pmatrix} \mathsf{VC}_1+\rho_1 & B_{1,2} & \cdots & B_{1,n-1} & B_{1,n} \\ B_{2,1} & \mathsf{VC}_2+\rho_2 & \cdots & B_{2,n-1} & B_{2,n} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ B_{n-1,1} & B_{n-1,2} & \cdots & \mathsf{VC}_{n-1}+\rho_{n-1} & B_{n-1,n} \\ B_{n,1} & B_{n,2} & \cdots & B_{n,n-1} & \mathsf{VC}_n+\rho_n \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_{n-1} \\ w_n \end{pmatrix}$$

$$= \begin{pmatrix} L-\mathsf{FC}_1-c_1(\lambda-\tau)-\sum_{j\in\mathsf{SB}_1}\sigma_{0j}-\sum_{j\in\mathsf{FA}_1}\sigma_{j0} \\ L-\mathsf{FC}_2-c_2(\lambda-\tau)-\sum_{j\in\mathsf{SB}_2}\sigma_{0j}-\sum_{j\in\mathsf{FA}_2}\sigma_{j0} \\ \vdots \\ L-\mathsf{FC}_{n-1}-c_{n-1}(\lambda-\tau)-\sum_{j\in\mathsf{SB}_{n-1}}\sigma_{0j}-\sum_{j\in\mathsf{FA}_{n-1}}\sigma_{j0} \\ L-\mathsf{FC}_n-c_n(\lambda-\tau)-\sum_{j\in\mathsf{SB}_n}\sigma_{0j}-\sum_{j\in\mathsf{FA}_n}\sigma_{j0} \end{pmatrix}$$

The $B_{i,j}$ are specified as follows:

Procedure Fill_In_Coefficients

begin

1. Set all $B_{i,j} \;=\; 0$
2. for $i \in \{1, 2, \ldots, n\}$
   for $j \in \{1, 2, \ldots, n\} - \{i\}$
   (a) if $j \in \mathsf{SB}_i$ then $B_{i,j} \;=\; B_{i,j} + \pi_0 + \tau$;
   (b) if $j \in \mathsf{FA}_i$ then $B_{i,j} \;=\; B_{i,j} + \tau\delta$

end

Robust Protocols Are Self-Scheduling

Theorem. *Every robust worksharing protocol is* self-scheduling*:*

The protocol's startup and finishing indexings determine:

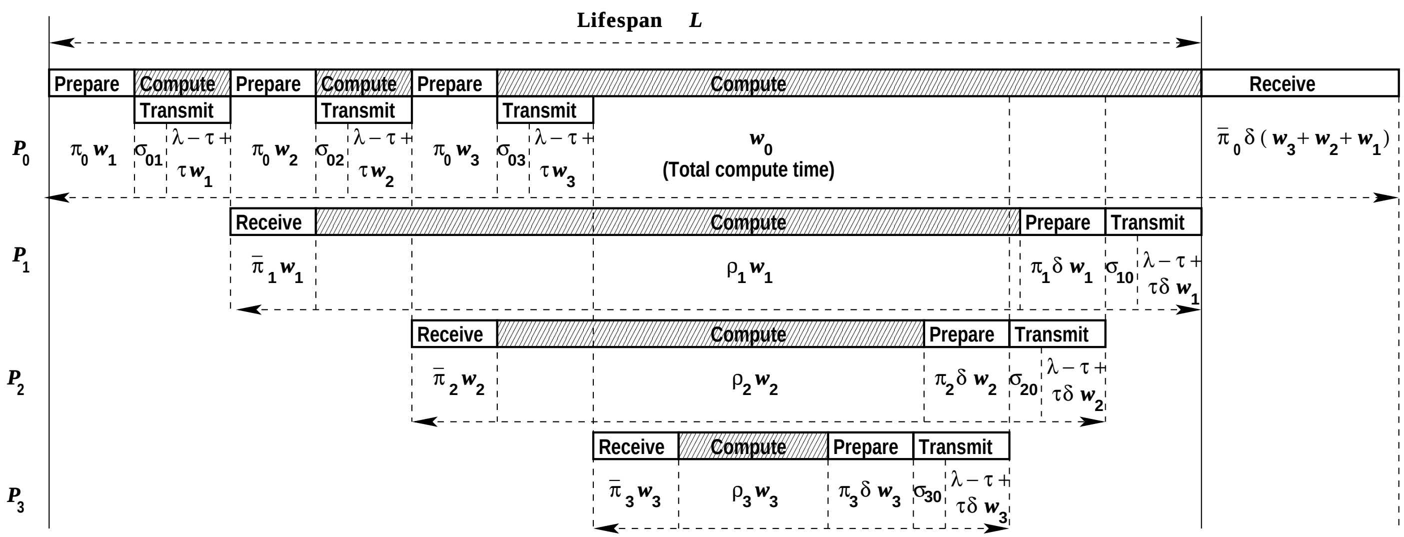
- *all work-allocations*
- *the times for all communications.*

Proof Sketch. In the system of linear equations specifying a robust protocol's work-allocations, *the matrix of coefficients is nonsingular.*

$\approx\approx\approx\approx\approx\approx\approx\approx$

We now focus on two special work-sharing protocols

- One epitomizes the greedy allocation of work to faster workstations.
- One moderates greed by a bit of balancing/fairness.



Lifespan L
Prepare
Compute
Transmit
Receive
P0
P1
P2
P3
w0
(Total compute time)

$\boxed{\text{The LIFO Work-Sharing Protocol}}$

The defining startup and finishing orderings:

$$\boxed{\text{For each } i \in \{1, 2, \ldots, n\} : \quad s_i = i \;\text{ and }\; f_i = n - i + 1.}$$

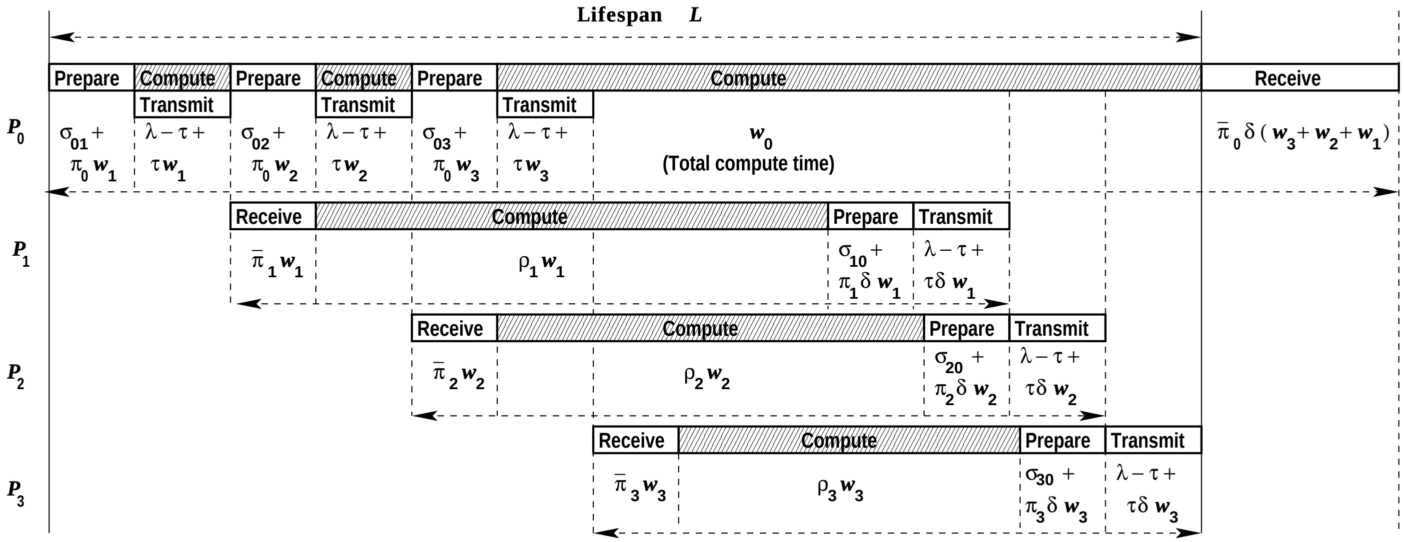
This epitomizes a greedy *allocation strategy:*

1. Allocate as much work as possible to the fastest workstation.
2. Within the constraints of (1), allocate as much work as possible to the second fastest workstation.
3. . . . and so on

The LIFO Protocol's Work-Allocations:

$$\begin{pmatrix} \mathsf{VC}_1 + \rho_1 & 0 & \cdots & 0 & 0 \\ \pi_0 + \tilde{\tau} & \mathsf{VC}_2 + \rho_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \pi_0 + \tilde{\tau} & \pi_0 + \tilde{\tau} & \cdots & \mathsf{VC}_{n-1} + \rho_{n-1} & 0 \\ \pi_0 + \tilde{\tau} & \pi_0 + \tilde{\tau} & \cdots & \pi_0 + \tilde{\tau} & \mathsf{VC}_n + \rho_n \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_{n-1} \\ w_n \end{pmatrix}$$

$$= \begin{pmatrix} L - \mathsf{FC}_1 \\ L - (\mathsf{FC}_1 + \mathsf{FC}_2) \\ \vdots \\ L - \Sigma_{i=1}^{n-1} \mathsf{FC}_i \\ L - \Sigma_{i=1}^{n} \mathsf{FC}_i \end{pmatrix}$$



Lifespan L
Prepare
Compute
Transmit
Receive
P0
P1
P2
P3
σ01 + π0 w1
λ − τ + τ w1
σ02 + π0 w2
λ − τ + τ w2
σ03 + π0 w3
λ − τ + τ w3
w0
(Total compute time)
π̄0 δ (w3 + w2 + w1)
π̄1 w1
ρ1 w1
σ10 + π1 δ w1
λ − τ + τδ w1
π̄2 w2
ρ2 w2
σ20 + π2 δ w2
λ − τ + τδ w2
π̄3 w3
ρ3 w3
σ30 + π3 δ w3
λ − τ + τδ w3

$\boxed{\textbf{The FIFO Work-Sharing Protocol}}$

The defining startup and finishing orderings:

$\boxed{\text{For each } i \in \{1, 2, \ldots, n\} : \quad s_i = \ f_i = i.}$

Greed mollified by fairness
(In fact, the greed is not so obvious.)

The FIFO Protocol's Work-Allocations:

$$\begin{pmatrix} \mathsf{VC}_1 + \rho_1 & \tau\delta & \cdots & \tau\delta & \tau\delta \\ \pi_0 + \tau & \mathsf{VC}_2 + \rho_2 & \cdots & \tau\delta & \tau\delta \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \pi_0 + \tau & \pi_0 + \tau & \cdots & \mathsf{VC}_{n-1} + \rho_{n-1} & \tau\delta \\ \pi_0 + \tau & \pi_0 + \tau & \cdots & \pi_0 + \tau & \mathsf{VC}_n + \rho_n \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_{n-1} \\ w_n \end{pmatrix}$$

$$= \begin{pmatrix} L - (n+1)(\lambda - \tau) - \sigma_{01} - \Sigma_{i=1}^{n} \sigma_{i0} \\ L - (n+1)(\lambda - \tau) - (\sigma_{01} + \sigma_{02}) - \Sigma_{i=2}^{n} \sigma_{i0} \\ \vdots \\ L - (n+1)(\lambda - \tau) - \Sigma_{i=1}^{n-1} \sigma_{0i} - (\sigma_{n-1,0} + \sigma_{n,0}) \\ L - (n+1)(\lambda - \tau) - \Sigma_{i=1}^{n} \sigma_{0i} - \sigma_{n,0} \end{pmatrix}$$

Greedy Is Not Optimal, 1

Theorem. *For all sufficiently long lifespans L:*

IF

- *the "rented" NOW is homogeneous (all ρ_i are equal)*

- *the first "rented" workstation is fast, in the sense that*

$$\rho_1 \ \le \ \frac{1}{1+\widetilde{\pi}_0}\left(\frac{L}{W^{(\mathrm{FIFO})}} \ - \ (\pi_0 + \tilde{\tau})\right),$$

THEN

the FIFO Protocol outperforms the LIFO Protocol

$$W^{(\mathrm{FIFO})} > W^{(\mathrm{LIFO})}$$

during a lifespan-L worksharing opportunity.

Greedy Is Not Optimal, 2

Proof Sketch. Note first that, as L grows without bound:

Asymptotic Approximation #1:

$$W^{(\mathrm{LIFO})} \;\to\; \left[\sum_{i=1}^{n} \frac{1}{\mathsf{VC}_i + \rho_i} \prod_{k=1}^{i-1} \left(1 - \frac{\pi_0 + \tilde{\tau}}{\mathsf{VC}_k + \rho_k}\right)\right] \cdot L$$

Asymptotic Approximation #2:

$$W^{(\mathrm{FIFO})} \;\to$$

$$\left[\sum_{i=1}^{n} \frac{1}{\mathsf{VC}_i + \rho_i - \tau\delta} \prod_{k=1}^{i-1} \left(1 \;-\; \frac{\pi_0 + \tau - \tau\delta}{\mathsf{VC}_k + \rho_k - \tau\delta}\right)\right] \cdot \left(L \;-\; \tau\delta W^{(\mathrm{FIFO})}\right)$$

Analyzing the Approximations:

Now compare the preceding sums term by term to complete the proof.

When does $W^{(\mathrm{FIFO})}$ overtake $W^{(\mathrm{LIFO})}$?, 1

How relevant are the theorem's asymptotics to real computations? —as a function of:

1. the size n of the "rented" NOW $\mathcal{N}$
2. the "degree of heterogeneity" of the NOW $\mathcal{N}$:
   —as exposed by the vector of power rates: $\langle \rho_1, \rho_2, \ldots, \rho_n \rangle$
3. the (non)pipelineability of $\mathcal{N}$'s network
4. the granularity of the tasks comprising our workload.

An Experimental Setup

Give all parameters "reasonable" values:

Parameter	Wall-Clock Time/Rate
Setup time σ	300 μsec
Latency λ	150 μsec
Transit rate τ (pipelined network)	1 μsec /work unit
Transit rate τ (nonpipelined network)	150 μsec /work unit
Packaging rate π_0	10 μsec /work unit

Set δ to 1 (for definiteness).

When does $W^{(\text{FIFO})}$ overtake $W^{(\text{LIFO})}$?, 2

Pipe-lined?	Power-Rate Vector $\langle \rho_i \rangle$	Task Grain	By what duration L is $W^{(\text{FIFO})} > W^{(\text{LIFO})}$? $n = 8$	$n = 32$	$n = 128$
YES	$\rho_i \equiv 1$	.1 sec	1 minute	1 minute	1 minute
		1 sec	1 minute	1 minute	1 minute
		10 sec	1 minute	1 minute	1 minute
	$(1 + 2^{i-n})/2$	.1 sec	1.59 hours	6.15 minutes	1 minute
		1 sec	6.95 days	10.6 hours	$37\frac{1}{2}$ minutes
		10 sec	31.8 years	44 days	69.45 hours
	$1 - 1/(i+1)$	.1 sec	2.39 hours	31 minutes	3.5 minutes
		1 sec	10.42 days	2.15 days	6.39 hours
		10 sec	318 years	$4\frac{1}{3}$ years	27.8 days
	$1 - 2^{-i}$	.1 sec	3.62 hours	26 minutes	$1\frac{1}{2}$ minutes
		1 sec	17.37 days	44 hours	3.12 hours
		10 sec	318 years	1.6 years	$13\frac{1}{3}$ days
NO	$\rho_i \equiv 1$	.1 sec	1 minute	1 minute	1 minute
		1 sec	1 minute	1 minute	1 minute
		10 sec	1 minute	1 minute	1 minute
	$(1 + 2^{i-n})/2$	.1 sec	1 minute	1 minute	1 minute
		1 sec	3 minutes	1 minute	1 minute
		10 sec	4.87 hours	20 minutes	1.07 minutes
	$1 - 1/(i+1)$	.1 sec	1 minute	1 minute	1 minute
		1 sec	4.5 minutes	1 minute	1 minute
		10 sec	7.37 hours	94 minutes	11.5 minutes
	$1 - 2^{-i}$	.1 sec	1 minute	1 minute	1 minute
		1 sec	$6\frac{1}{2}$ minutes	1 minute	1 minute
		10 sec	$10\frac{5}{6}$ hours	79 minutes	$5\frac{1}{2}$ minutes