Breaking the Valiant Load Balancing Barrier for Oblivious Reconfigurable Networks

Tegan Wilson

Cornell ⟶ Northeastern

Daniel Amir

Cornell

Nitika Saran

Cornell

Robert Kleinberg

Cornell

Vishal Shrivastav

Purdue

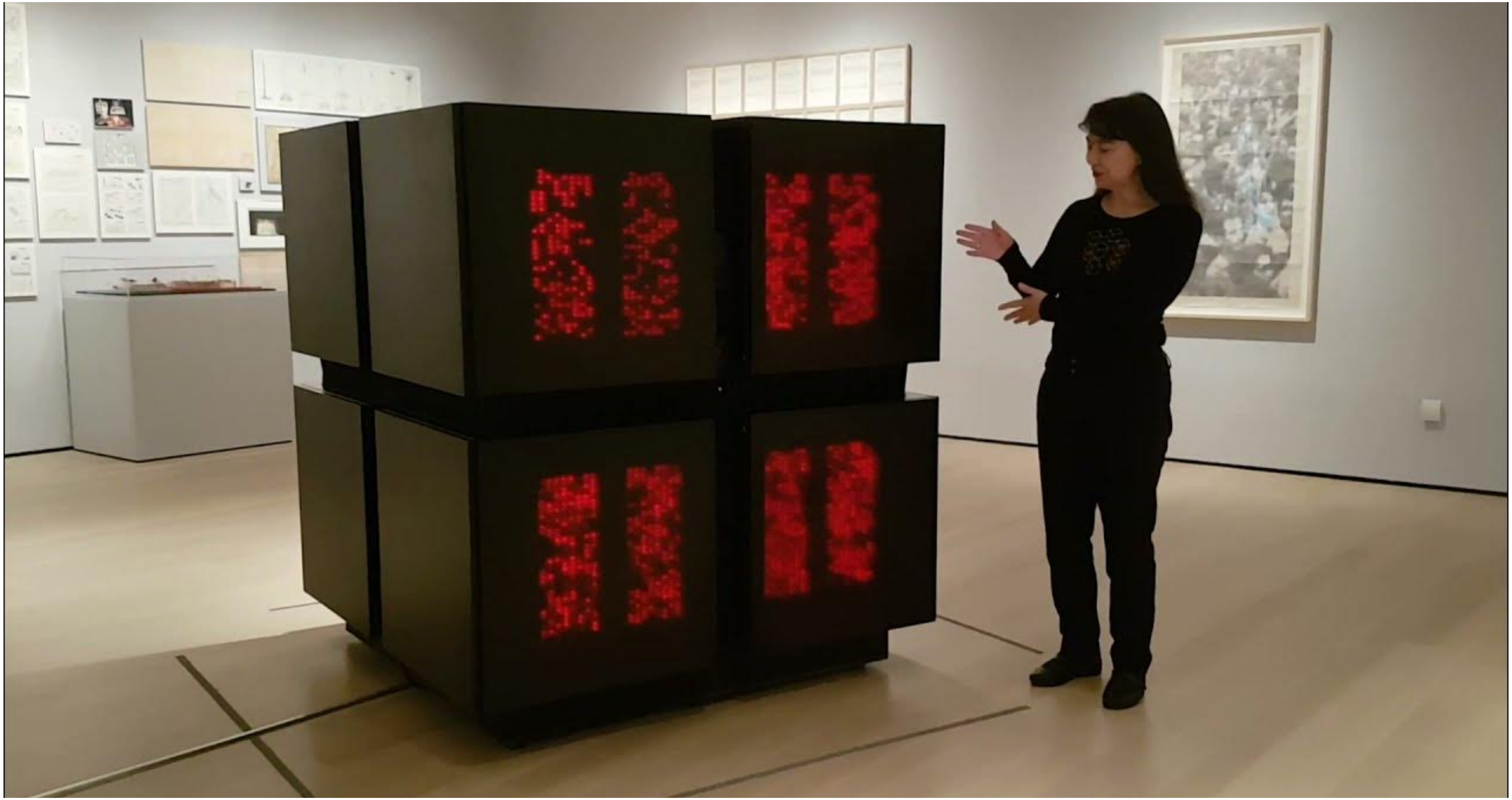
Hakim Weatherspoon

Cornell

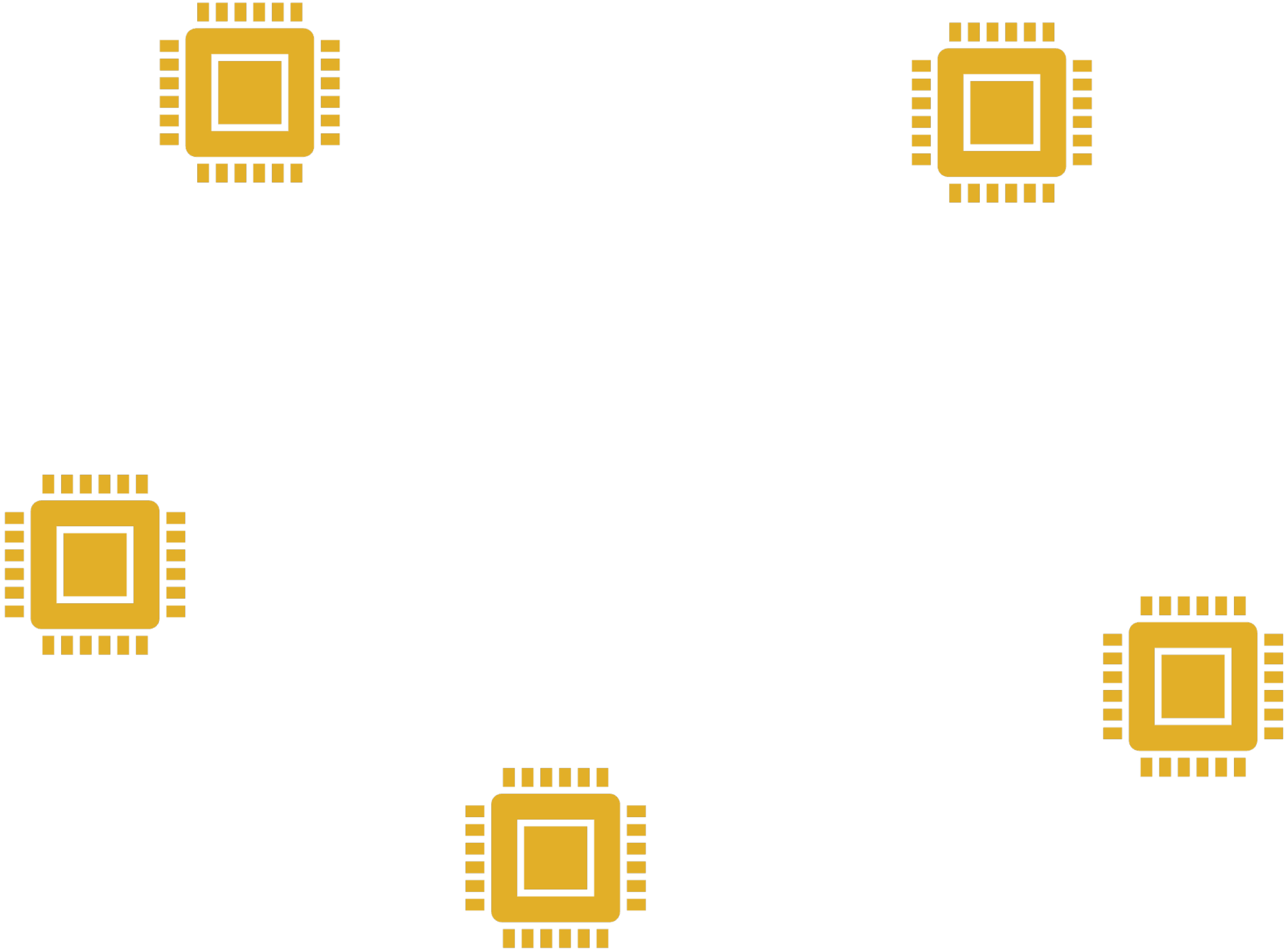
Thinking Machines Corporation

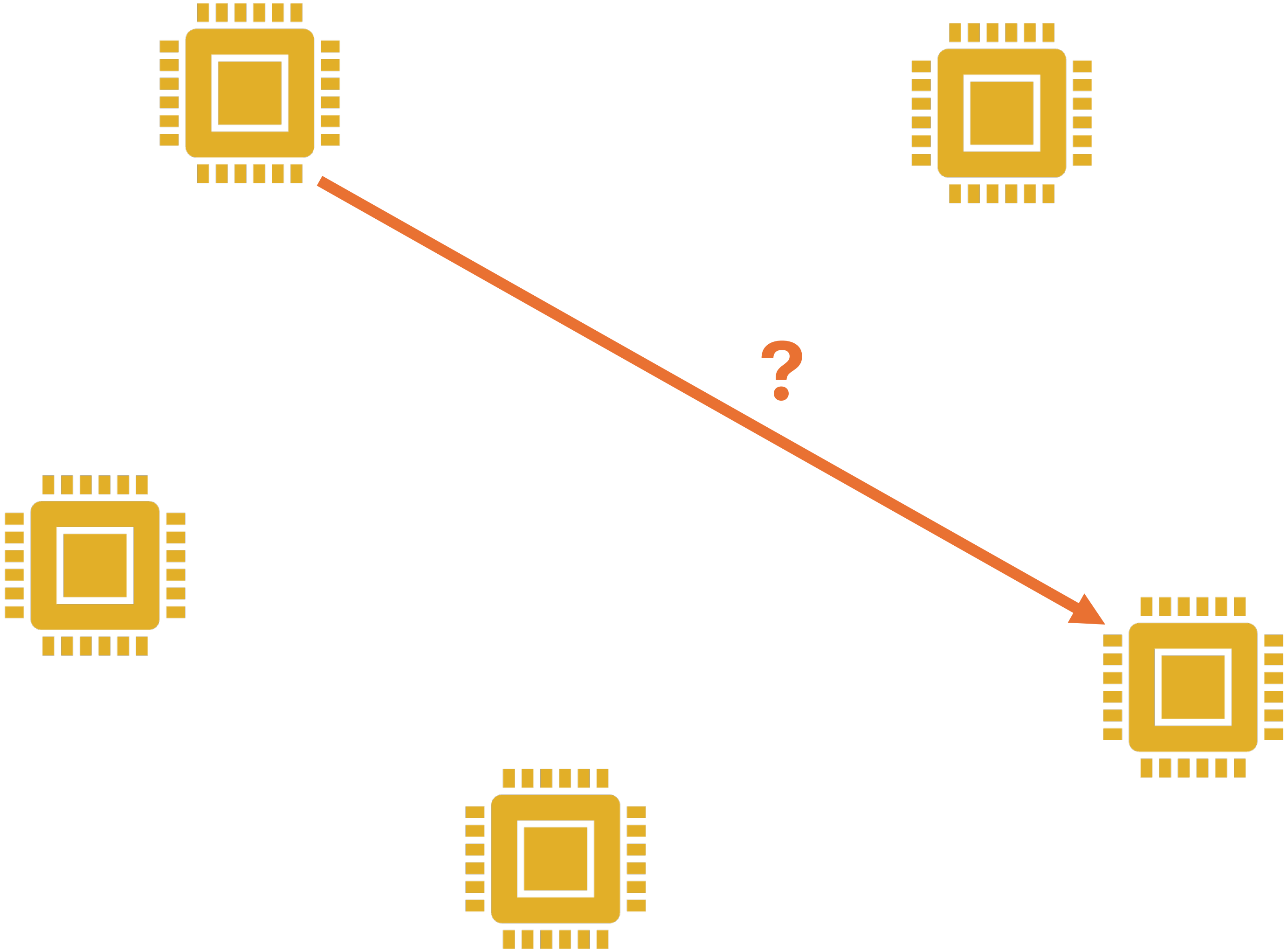
CRAY

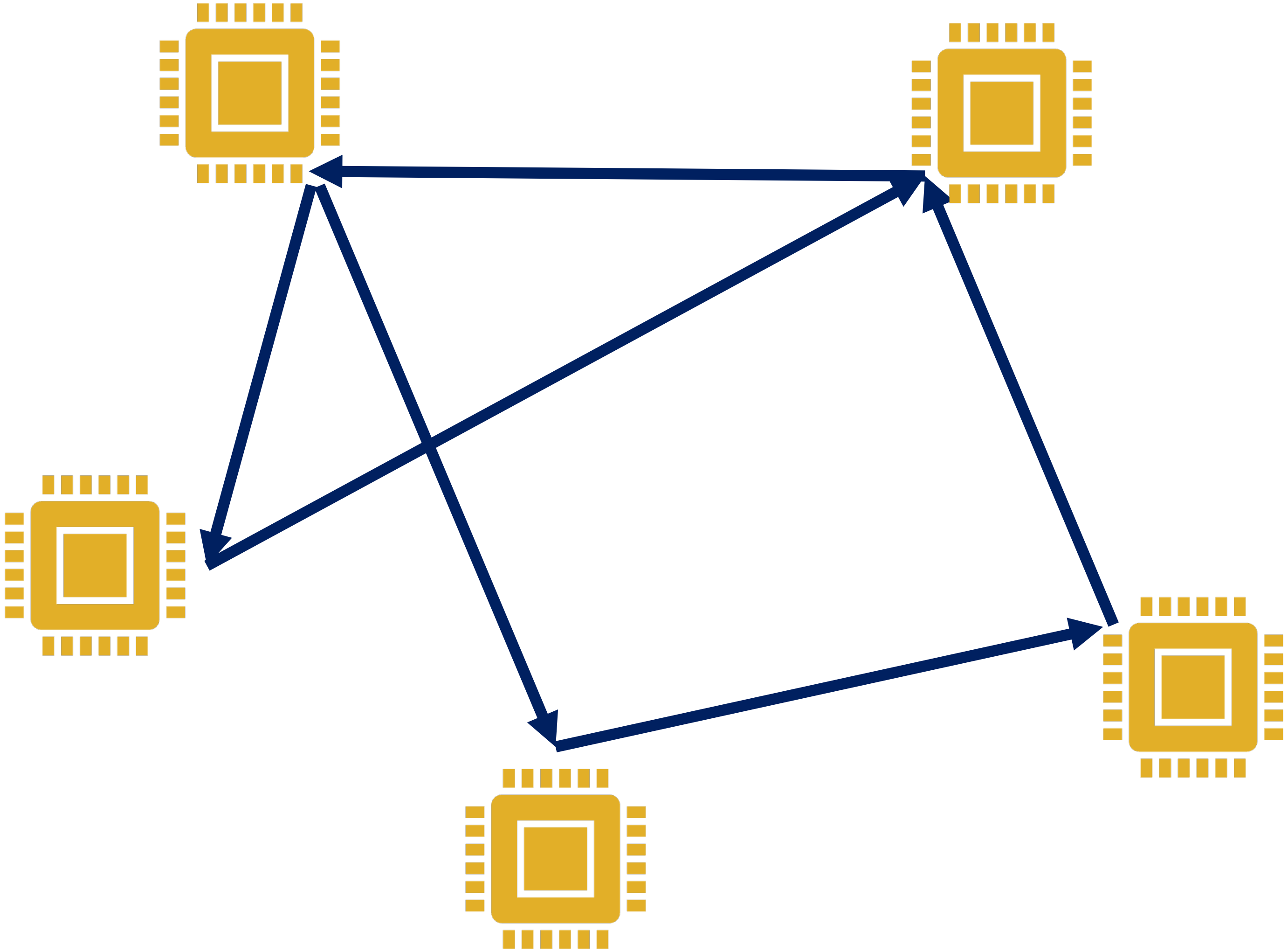
Supercomputer

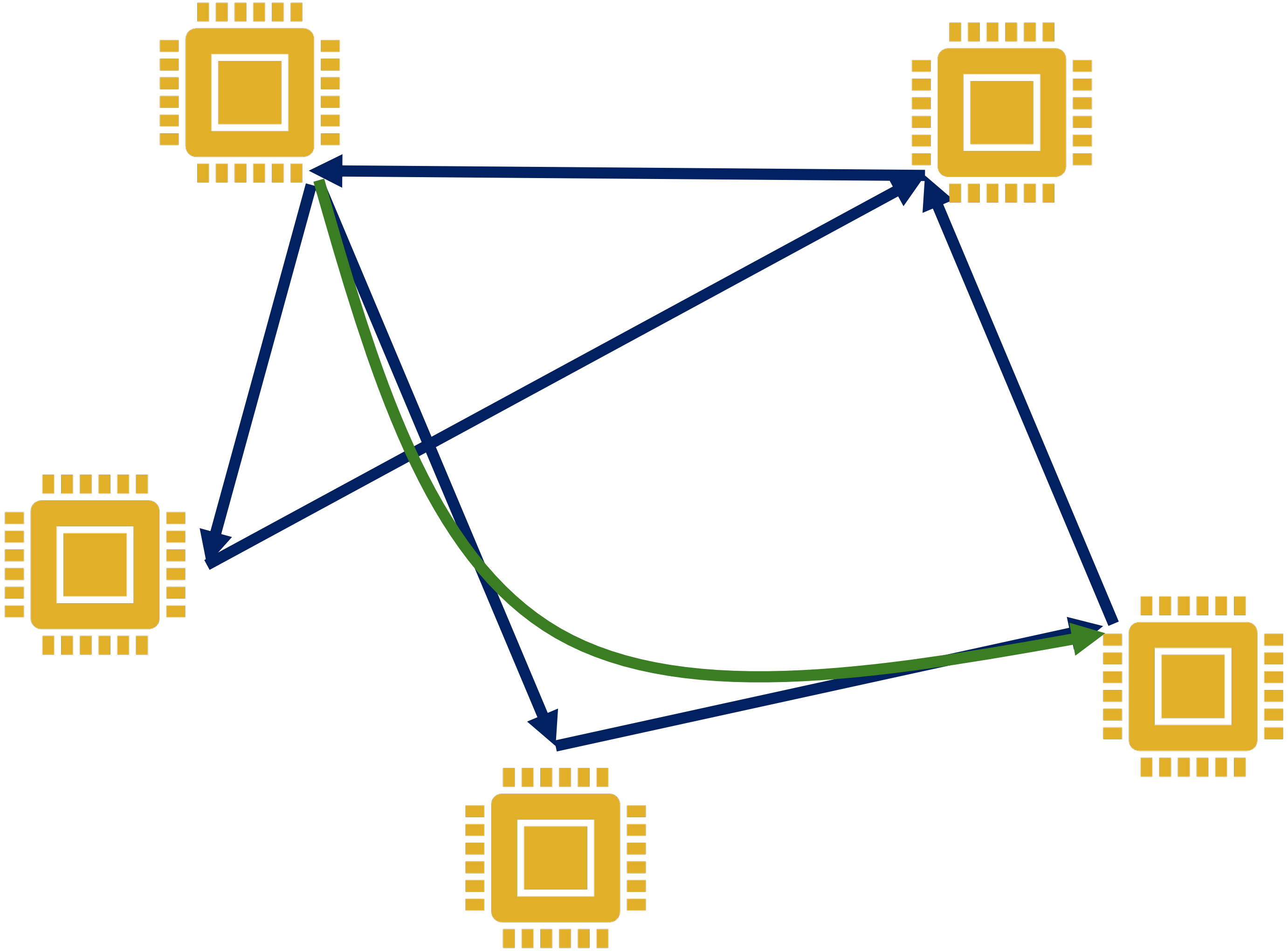


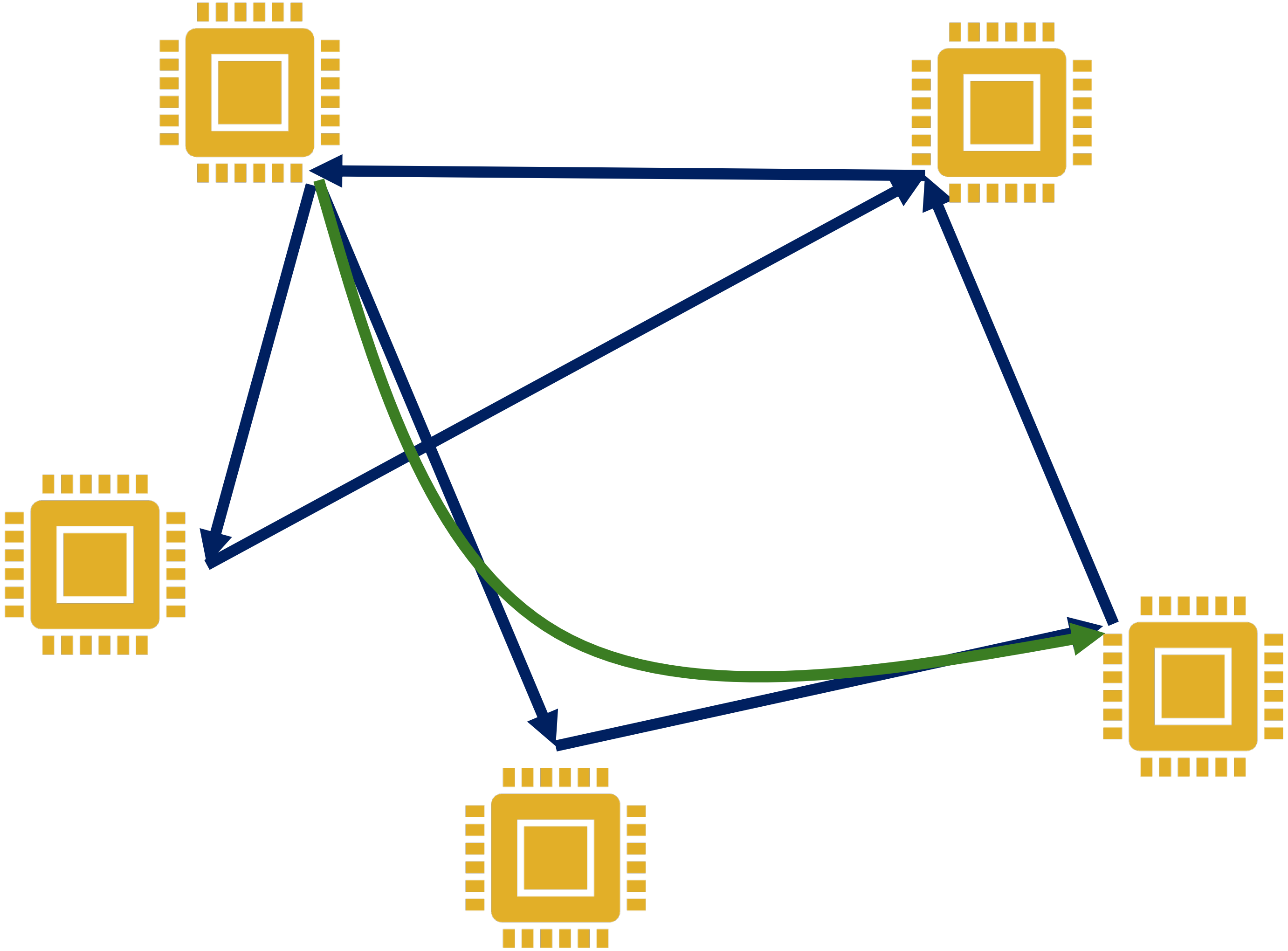
Thinking Machine Corporation CM-1 at the MoMA







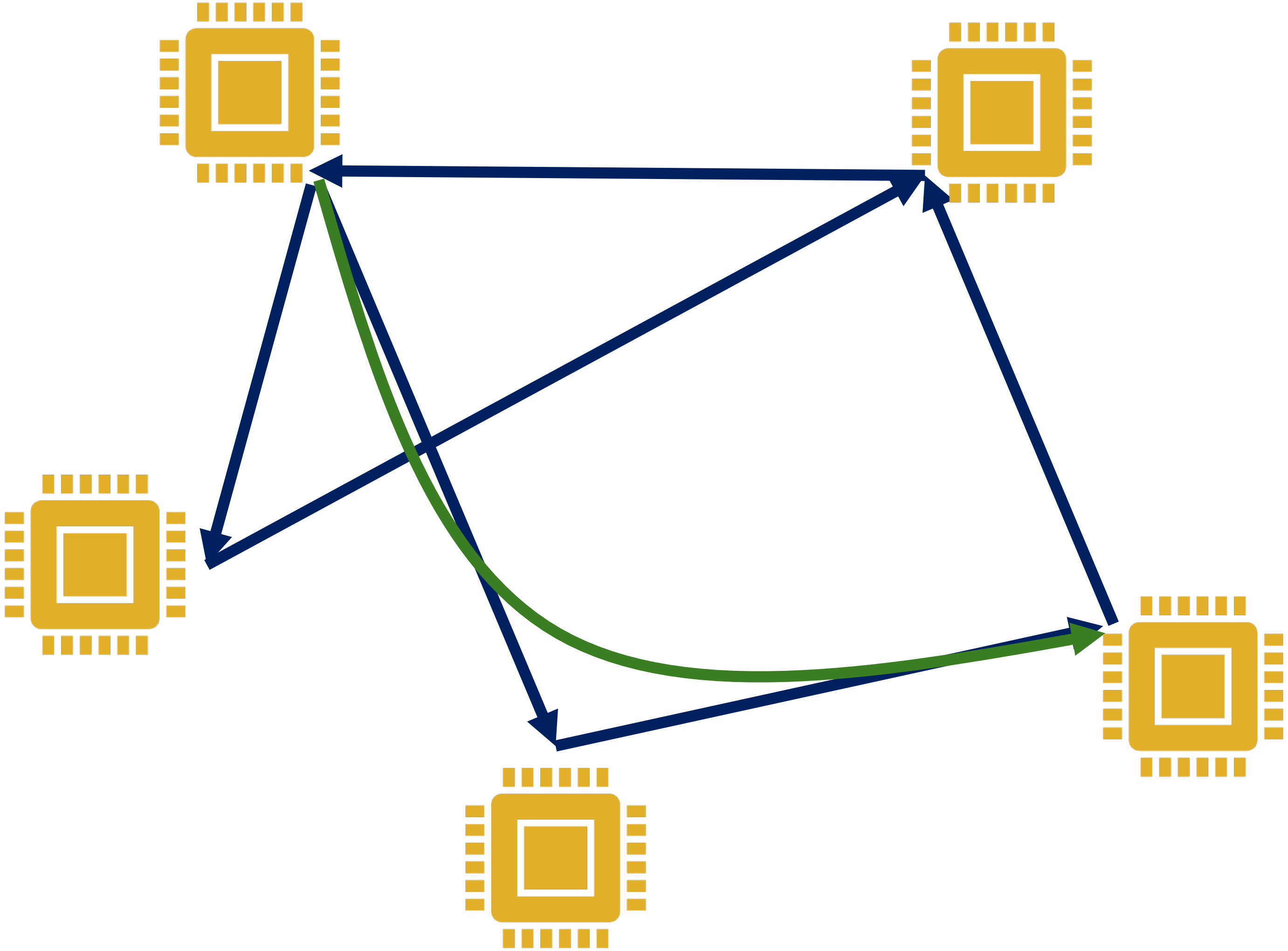




Network Topology

+

Routing Protocol



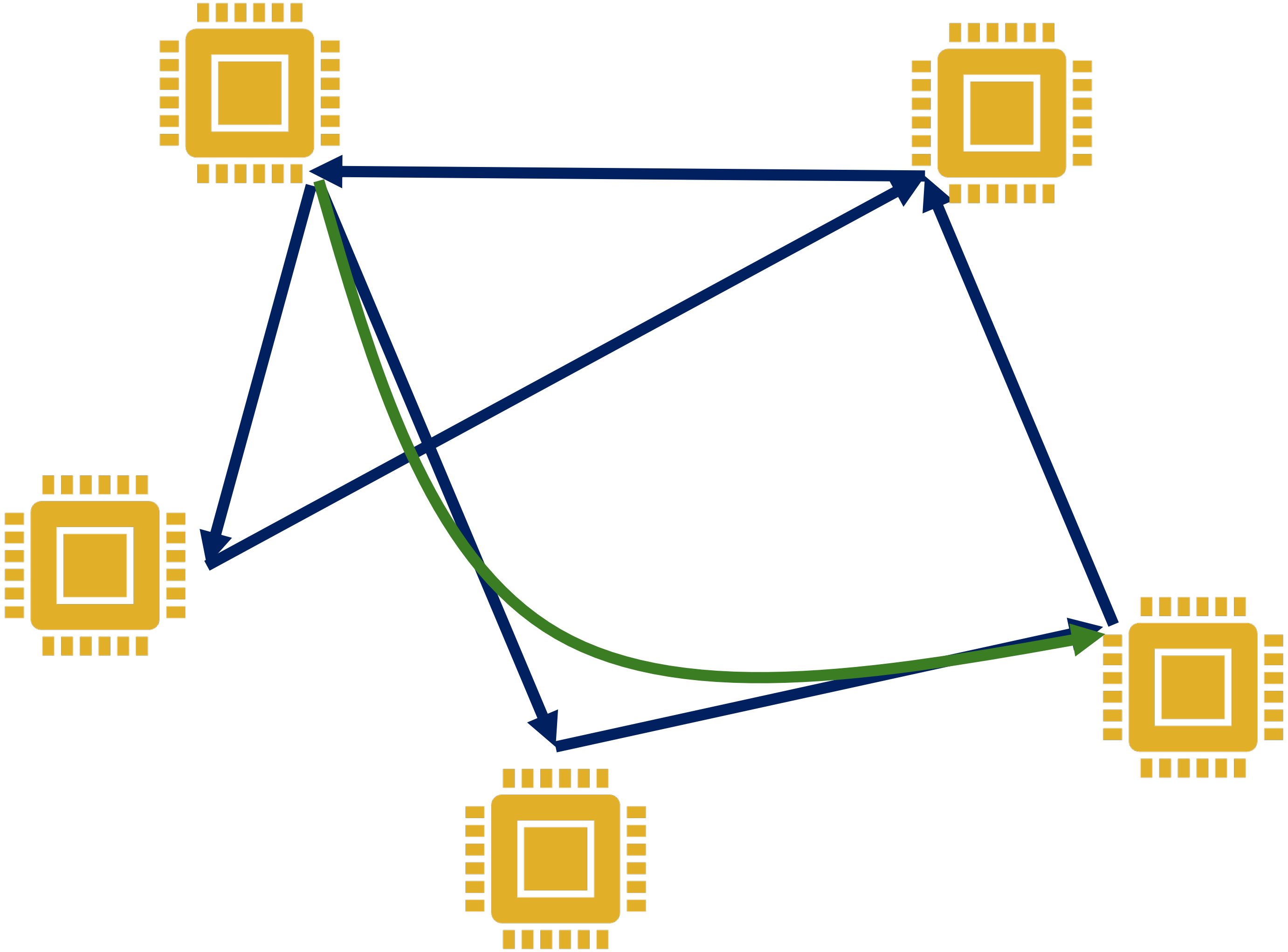
Network Topology

+

Routing Protocol

=

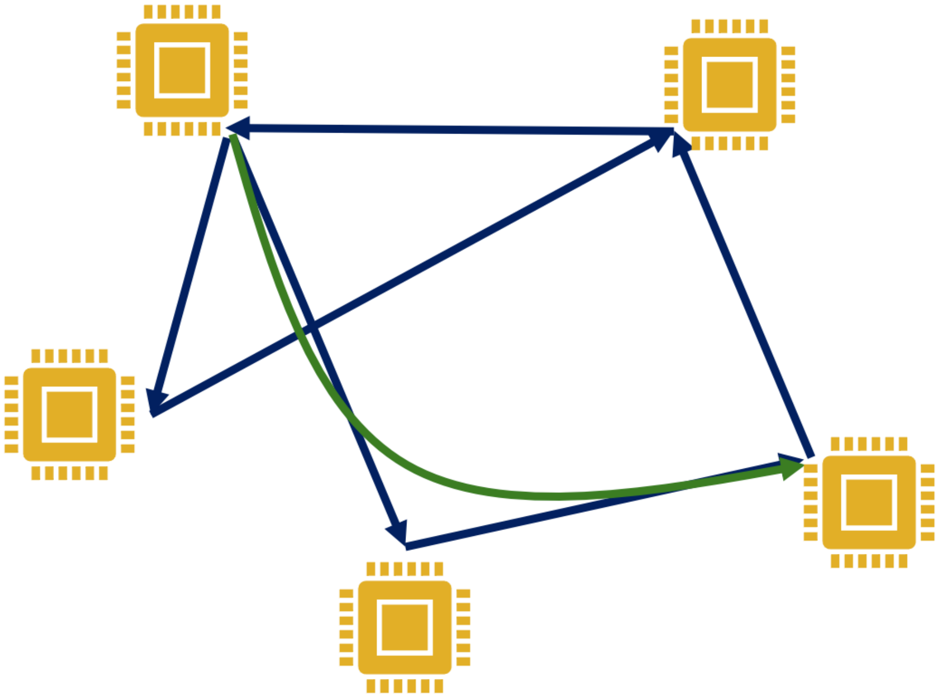
Network Design



Routing on “optimized topologies”

Network Topology

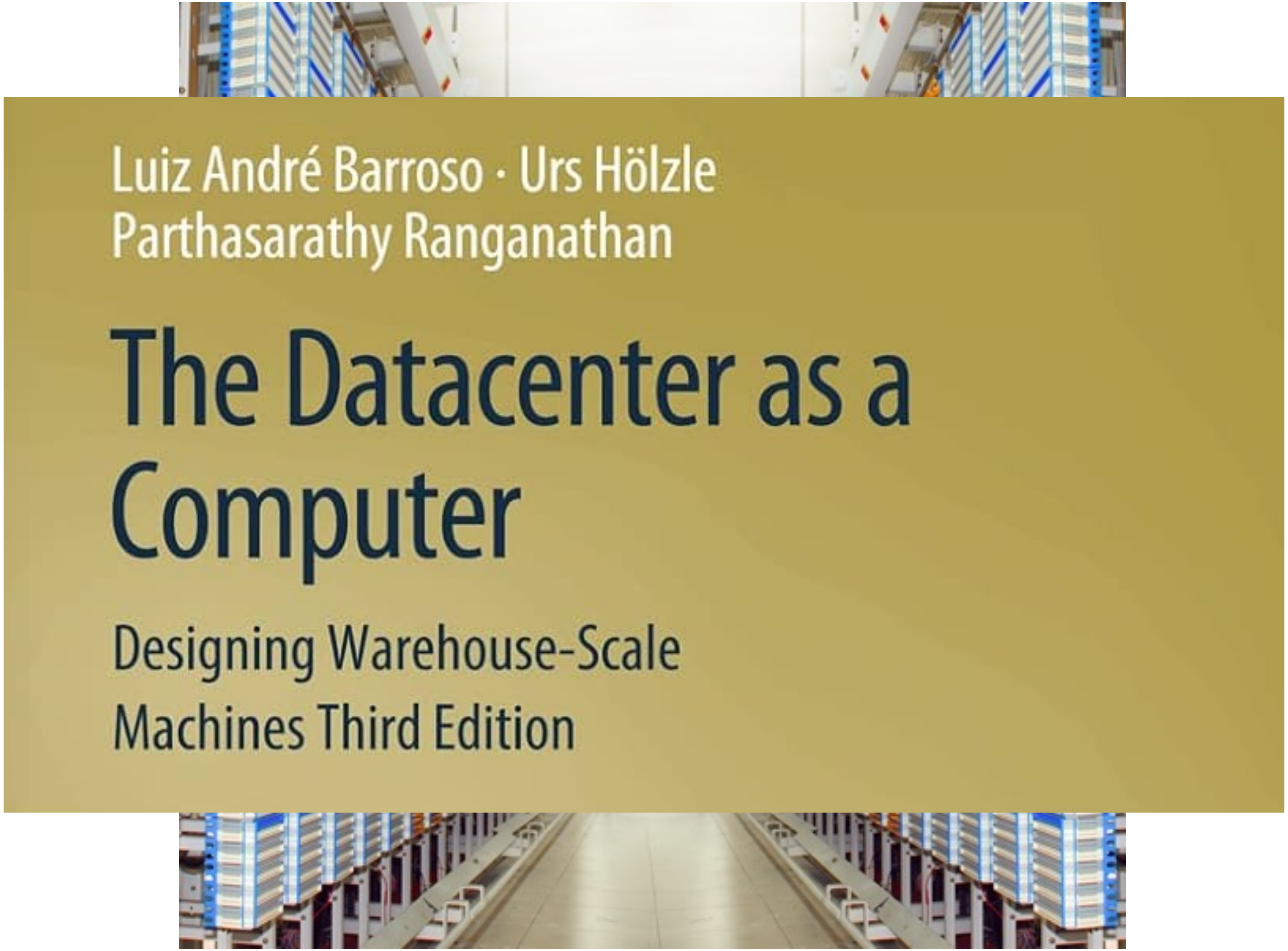
> *“The fundamental problem… is that of simulating arbitrary connection patterns among the processors via a fixed sparse network… For routing packets, the strategy will have to be based on only a minute fraction of the total information necessary to specify the complete communication pattern.”*
>
> *—Leslie Valiant and Gordon Brebner (1981)*



Goal: *Oblivious* Routing

All routing decisions made before traffic is seen

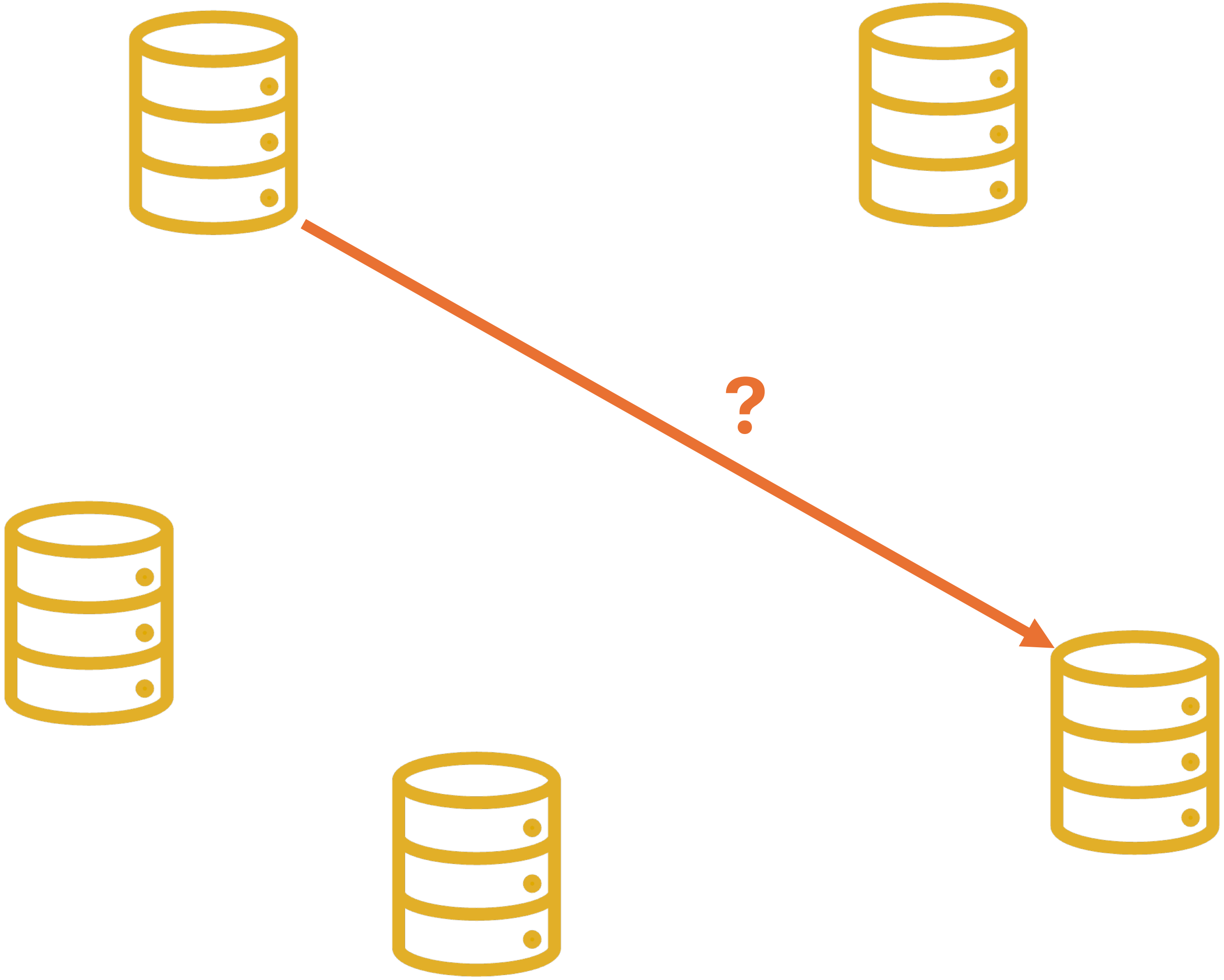
For every source-sink pair, define a distribution over routing paths

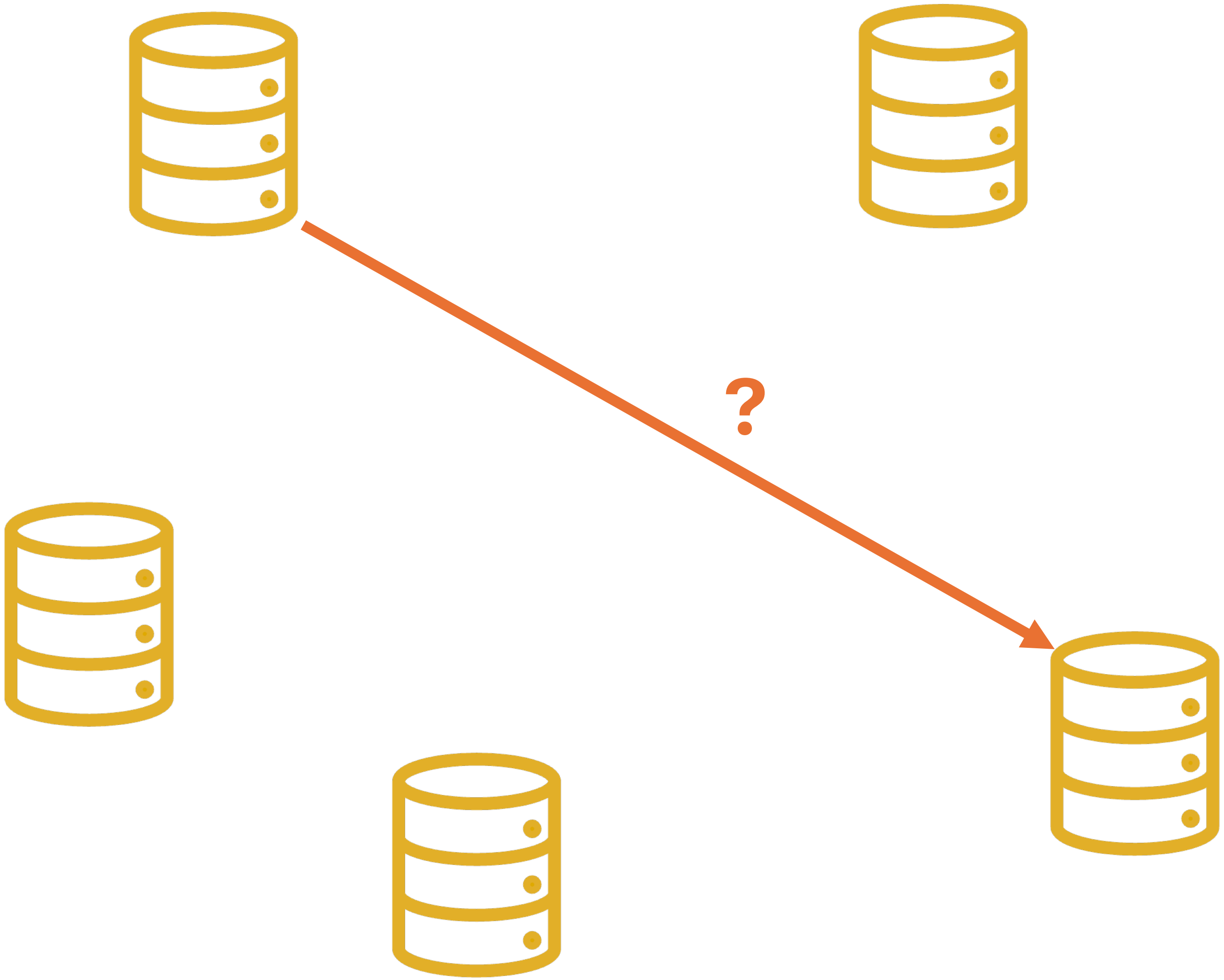


Luiz André Barroso · Urs Hölzle
Parthasarathy Ranganathan

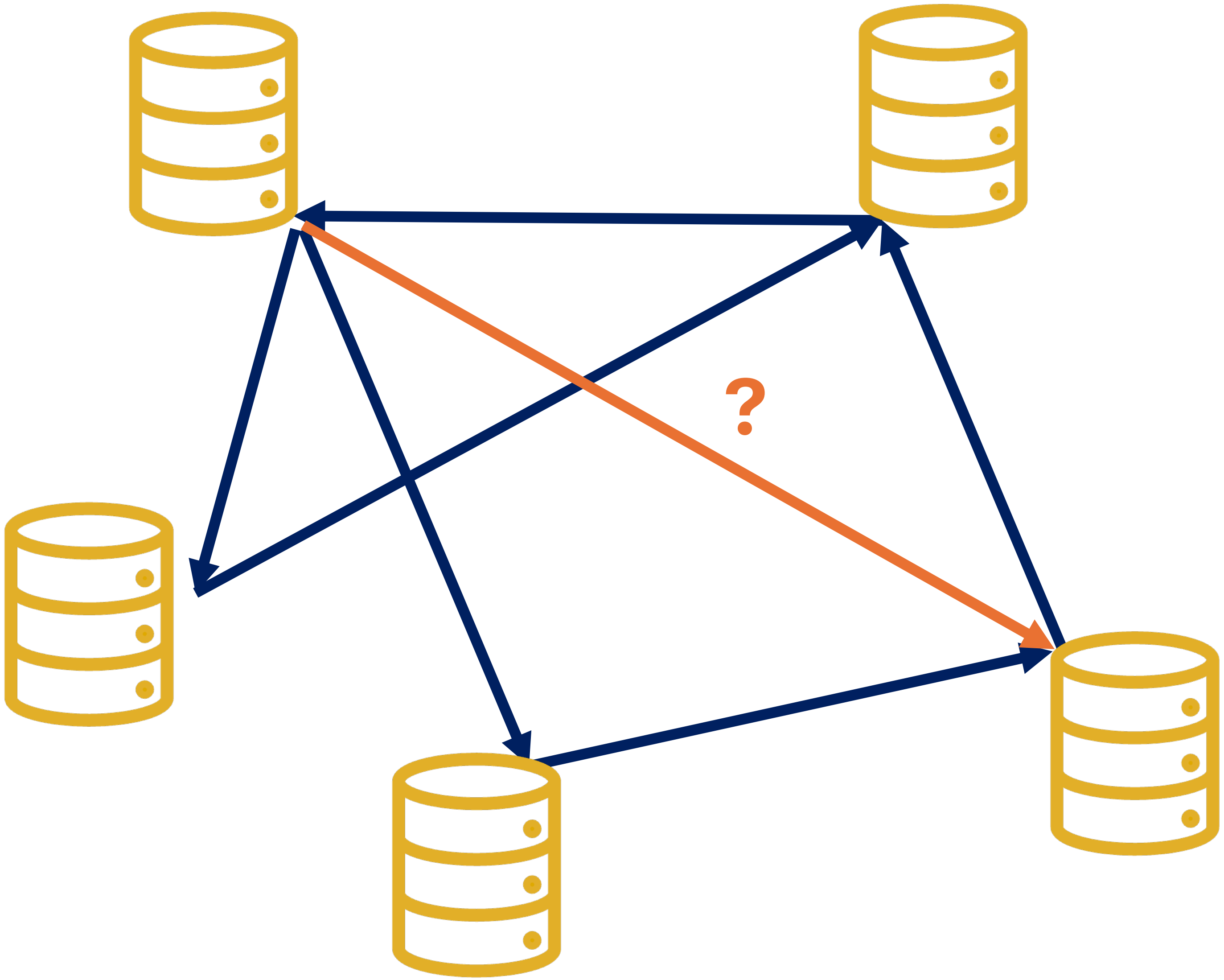
The Datacenter as a Computer

Designing Warehouse-Scale Machines Third Edition



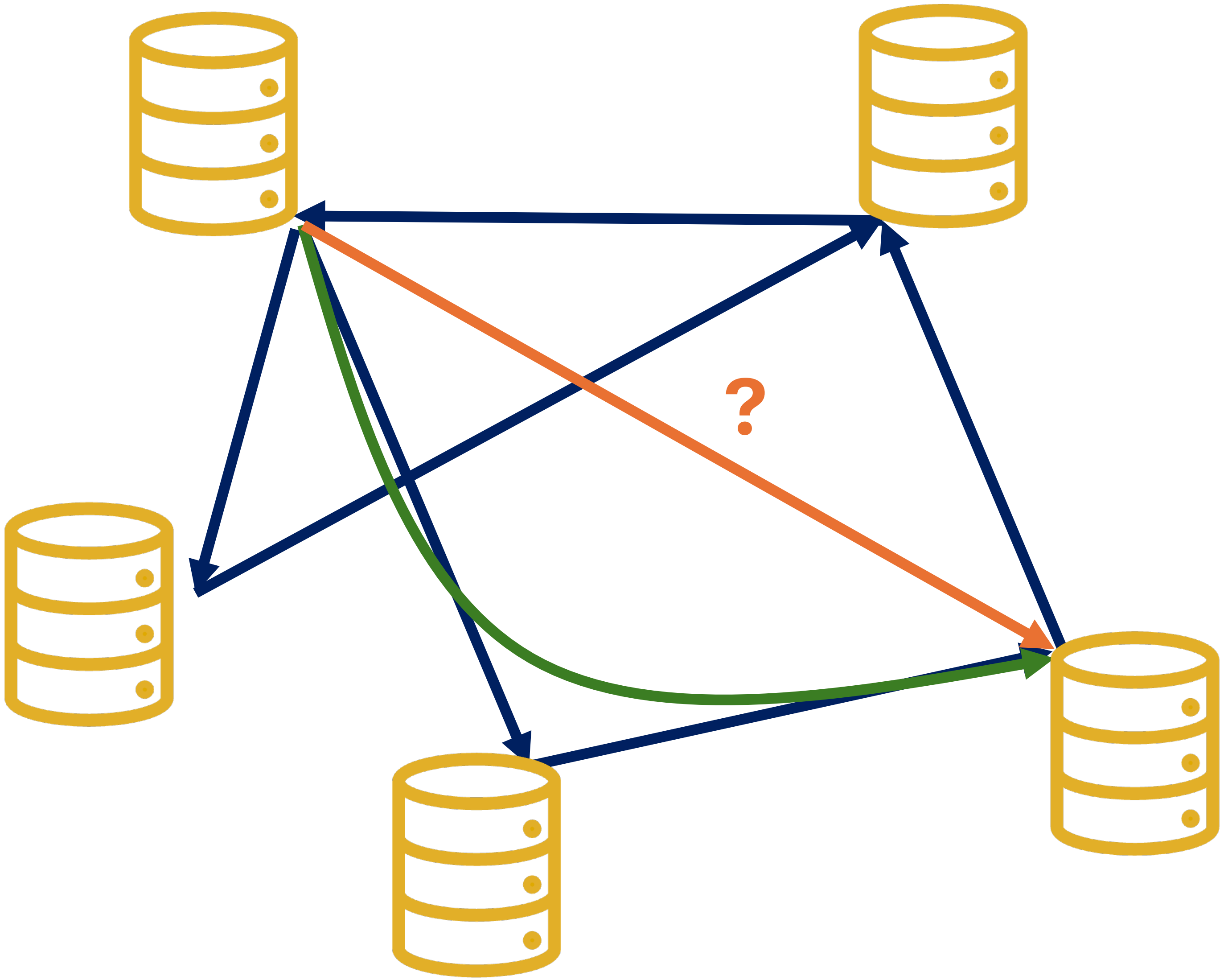


Network Design:



Network Design:

Network Topology



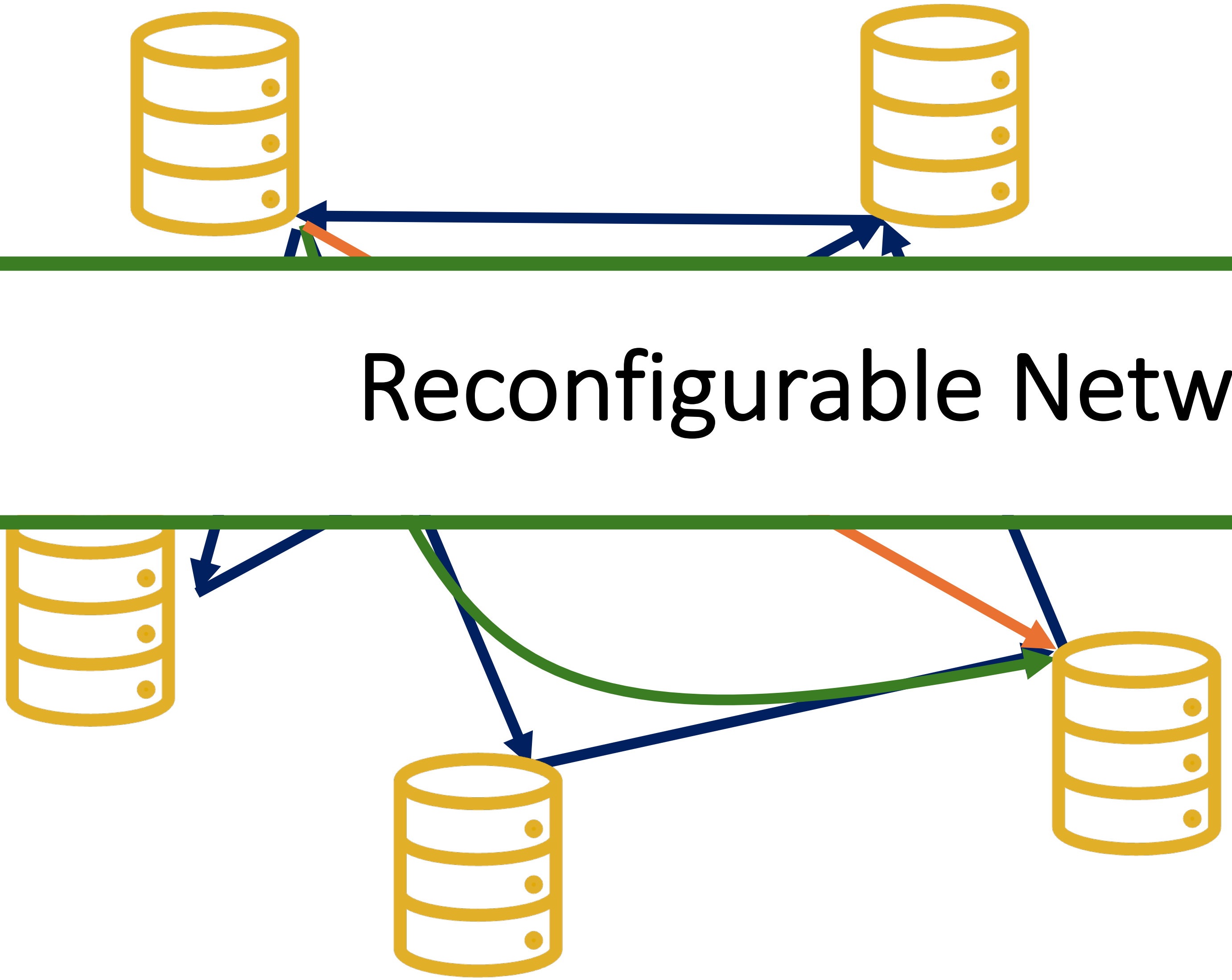
Network Design:

Network Topology

+

Routing Protocol

Reconfigurable Networks



Network Design:

+

Routing Protocol

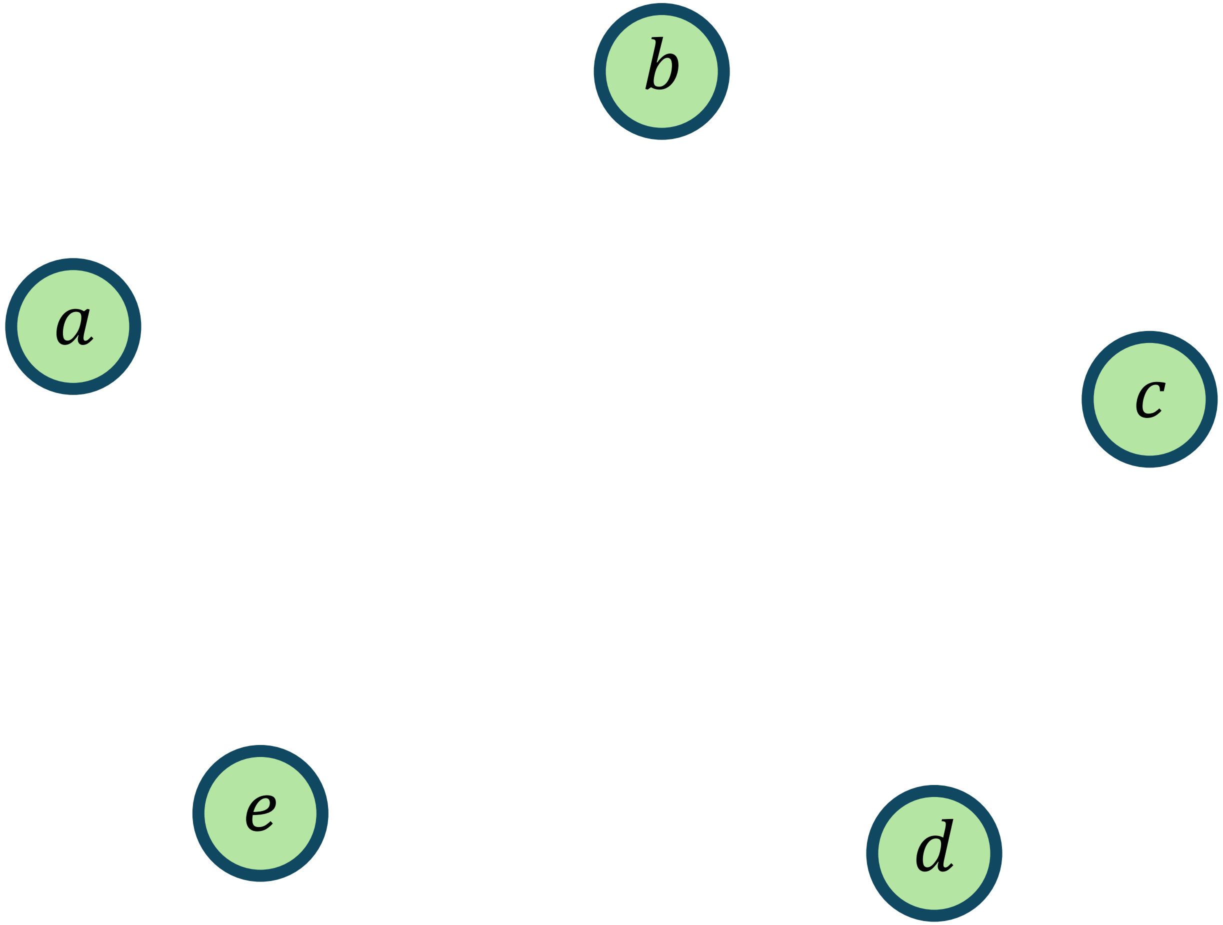
Reconfigurable Networks

- Edges can be reconfigured over time
- Edge set at each timestep may be arbitrary, with a small in/out-degree constraint d

Reconfigurable Networks

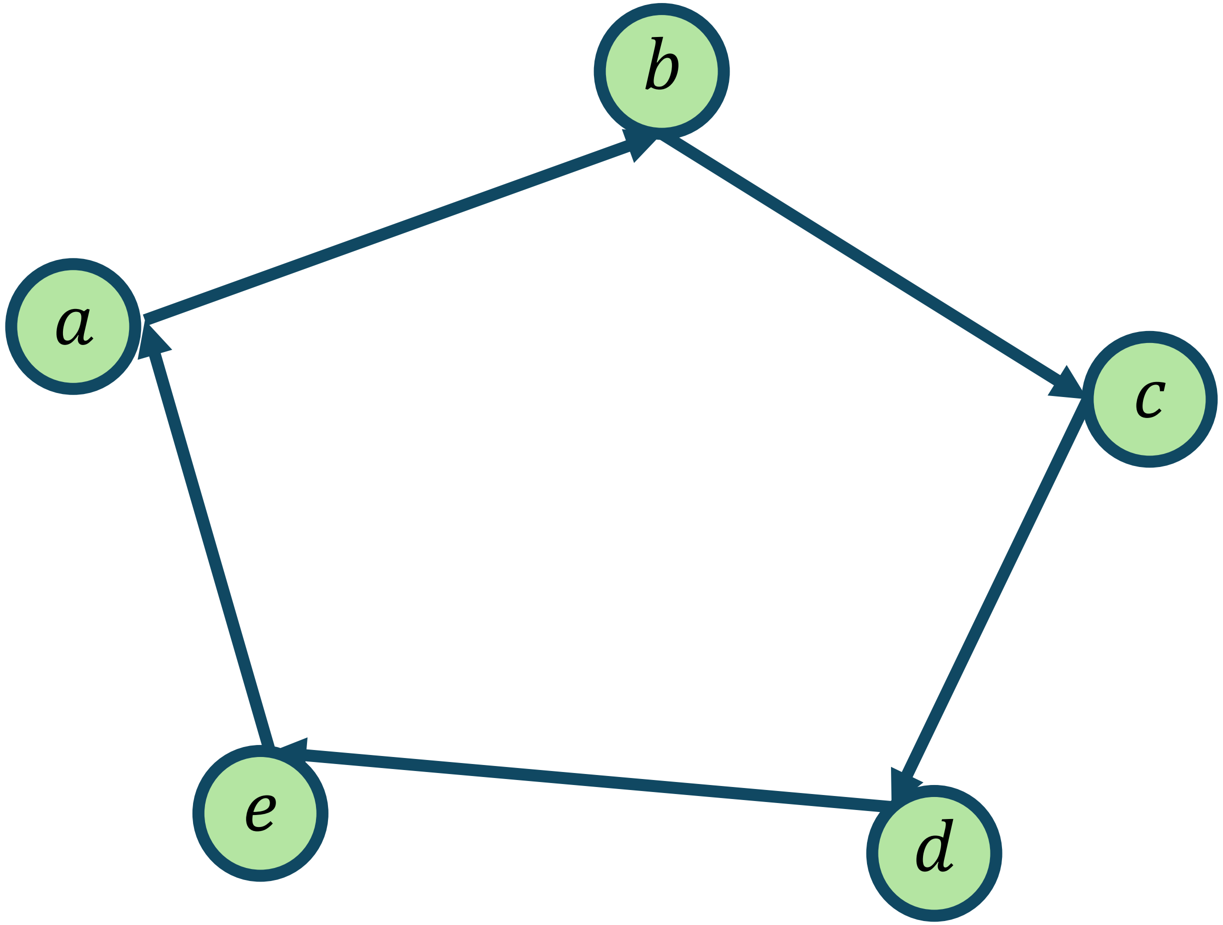
In/out-degree constraint: 1

Reconfigurable Networks



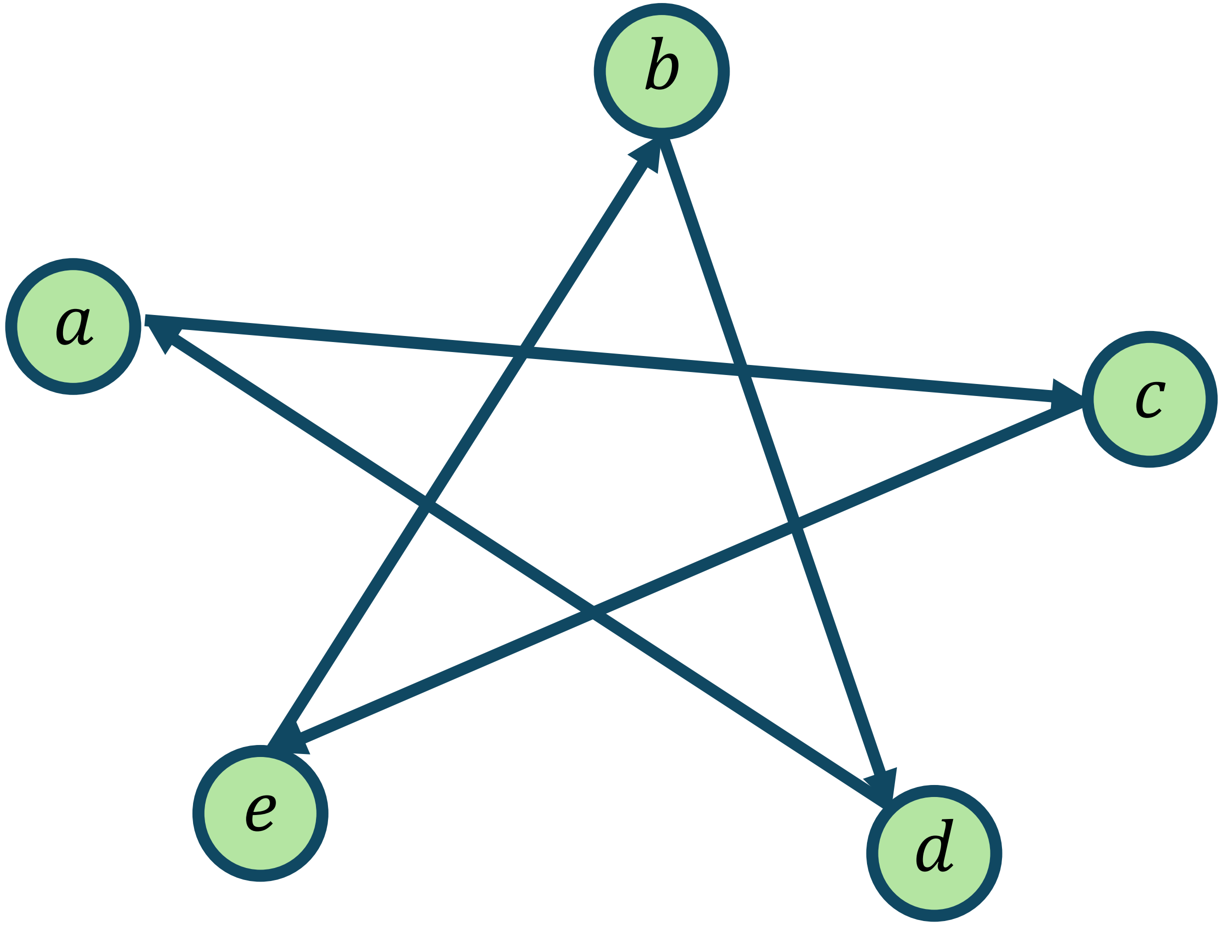
In/out-degree constraint: 1

Reconfigurable Networks

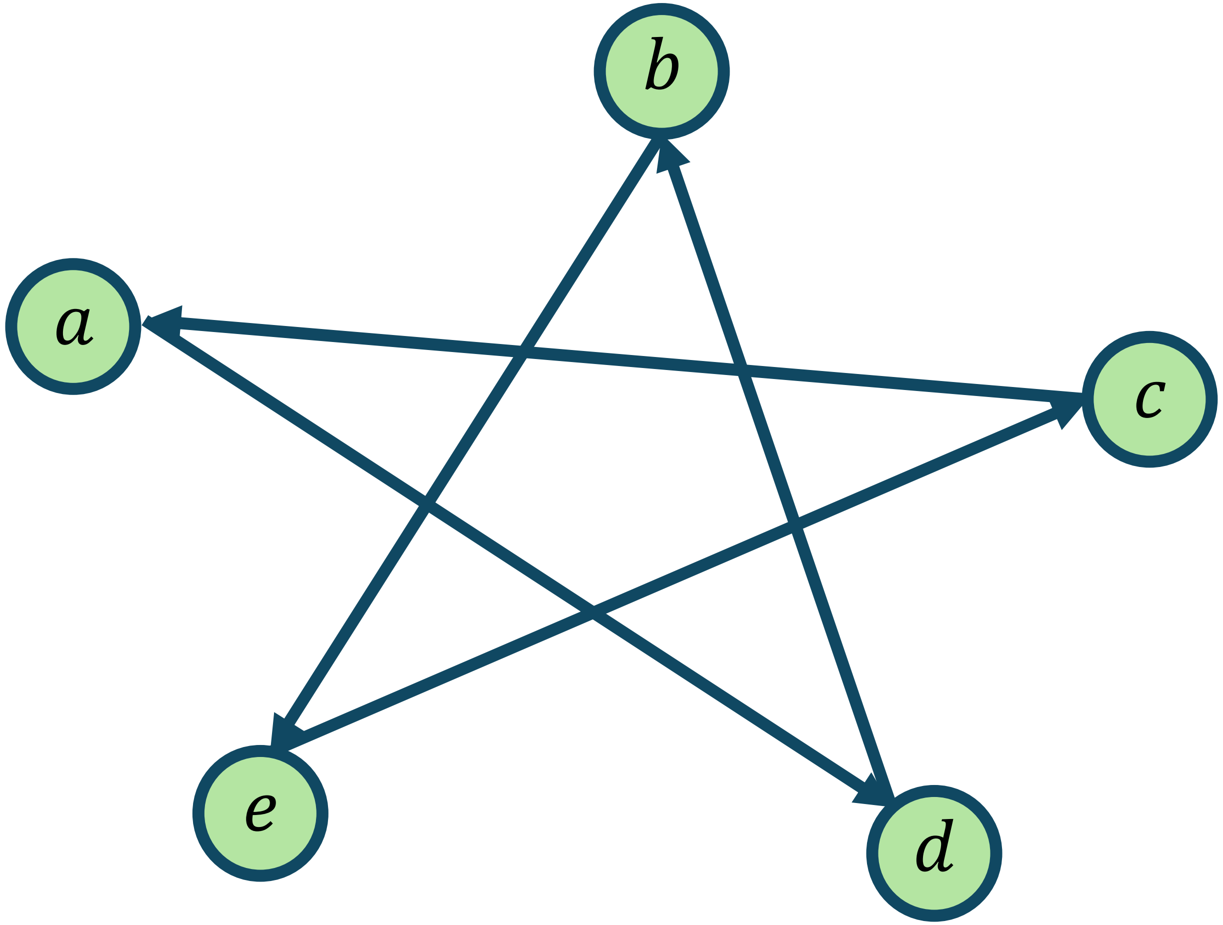


In/out-degree constraint: 1

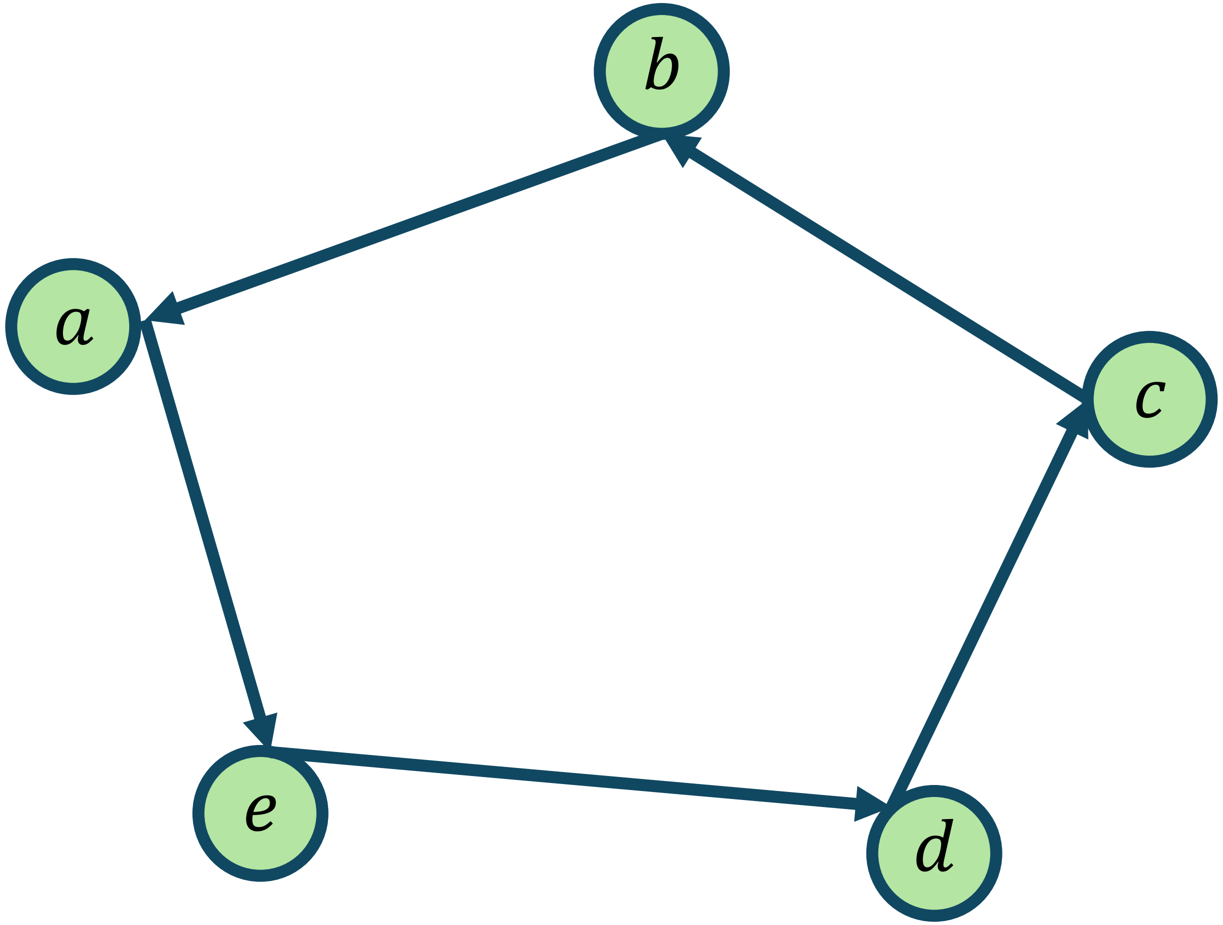
Reconfigurable Networks



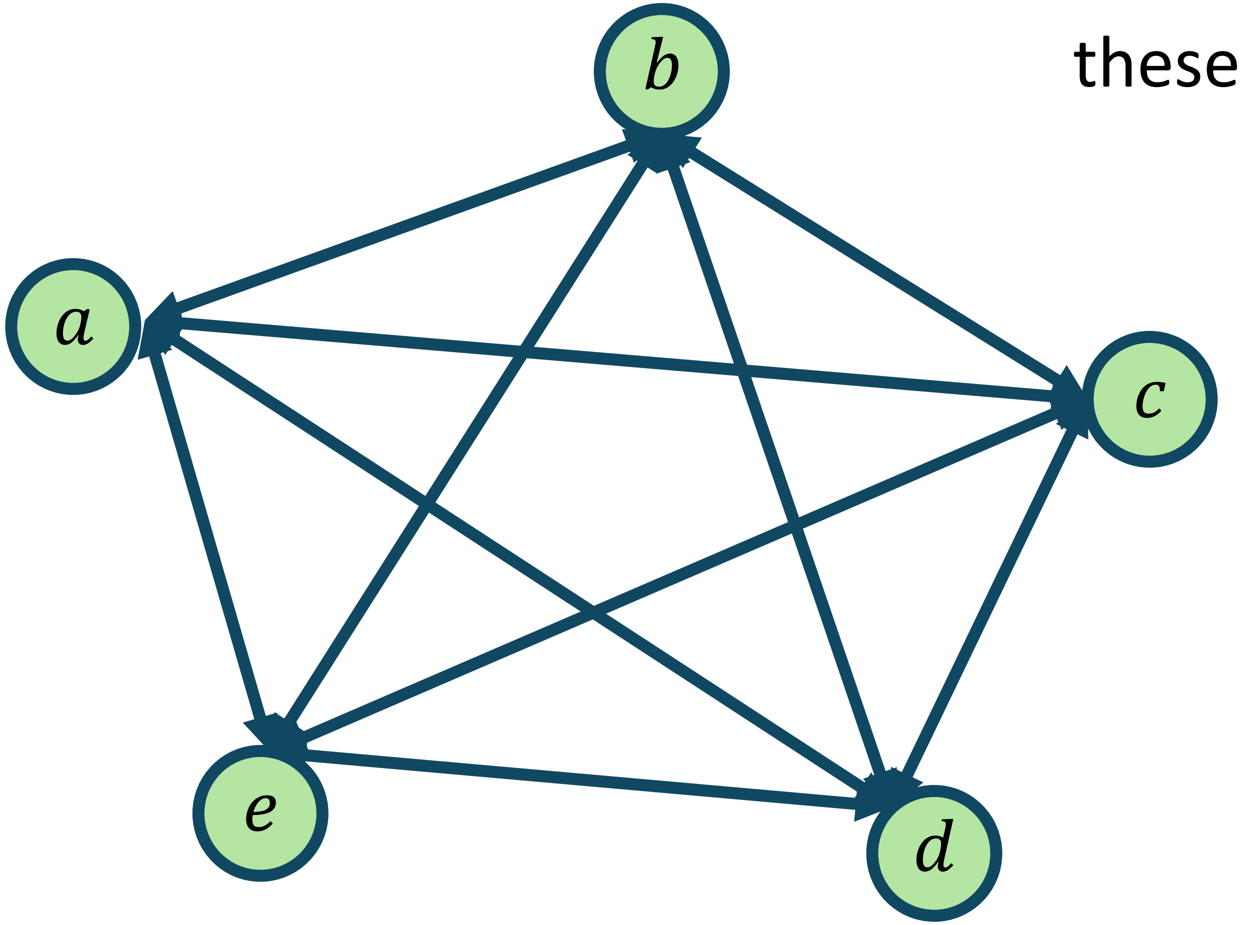
Reconfigurable Networks



Reconfigurable Networks

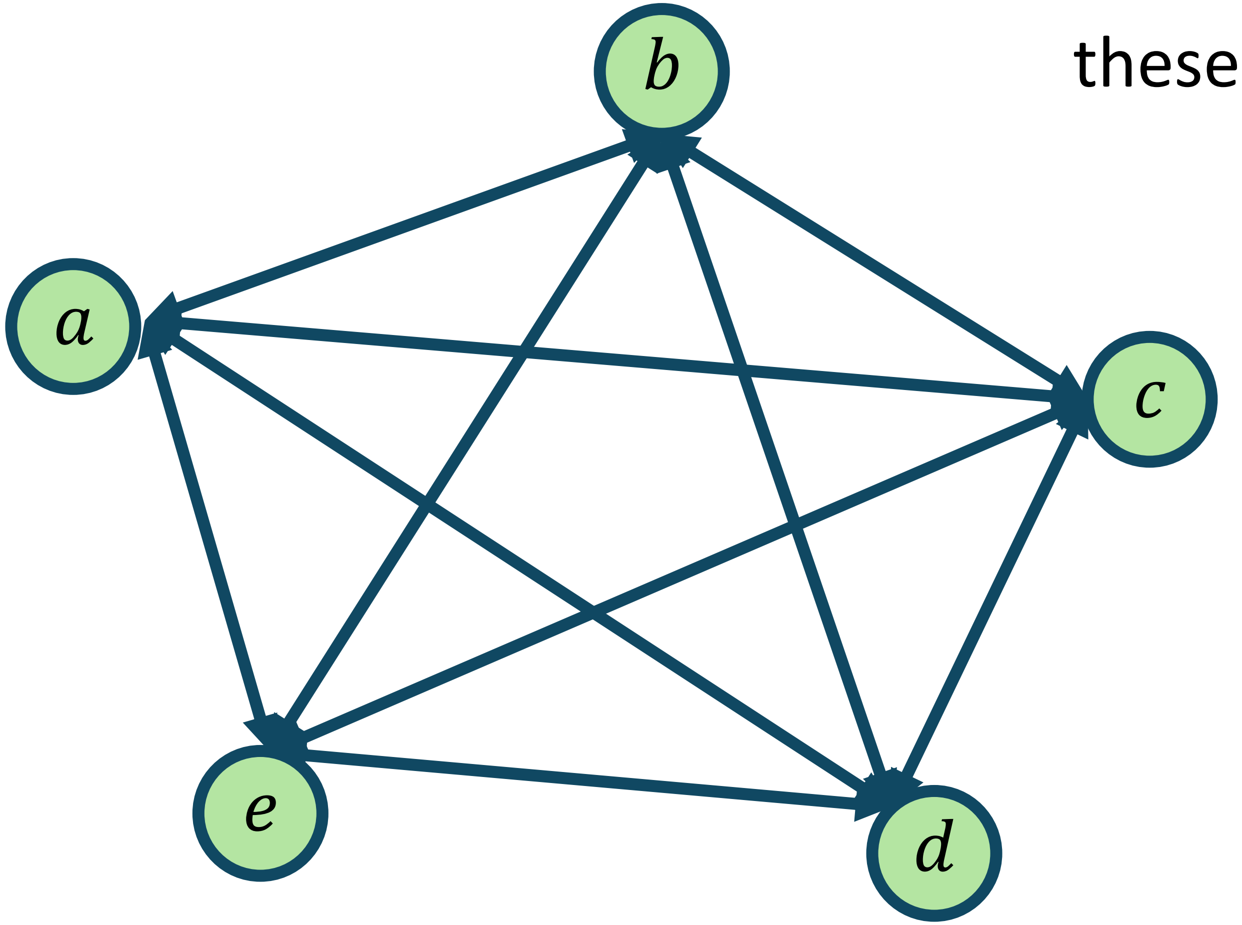


Reconfigurable Networks



Periodically rotate through these connections

Reconfigurable Networks



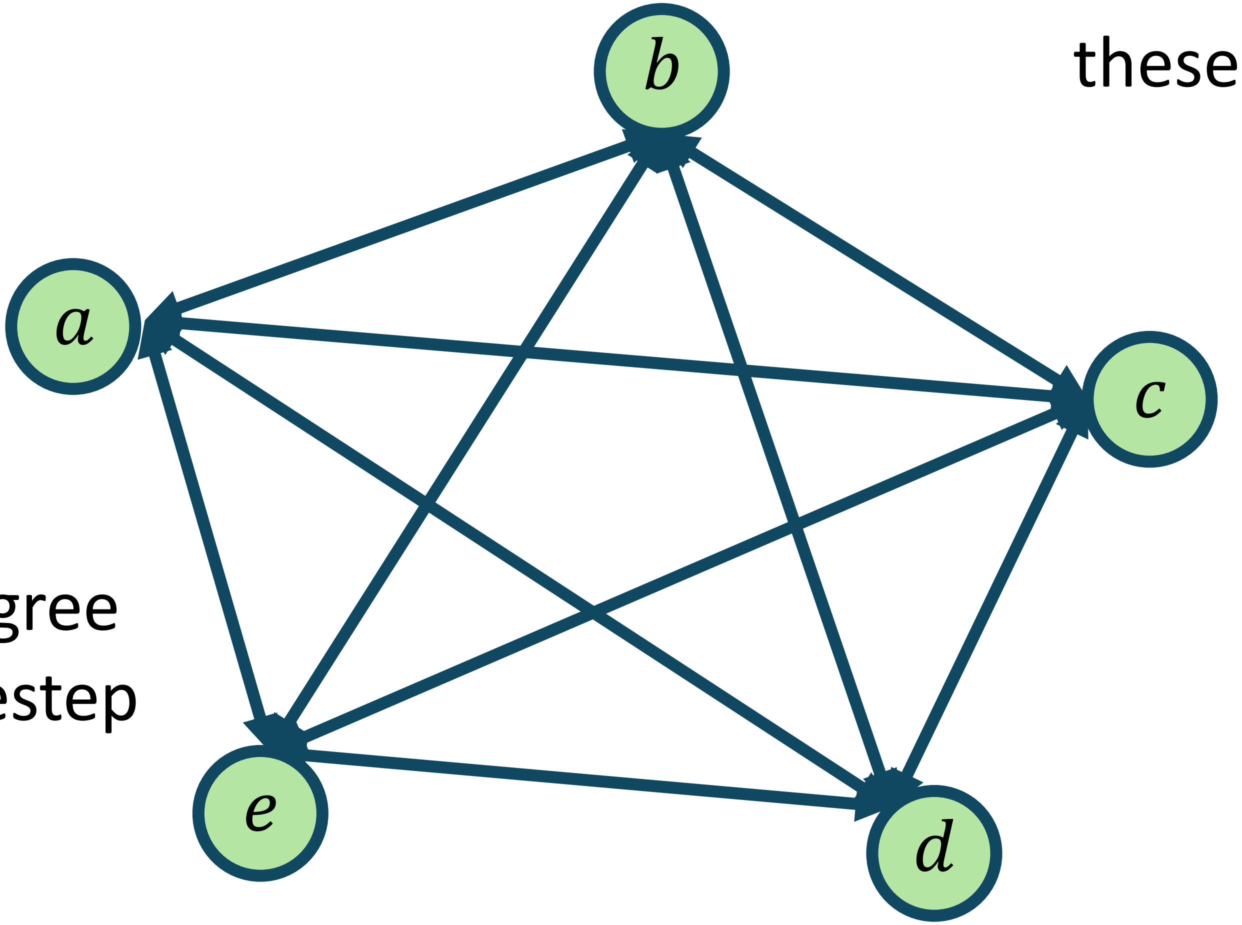
Periodically rotate through these connections

→ a connection schedule (network topology)

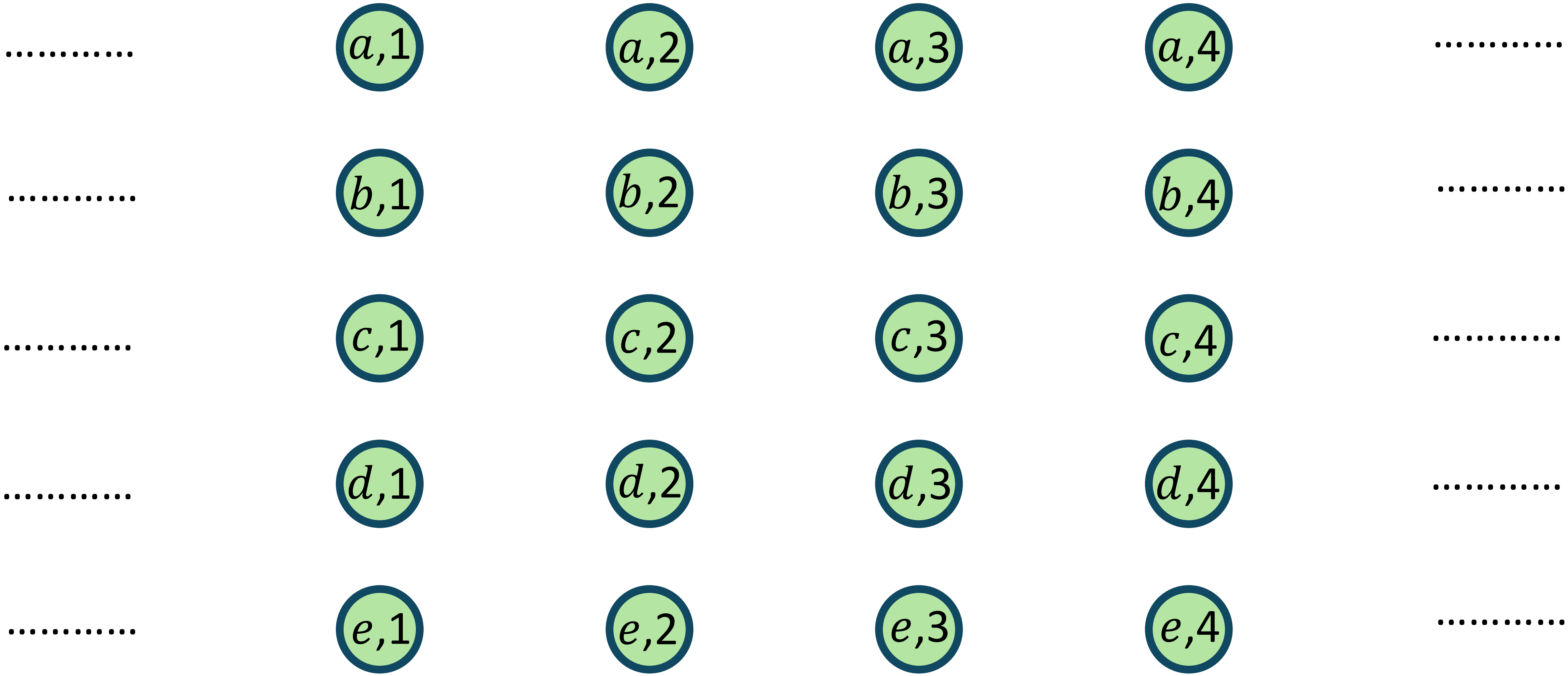
Reconfigurable Networks

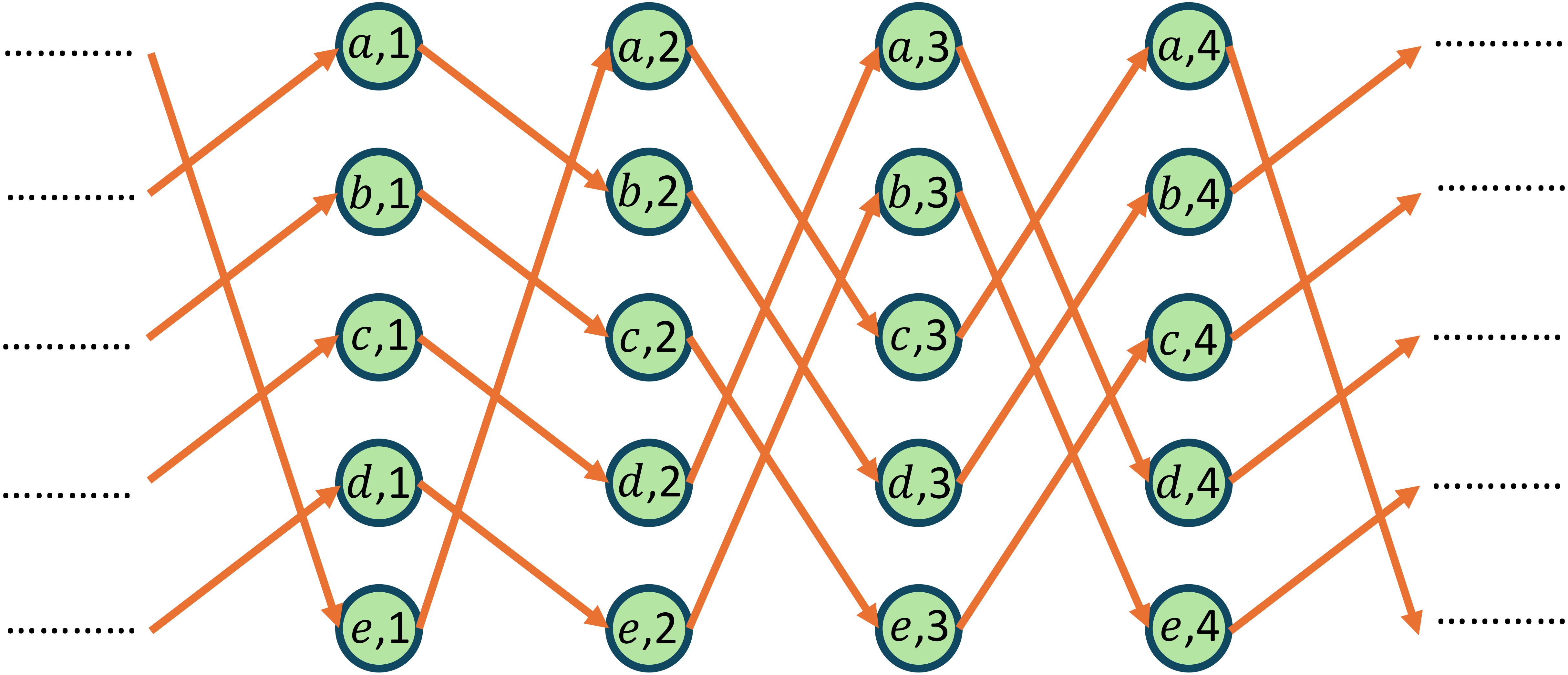
Periodically rotate through these connections

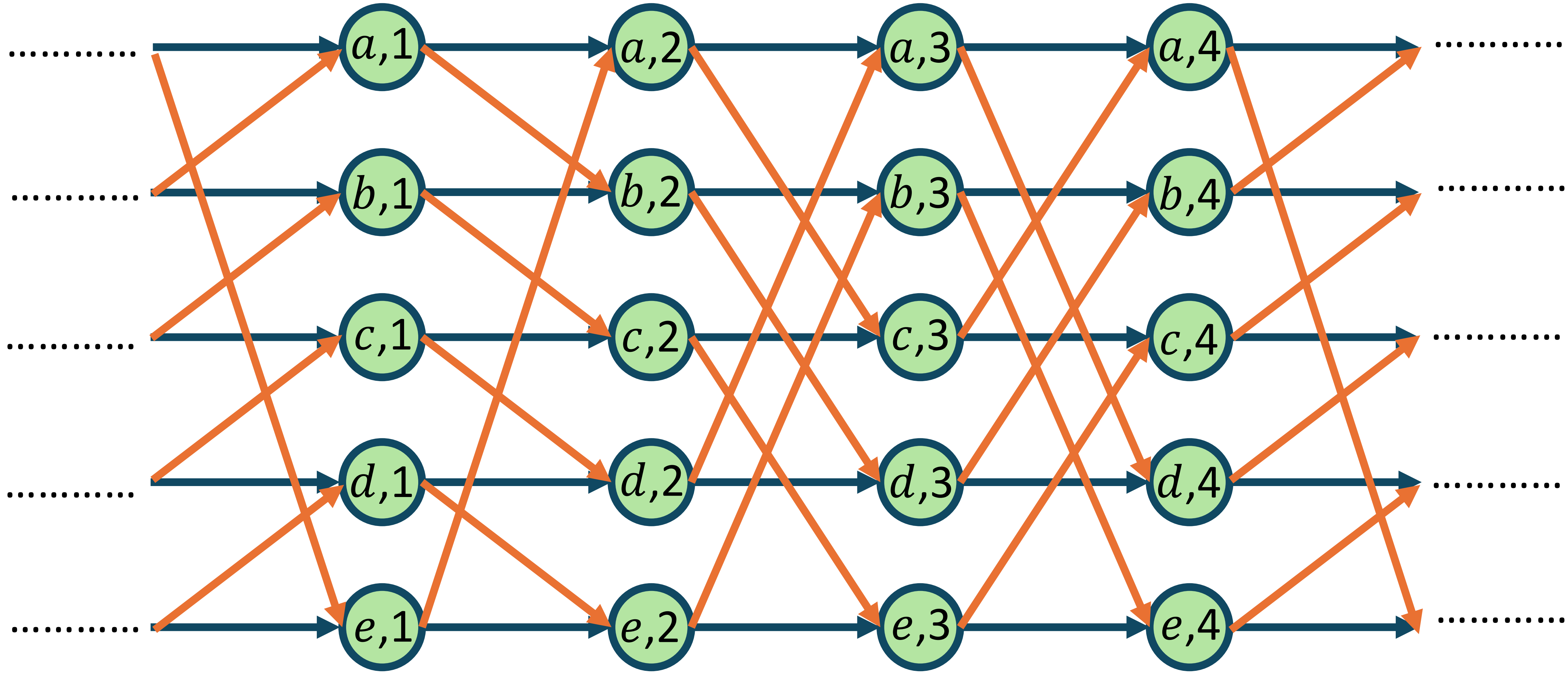
→ a connection schedule (network topology)



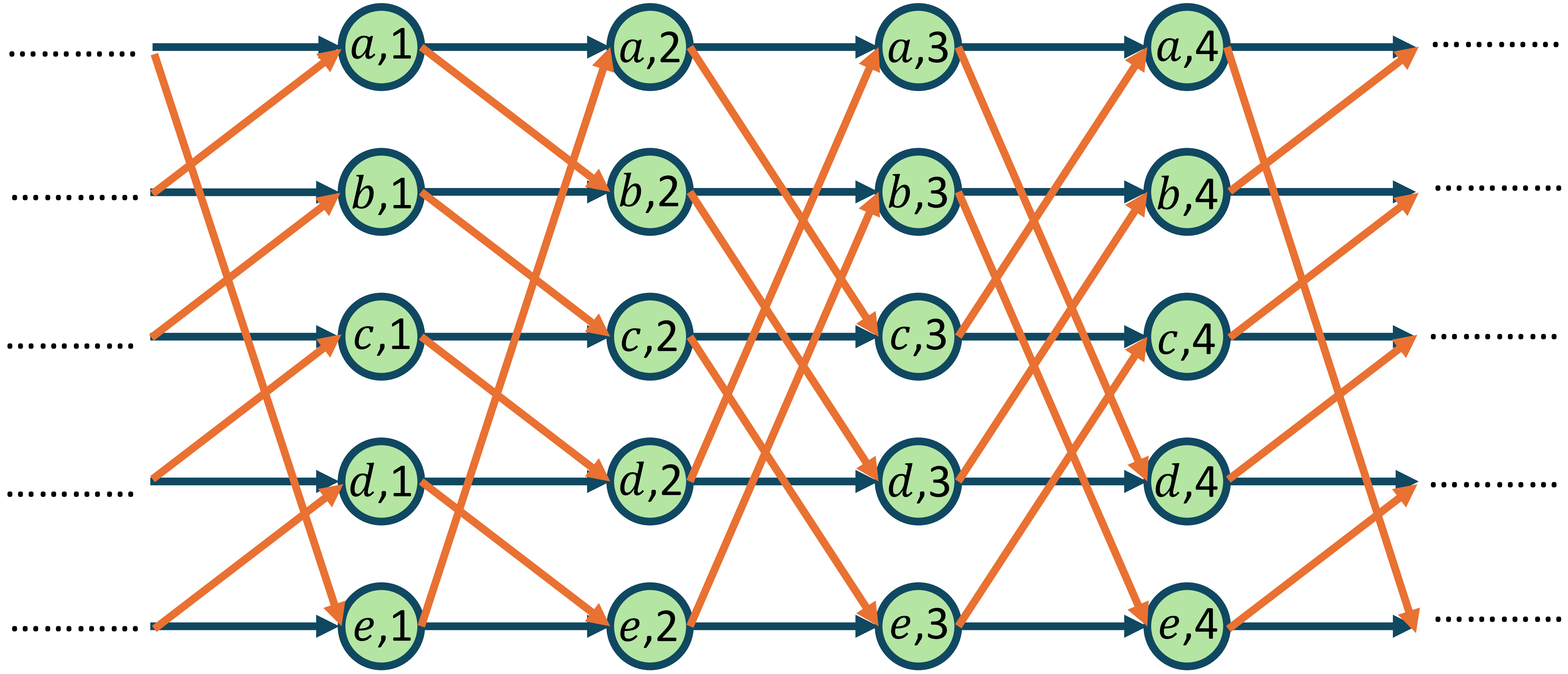
In- and out-degree 1 at every timestep


$a,1$
$a,2$
$a,3$
$a,4$
$b,1$
$b,2$
$b,3$
$b,4$
$c,1$
$c,2$
$c,3$
$c,4$
$d,1$
$d,2$
$d,3$
$d,4$
$e,1$
$e,2$
$e,3$
$e,4$

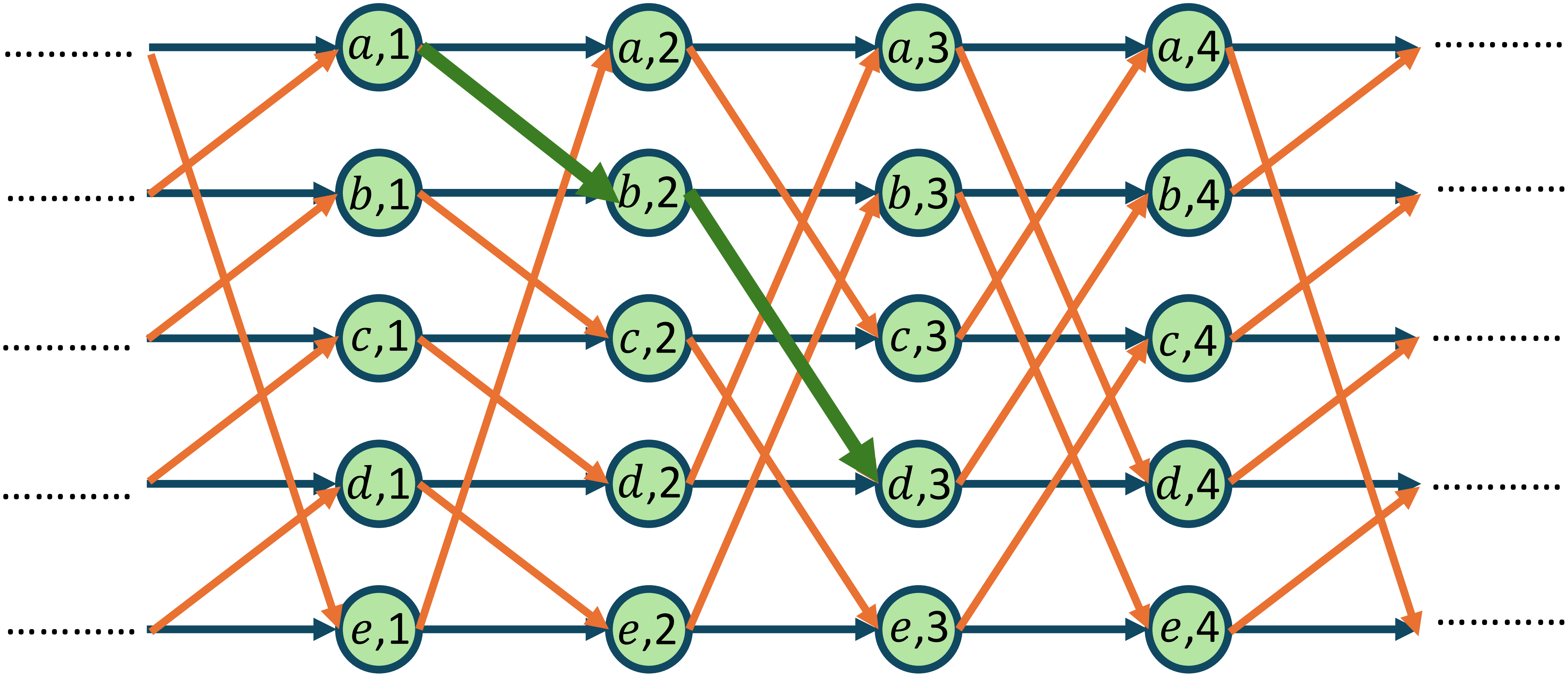

....
$a,1$
$a,2$
$a,3$
$a,4$
....
....
$b,1$
$b,2$
$b,3$
$b,4$
....
....
$c,1$
$c,2$
$c,3$
$c,4$
....
....
$d,1$
$d,2$
$d,3$
$d,4$
....
....
$e,1$
$e,2$
$e,3$
$e,4$
....


$a,1$
$a,2$
$a,3$
$a,4$
$b,1$
$b,2$
$b,3$
$b,4$
$c,1$
$c,2$
$c,3$
$c,4$
$d,1$
$d,2$
$d,3$
$d,4$
$e,1$
$e,2$
$e,3$
$e,4$

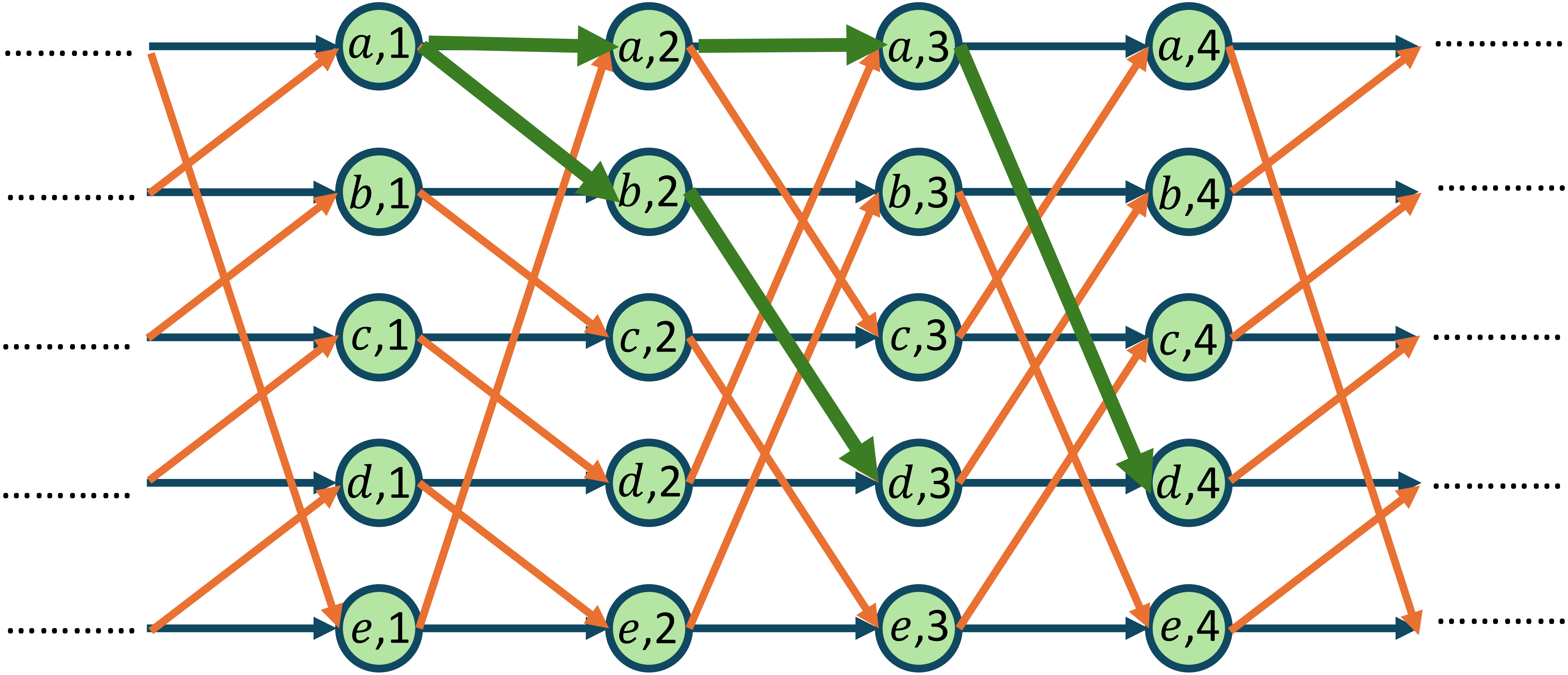
To route $a \rightarrow d$ starting at $t = 1$,



To route $a \rightarrow d$ starting at $t = 1$,

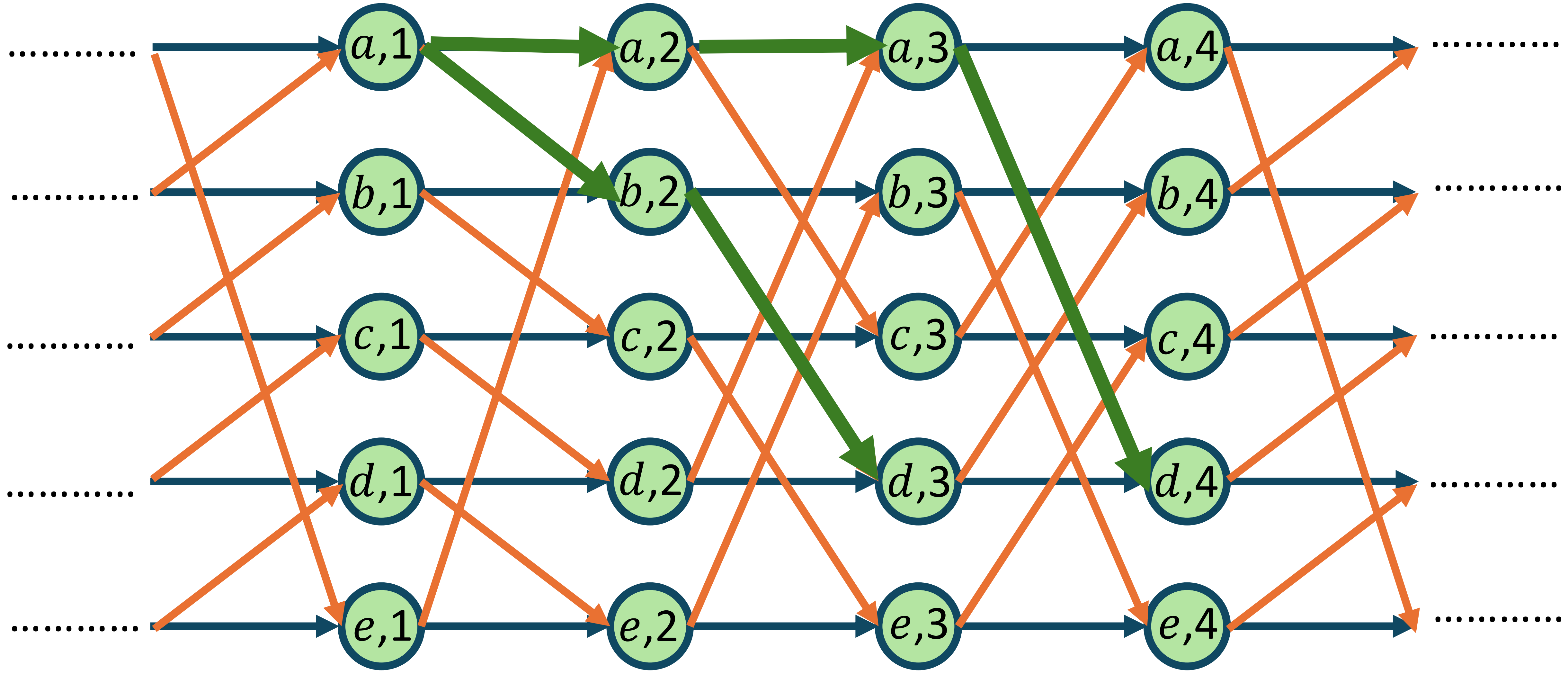


To route $a \rightarrow d$ starting at $t = 1$,



To route $a \rightarrow d$ starting at $t = 1$,

Build oblivious routing protocol with *bounded max latency* L



Congestion

At each timestep t we will receive arbitrary permutation demand D_{σ_t}

- $\forall a$, send 1 unit of flow from $a \rightarrow \sigma_t(a)$ starting at timestep t.

An oblivious routing protocol *guarantees* max congestion c if $\forall D_{\sigma_t}$ across all time, the max flow traversing any physical edge is $\leq c$.

Congestion on virtual edges is ignored

**If flow is balanced evenly across edges,

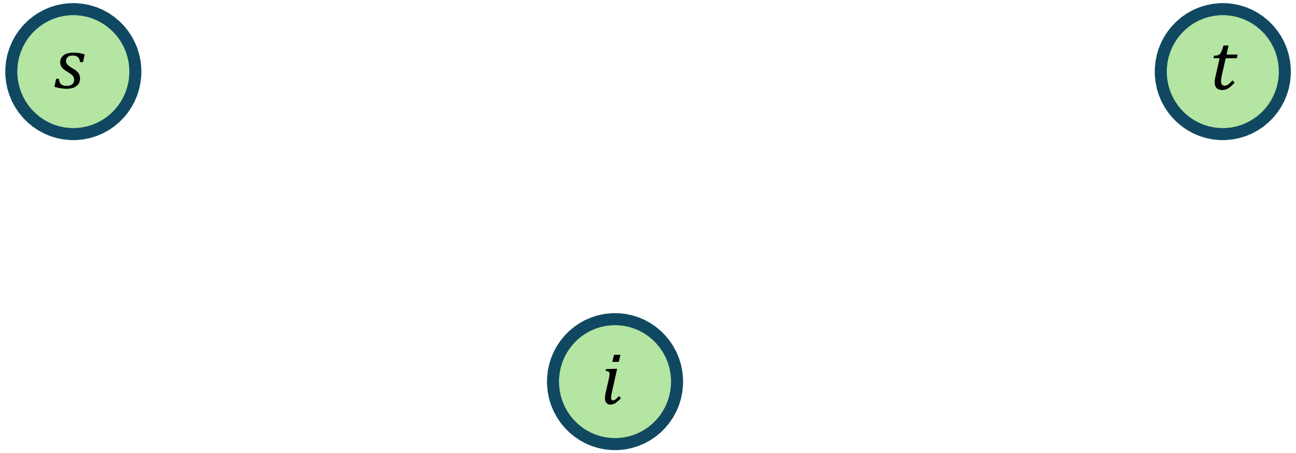
max congestion = average physical hop count

[Valiant & Brebner, STOC‘81]

- To route *obliviously* from $s \to t$,

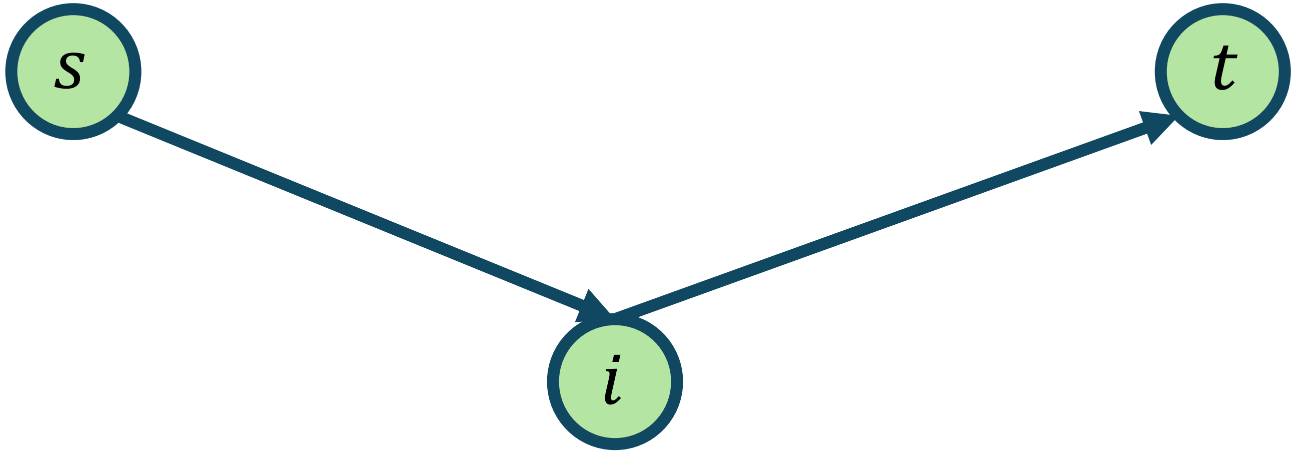
[Valiant & Brebner, STOC‘81]

- To route *obliviously* from $s \rightarrow t$,
  - Sample intermediate node i uniformly at random



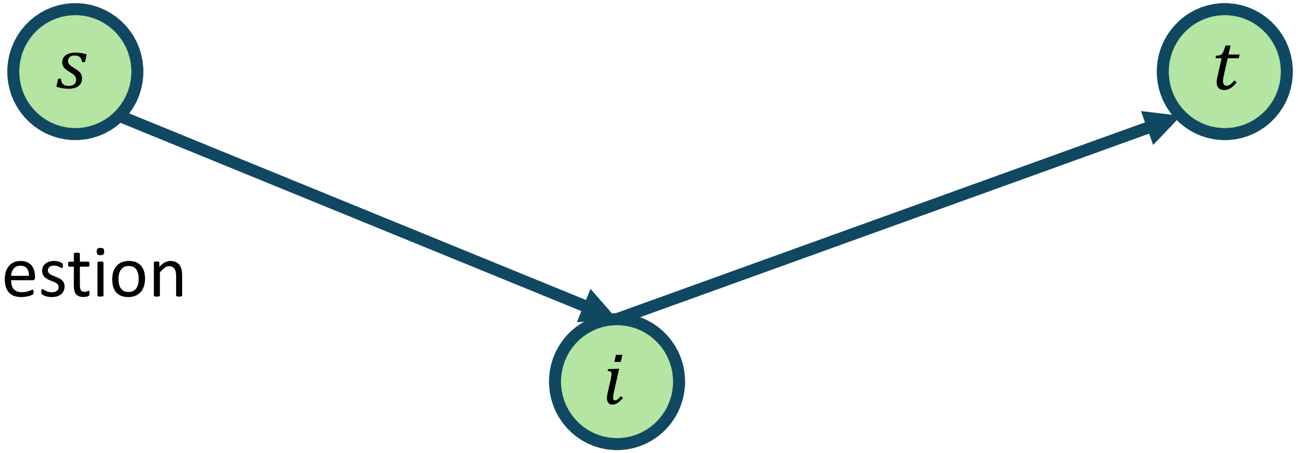
[Valiant & Brebner, STOC‘81]

- To route *obliviously* from $s \to t$,
  - Sample intermediate node i uniformly at random
  - Route on direct (shortest) paths from $s \to i$ then $i \to t$



[Valiant & Brebner, STOC‘81]

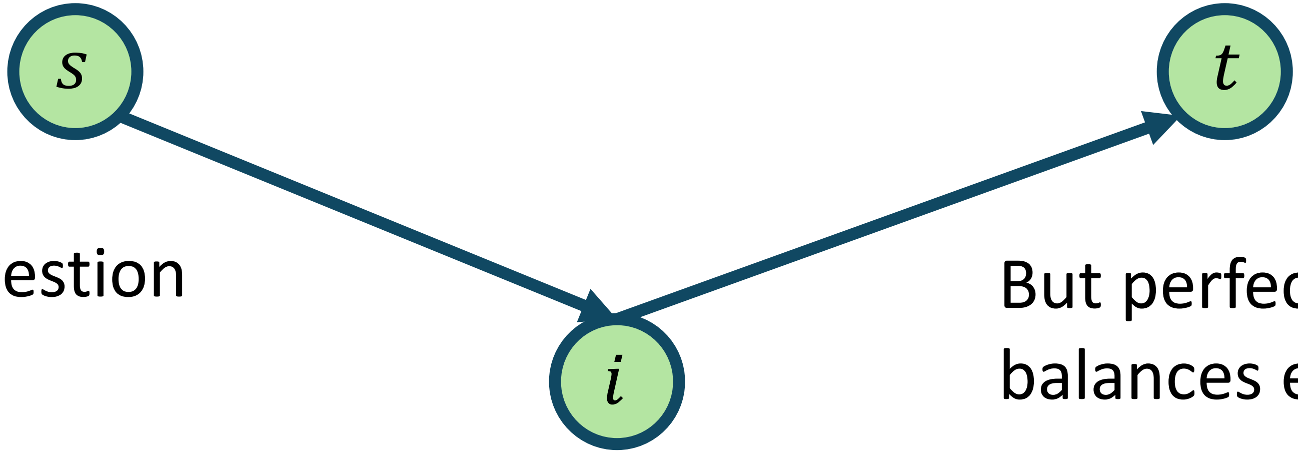
- To route *obliviously* from $s \to t$,
  - Sample intermediate node i uniformly at random
  - Route on direct (shortest) paths from $s \to i$ then $i \to t$



Doubles congestion
in network

[Valiant & Brebner, STOC'81]

- To route *obliviously* from $s \rightarrow t$,
    - Sample intermediate node i uniformly at random
    - Route on direct (shortest) paths from $s \rightarrow i$ then $i \rightarrow t$



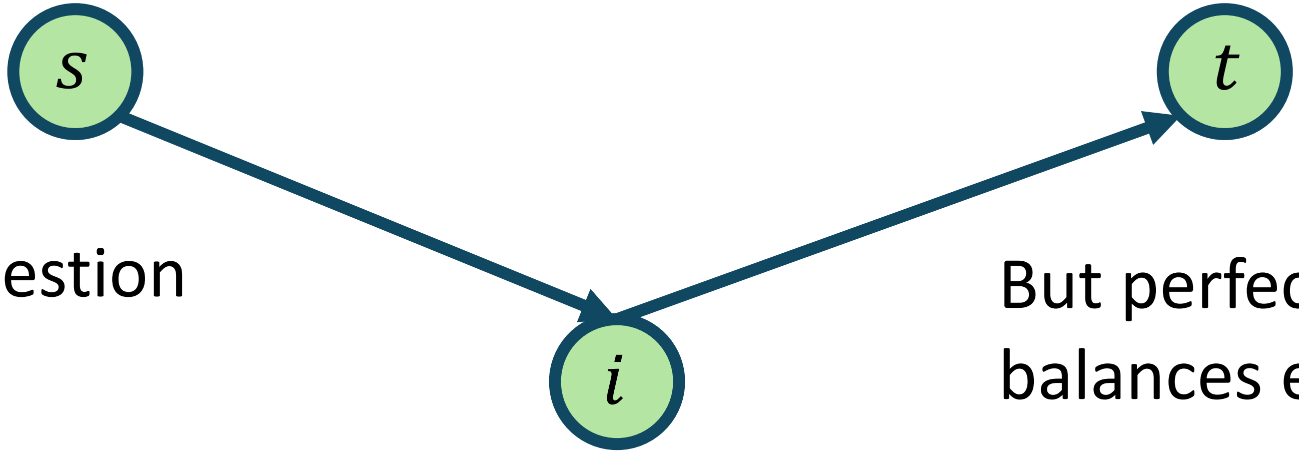
Doubles congestion in network

But perfectly load balances edges

Valiant Load Balancing

[Valiant & Brebner, STOC‘81]

- To route *obliviously* from $s \to t$,
  - Sample intermediate node i uniformly at random
  - Route on direct (shortest) paths from $s \to i$ then $i \to t$



Doubles congestion in network

But perfectly load balances edges

VLB Factor 2 Overprovisioning is Optimal for:

- Static networks with fixed-capacity links
  [Shen & McKeown’05][KCML’05]
  [Babaioff & Chuang ’07]

VLB Factor 2 Overprovisioning is Optimal for:

- Static networks with fixed-capacity links
  [Shen & McKeown’05][KCML’05]
  [Babaioff & Chuang ’07]
- **Reconfigurable networks** with bounded maximum latency
  [A**W**SKWA’22]

VLB Factor 2 Overprovisioning is Optimal for:

- Static networks with fixed-capacity links
  [Shen & McKeown’05][KCML’05]
  [Babaioff & Chuang ’07]
- **Reconfigurable networks** with bounded maximum latency
  [A**W**SKWA’22]

How to improve?

Räcke’s Hierarchical Tree Decomposition

[Räcke, STOC‘08]

- $\mathrm{O}(\log n)$-competitive and optimal oblivious routing protocol for *general networks*

- For optimized topologies in datacenters, even factor 2 overprovisioning is undesirable

We show that the *ability to randomize* a reconfigurable network allows oblivious routing protocols that break the “VLB Barrier”

Given a latency bound of $\tilde{O}\left(gN^{1/g}\right)$ for integer g:

Goal	Average Hop Count	Congestion	
Full Network Connectivity (lower bound)	g	$-$	Naïve counting

Given a latency bound of $\tilde{O}\left(gN^{1/g}\right)$ for integer g:

Goal	Average Hop Count	Congestion	
Full Network Connectivity (lower bound)	g	–	Naïve counting
Uniform Multicommodity Flow	g	g	[AWSKWA'22]
Oblivious Routing (prob. 1)	$2g$	$2g$	[AWSKWA'22] (uses VLB)

Given a latency bound of $\tilde{O}\left(gN^{1/g}\right)$ for integer g:

Goal	Average Hop Count	Congestion	
Full Network Connectivity (lower bound)	g	$-$	Naïve counting
Uniform Multicommodity Flow	g	g	[A**W**SKWA’22]
Oblivious Routing (prob. 1)	$2g$	$2g$	[A**W**SKWA’22] (uses VLB)
Oblivious Routing (w.h.p.)	$g+1$	$g+1+\delta$ $\forall\delta>0$	**This work**

Probability that the congestion bound is violated is *negligible* in the network size

High-Level Overview

- Instead of routing to uniform random node i
  - Take a single physical hop to random neighbor
  - Then route on a shortest path to destination
  - Use randomness of connection schedule to prove load is effectively balanced
- Analysis relies on a complicated tail bound
  - Bilinear form on an orbit of a permutation group action
  - Negative association + suitable decomposition and conditioning

Given a latency bound of $\tilde{O}\left(gN^{1/g}\right)$ for integer g:

Goal	Average Hop Count	Congestion	
Full Network Connectivity (lower bound)	g	–	Naïve counting
Uniform Multicommodity Flow	g	g	[AWSKWA'22]
Oblivious Routing (prob. 1)	$2g$	$2g$	[AWSKWA'22] (uses VLB)
Oblivious Routing (w.h.p.)	$g+1$	$g+1+\delta$ $\forall\delta > 0$	**This work**
Semi-Oblivious Routing (prob. 1)	$g+1$	$g+1+\delta$ $\forall\delta > 0$	**This work**

Thank you!

teganwilson@cs.cornell.edu