Peer-to-peer energy-aware tree network formation

Adelina Madhja
Sotiris Nikoletseas
Dimitrios Tsolovos
Alexandros Voudouris

The problem

- A population of nodes
- Each node has some initial energy, *limited memory*, and moves arbitrarily in a bounded networking area
- When two nodes meet, they **interact**
  - create network connections and exchange energy
- Goal: define interaction protocols so that the nodes are able to form **tree networks** and achieve desired **energy distributions**

Related work

- **Population protocols**
  - Angluin, Aspnes, Eisenstat, and Ruppert (Distributed Computing 2007)
- **Network formation among computationally weak agents**
  - Michail and Spyrakis (Distributed Computing 2016)
- **Peer-to-peer energy balance**
  - Nikoletseas, Raptis and Raptopoulos (MASS 2016)
- **Star formation + peer-to-peer energy balance**
  - Madhja, Nikoletseas, Raptopoulos, and Tsolovos (MSWIM 2016)

Modeling assumption

- Mobility can be abstracted by assuming that all interactions between nodes are planned by a **fair probabilistic scheduler**
- At each time step, a pair of nodes is selected independently and uniformly at random among all possible pairs

Subgoals

- Goal: define interaction protocols so that the nodes are able to form **tree networks** and achieve desired **energy distributions**
- Subgoal 1: tree network formation
  - **Arbitrary** and **binary** spanning trees
- Subgoal 2: peer-to-peer energy exchange
  - **Exact, relaxed,** and **exact up to the root** energy distributions

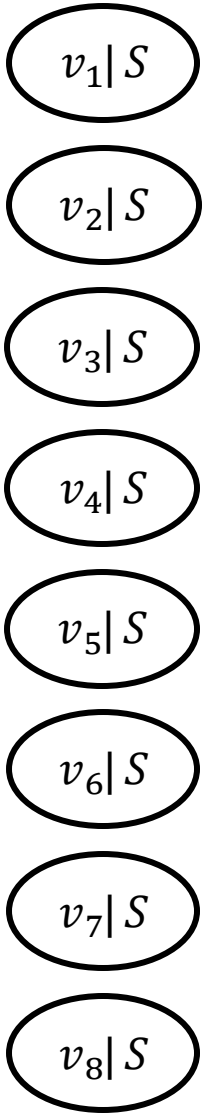
Arbitrary tree network formation

- A node can be
  - **isolated** if it is not connected to any other node (S)
  - **leaf** if it has a parent, but no children (L)
  - **internal** if it has a parent and children (I)
  - **root** if it does not have parent, but has children (R)

- When two nodes (u, v) interact, v becomes a child of u if
  - v is isolated, or
  - both u and v are roots

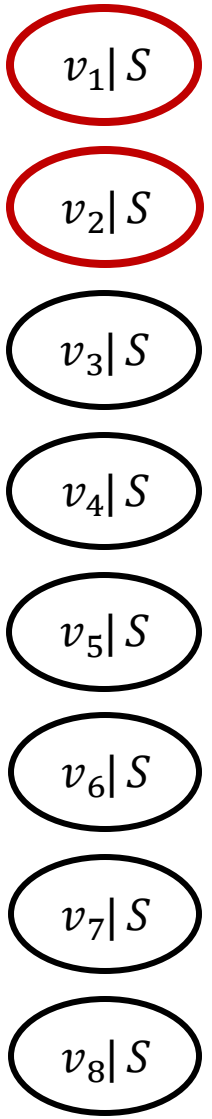
Arbitrary tree network formation: example

scheduler
(v_1, v_2)
(v_3, v_4)
(v_1, v_4)
(v_3, v_5)
(v_1, v_2)
(v_1, v_3)
(v_2, v_6)
(v_7, v_8)
(v_1, v_7)



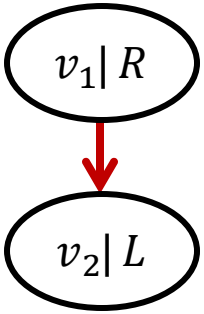
Arbitrary tree network formation: example

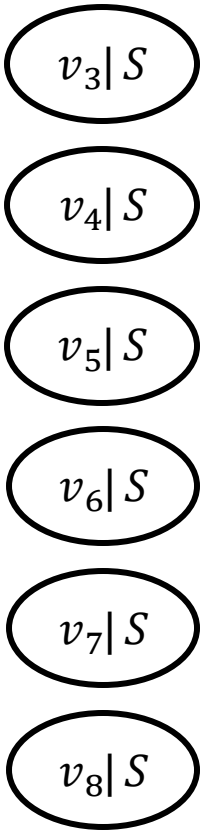
scheduler
(v_1, v_2)
(v_3, v_4)
(v_1, v_4)
(v_3, v_5)
(v_1, v_2)
(v_1, v_3)
(v_2, v_6)
(v_7, v_8)
(v_1, v_7)



Arbitrary tree network formation: example

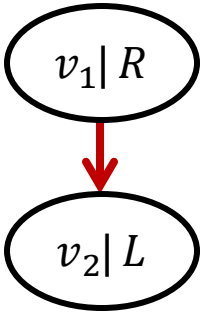
scheduler
(v_1, v_2)
(v_3, v_4)
(v_1, v_4)
(v_3, v_5)
(v_1, v_2)
(v_1, v_3)
(v_2, v_6)
(v_7, v_8)
(v_1, v_7)

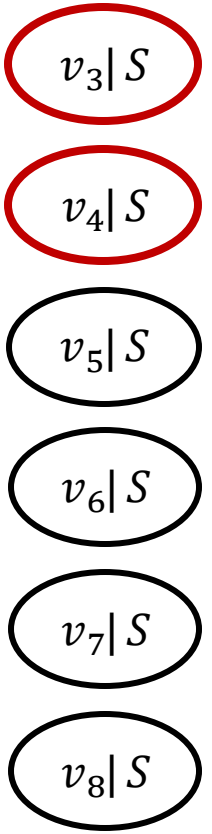




Arbitrary tree network formation: example

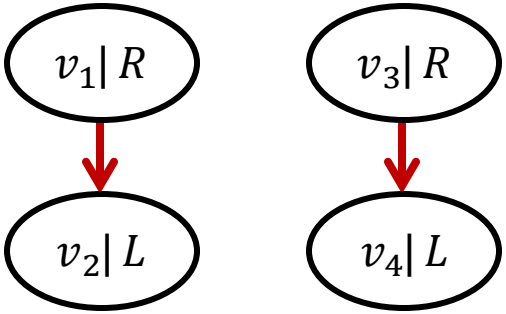
scheduler
(v_1, v_2)
(v_3, v_4)
(v_1, v_4)
(v_3, v_5)
(v_1, v_2)
(v_1, v_3)
(v_2, v_6)
(v_7, v_8)
(v_1, v_7)

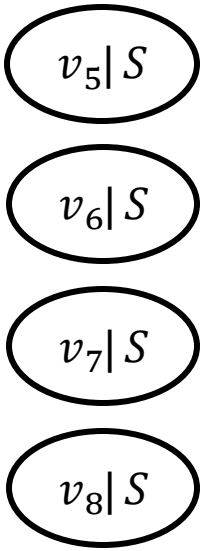




Arbitrary tree network formation: example

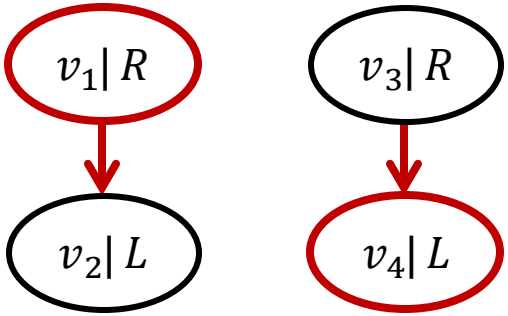
scheduler
(v_1, v_2)
(v_3, v_4)
(v_1, v_4)
(v_3, v_5)
(v_1, v_2)
(v_1, v_3)
(v_2, v_6)
(v_7, v_8)
(v_1, v_7)

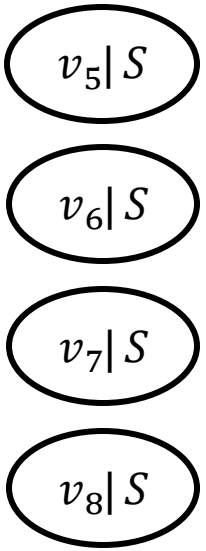




Arbitrary tree network formation: example

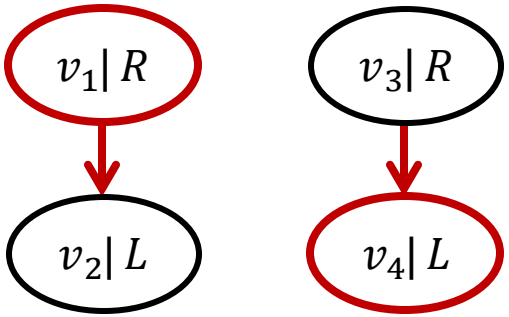
scheduler
(v_1, v_2)
(v_3, v_4)
$\boldsymbol{(v_1, v_4)}$
(v_3, v_5)
(v_1, v_2)
(v_1, v_3)
(v_2, v_6)
(v_7, v_8)
(v_1, v_7)





Arbitrary tree network formation: example

scheduler
(v_1, v_2)
(v_3, v_4)
$\boldsymbol{(v_1, v_4)}$
(v_3, v_5)
(v_1, v_2)
(v_1, v_3)
(v_2, v_6)
(v_7, v_8)
(v_1, v_7)



Connections not allowed between root and (leaf or internal) nodes

$v_5|\, S$

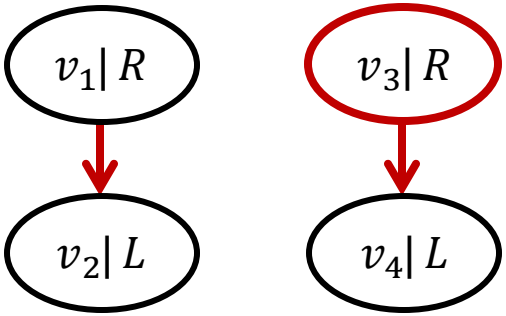
$v_6|\, S$

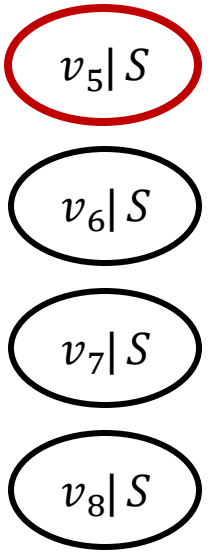
$v_7|\, S$

$v_8|\, S$

Arbitrary tree network formation: example

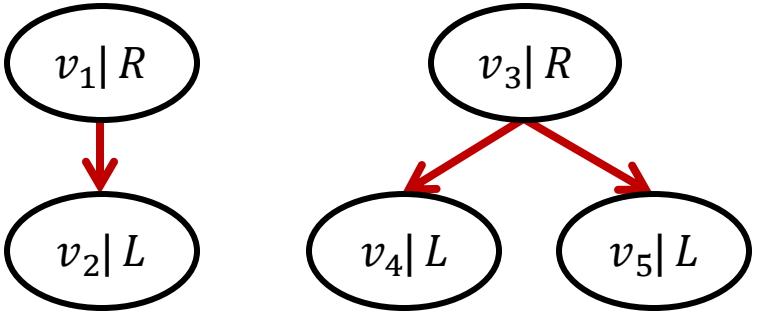
scheduler
(v_1, v_2)
(v_3, v_4)
(v_1, v_4)
$\boldsymbol{(v_3, v_5)}$
(v_1, v_2)
(v_1, v_3)
(v_2, v_6)
(v_7, v_8)
(v_1, v_7)

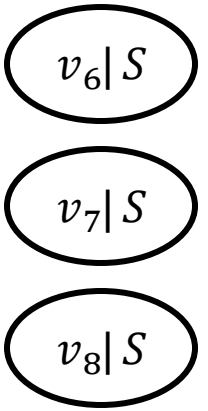




Arbitrary tree network formation: example

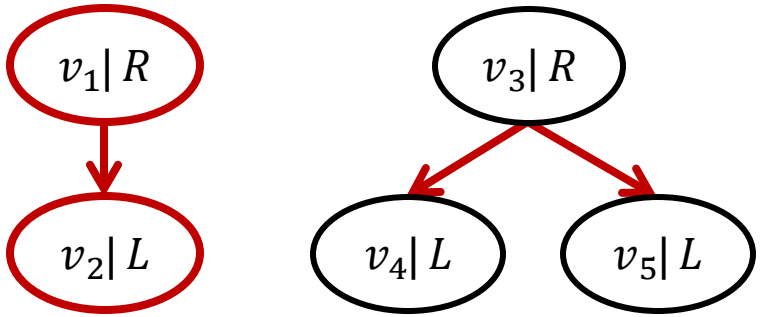
scheduler
(v_1, v_2)
(v_3, v_4)
(v_1, v_4)
$\boldsymbol{(v_3, v_5)}$
(v_1, v_2)
(v_1, v_3)
(v_2, v_6)
(v_7, v_8)
(v_1, v_7)

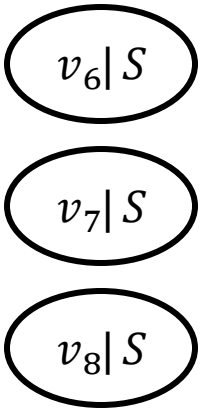




Arbitrary tree network formation: example

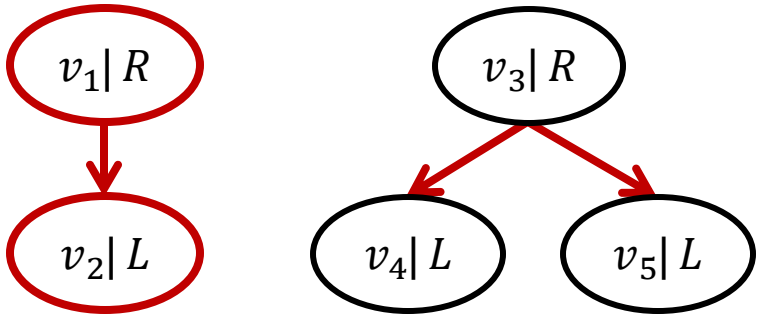
scheduler
(v_1, v_2)
(v_3, v_4)
(v_1, v_4)
(v_3, v_5)
$\boldsymbol{(v_1, v_2)}$
(v_1, v_3)
(v_2, v_6)
(v_7, v_8)
(v_1, v_7)



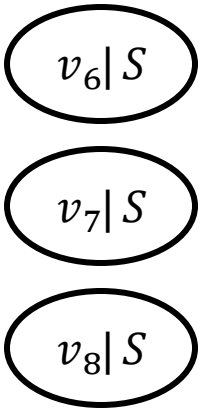


Arbitrary tree network formation: example

scheduler
(v_1, v_2)
(v_3, v_4)
(v_1, v_4)
(v_3, v_5)
$\boldsymbol{(v_1, v_2)}$
(v_1, v_3)
(v_2, v_6)
(v_7, v_8)
(v_1, v_7)

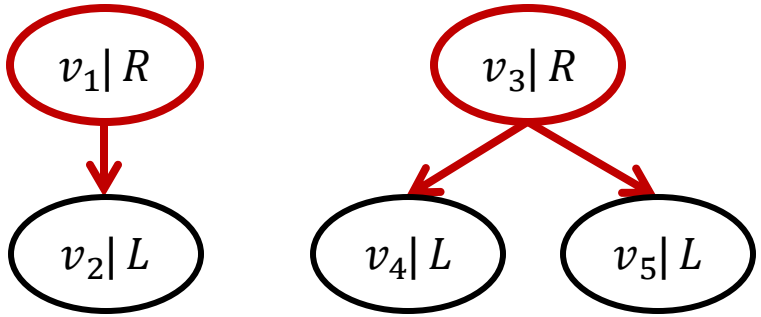


The scheduler might select the same pair again; nothing happens



Arbitrary tree network formation: example

scheduler
(v_1, v_2)
(v_3, v_4)
(v_1, v_4)
(v_3, v_5)
(v_1, v_2)
$\boldsymbol{(v_1, v_3)}$
(v_2, v_6)
(v_7, v_8)
(v_1, v_7)



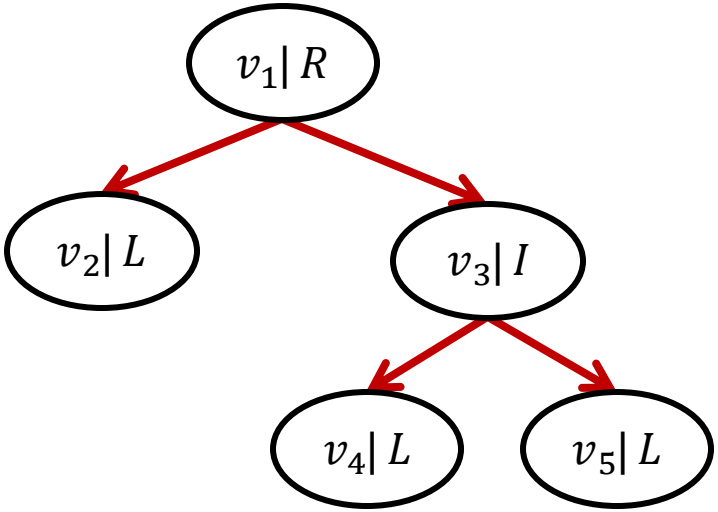
$v_6|\, S$

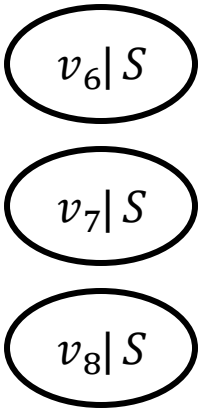
$v_7|\, S$

$v_8|\, S$

Arbitrary tree network formation: example

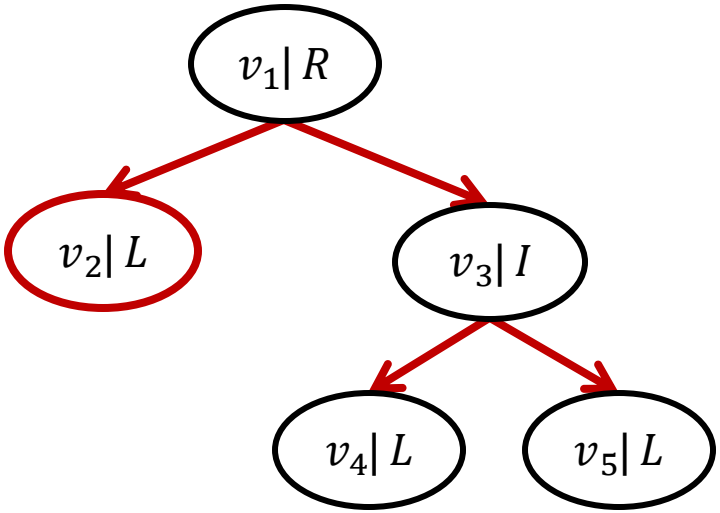
scheduler
(v_1, v_2)
(v_3, v_4)
(v_1, v_4)
(v_3, v_5)
(v_1, v_2)
$\boldsymbol{(v_1, v_3)}$
(v_2, v_6)
(v_7, v_8)
(v_1, v_7)

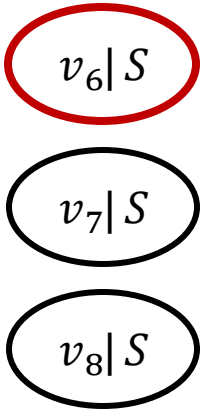




Arbitrary tree network formation: example

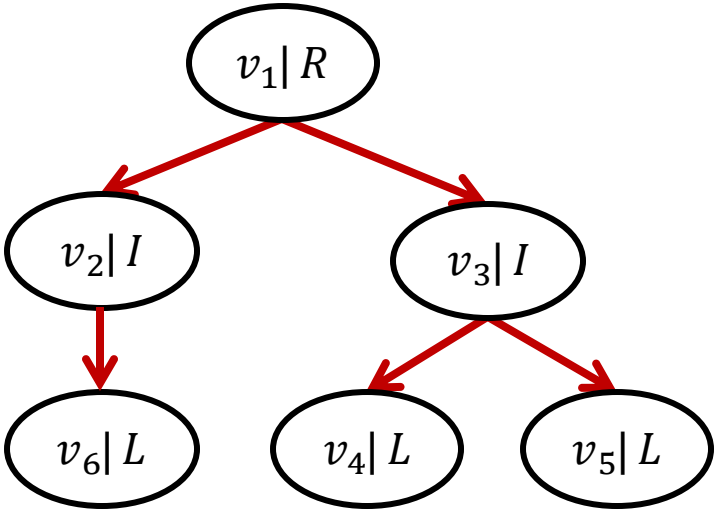
scheduler
(v_1, v_2)
(v_3, v_4)
(v_1, v_4)
(v_3, v_5)
(v_1, v_2)
(v_1, v_3)
$\boldsymbol{(v_2, v_6)}$
(v_7, v_8)
(v_1, v_7)

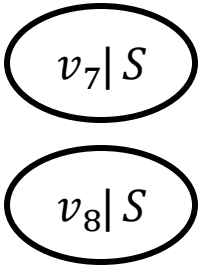




Arbitrary tree network formation: example

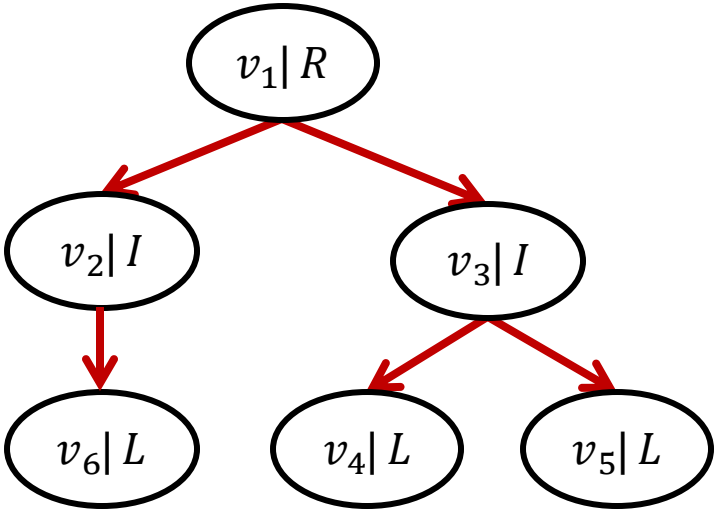
scheduler
(v_1, v_2)
(v_3, v_4)
(v_1, v_4)
(v_3, v_5)
(v_1, v_2)
(v_1, v_3)
$\boldsymbol{(v_2, v_6)}$
(v_7, v_8)
(v_1, v_7)

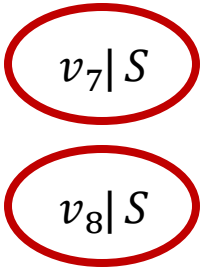




Arbitrary tree network formation: example

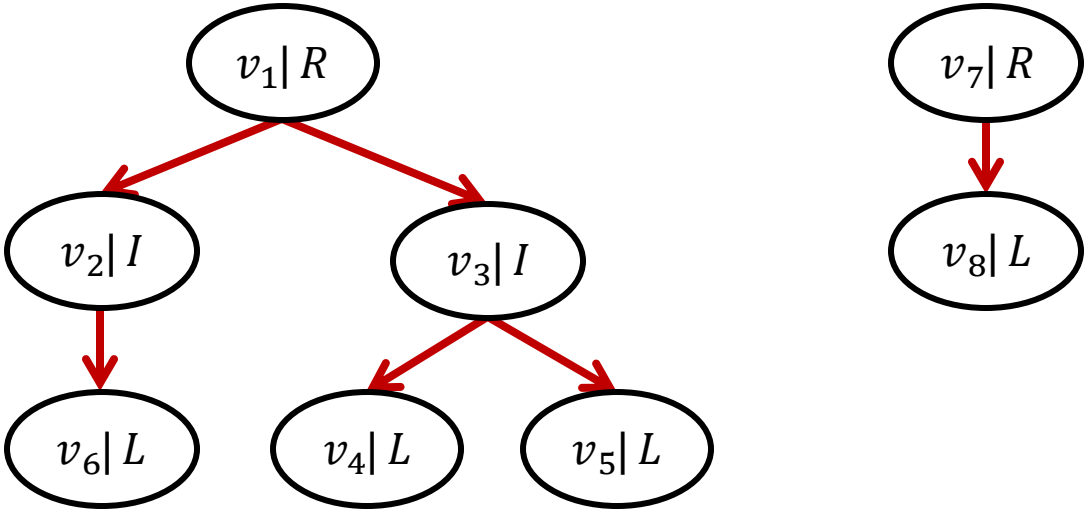
scheduler
(v_1, v_2)
(v_3, v_4)
(v_1, v_4)
(v_3, v_5)
(v_1, v_2)
(v_1, v_3)
(v_2, v_6)
(v_7, v_8)
(v_1, v_7)





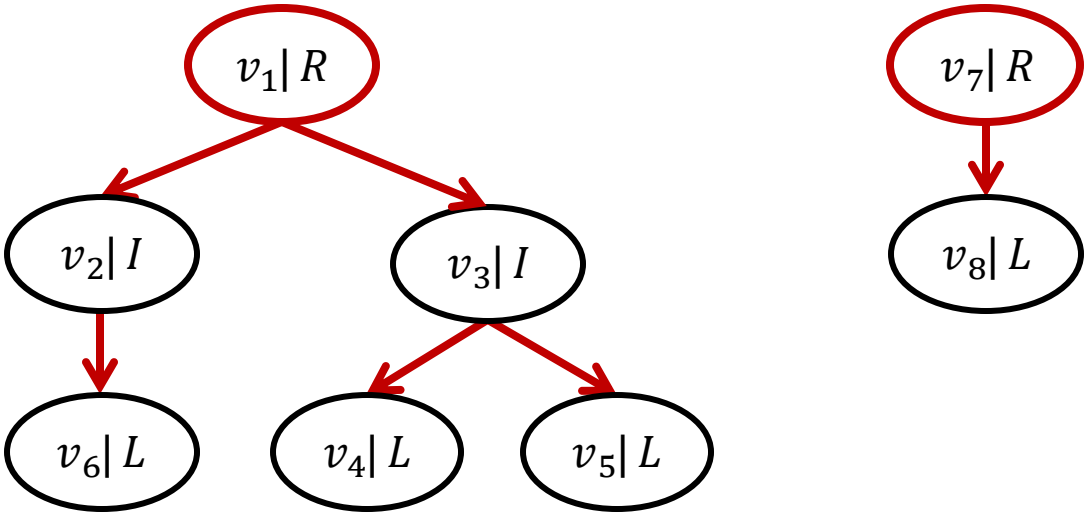
Arbitrary tree network formation: example

scheduler
(v_1, v_2)
(v_3, v_4)
(v_1, v_4)
(v_3, v_5)
(v_1, v_2)
(v_1, v_3)
(v_2, v_6)
(v_7, v_8)
(v_1, v_7)



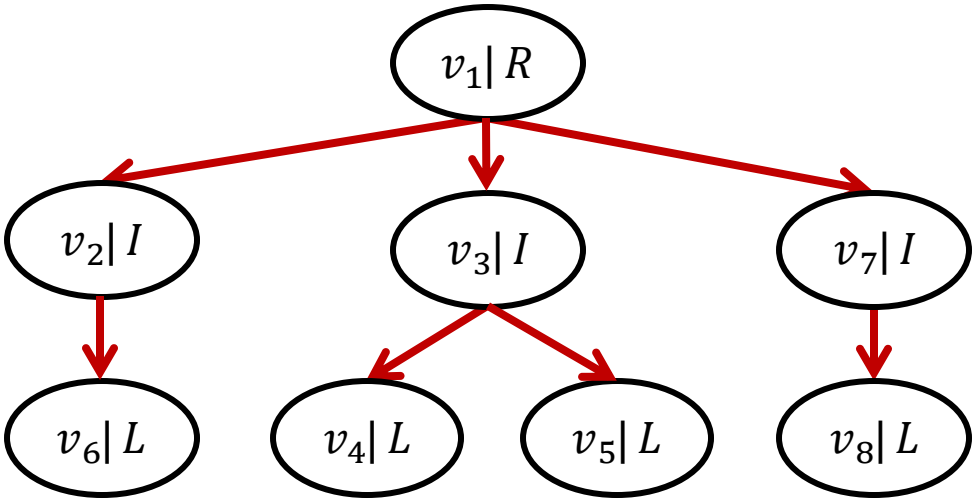
Arbitrary tree network formation: example

scheduler
(v_1, v_2)
(v_3, v_4)
(v_1, v_4)
(v_3, v_5)
(v_1, v_2)
(v_1, v_3)
(v_2, v_6)
(v_7, v_8)
$\boldsymbol{(v_1, v_7)}$



Arbitrary tree network formation: example

scheduler
(v_1, v_2)
(v_3, v_4)
(v_1, v_4)
(v_3, v_5)
(v_1, v_2)
(v_1, v_3)
(v_2, v_6)
(v_7, v_8)
$\boldsymbol{(v_1, v_7)}$

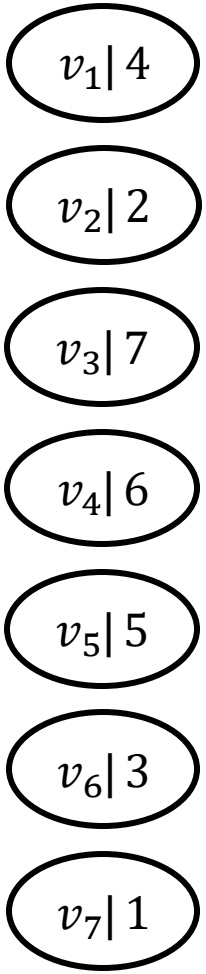


Binary tree network formation

- A node can be
  - **isolated**
  - **leaf**
  - **internal with one child**
  - **Internal with two children**
  - **root with one child**
  - **root with two children**
- Each node v stores an initially **unique random number w_v**
- When two nodes (u, v) interact, v becomes a child of u if
  - v is isolated and u has less than two children, or
  - v is a root, u has less than two children, and $w_u < w_v$
- When a parent-child pair interacts, the child adopts the number stored in the parent

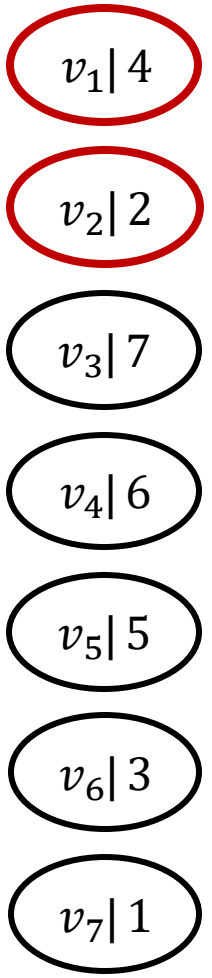
Binary tree network formation: example

scheduler
(v_1, v_2)
(v_3, v_4)
(v_7, v_6)
(v_1, v_4)
(v_5, v_1)
(v_5, v_6)
(v_2, v_7)
(v_4, v_3)
(v_1, v_6)



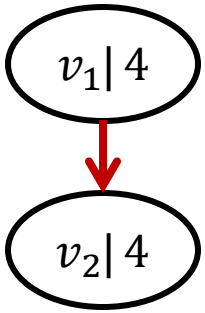
Binary tree network formation: example

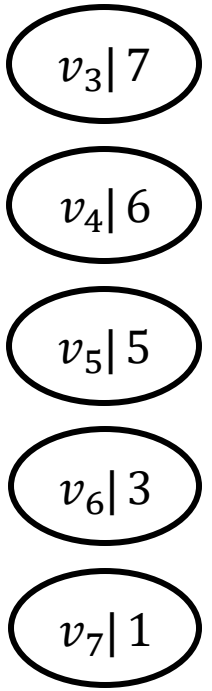
scheduler
(v_1, v_2)
(v_3, v_4)
(v_7, v_6)
(v_1, v_4)
(v_5, v_1)
(v_5, v_6)
(v_2, v_7)
(v_4, v_3)
(v_1, v_6)



Binary tree network formation: example

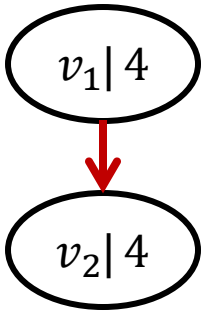
scheduler
(v_1, v_2)
(v_4, v_3)
(v_7, v_6)
(v_1, v_4)
(v_5, v_1)
(v_5, v_6)
(v_2, v_7)
(v_4, v_3)
(v_1, v_6)

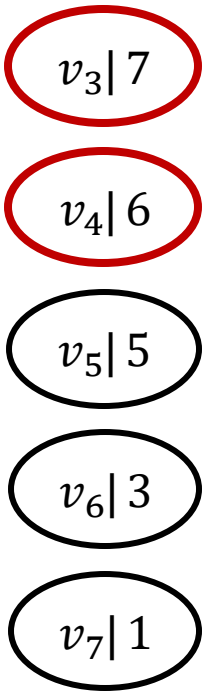




Binary tree network formation: example

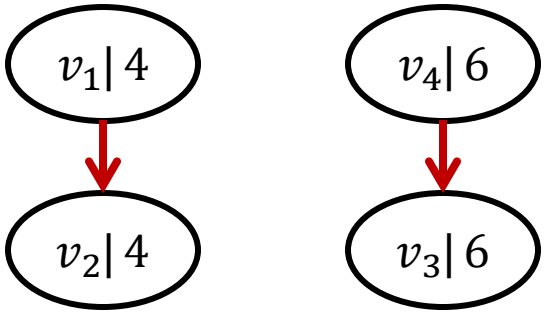
scheduler
(v_1, v_2)
(v_4, v_3)
(v_7, v_6)
(v_1, v_4)
(v_5, v_1)
(v_5, v_6)
(v_2, v_7)
(v_4, v_3)
(v_1, v_6)

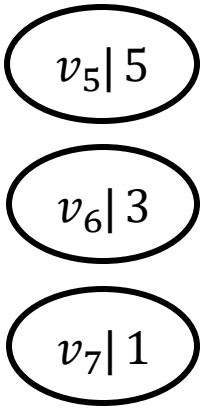




Binary tree network formation: example

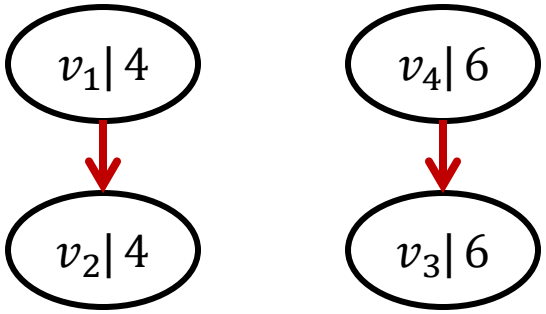
scheduler
(v_1, v_2)
(v_4, v_3)
(v_7, v_6)
(v_1, v_4)
(v_5, v_1)
(v_5, v_6)
(v_2, v_7)
(v_4, v_3)
(v_1, v_6)

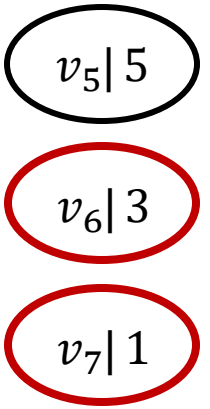




Binary tree network formation: example

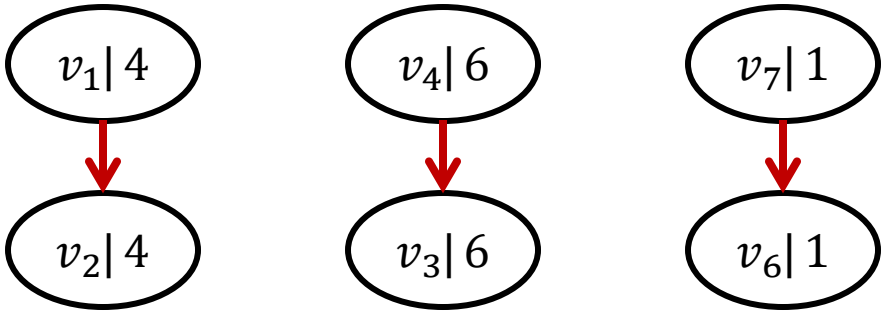
scheduler
(v_1, v_2)
(v_4, v_3)
(v_7, v_6)
(v_1, v_4)
(v_5, v_1)
(v_5, v_6)
(v_2, v_7)
(v_4, v_3)
(v_1, v_6)

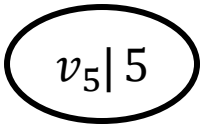




Binary tree network formation: example

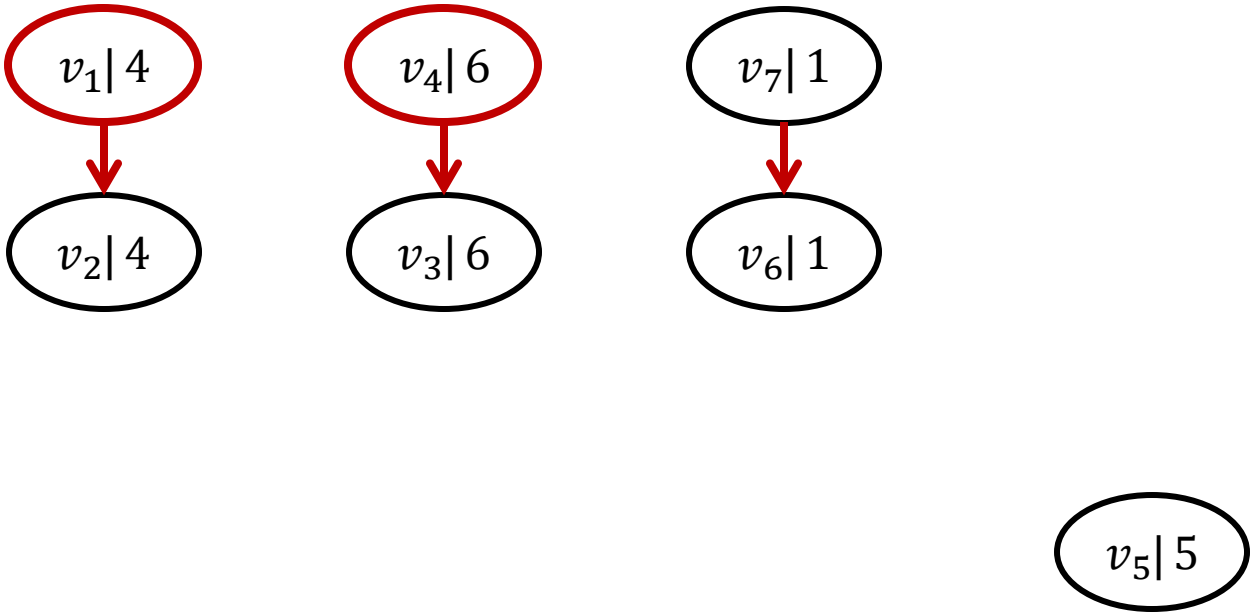
scheduler
(v_1, v_2)
(v_4, v_3)
(v_7, v_6)
(v_1, v_4)
(v_5, v_1)
(v_5, v_6)
(v_2, v_7)
(v_4, v_3)
(v_1, v_6)





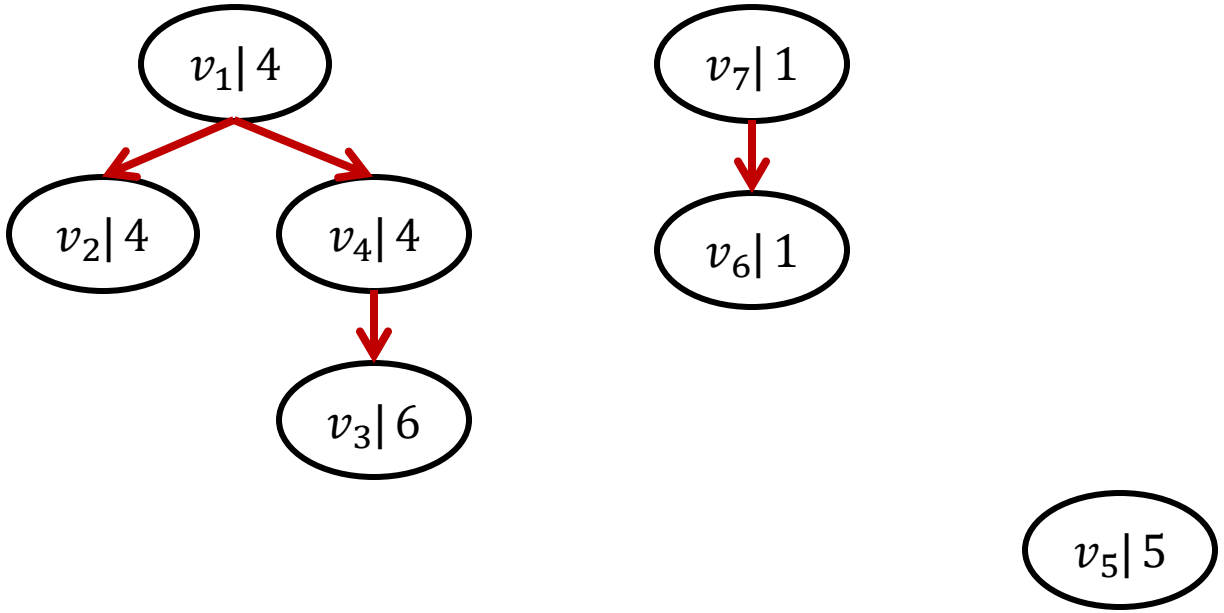
Binary tree network formation: example

scheduler
(v_1, v_2)
(v_4, v_3)
(v_7, v_6)
$\boldsymbol{(v_1, v_4)}$
(v_5, v_1)
(v_5, v_6)
(v_2, v_7)
(v_4, v_3)
(v_1, v_6)



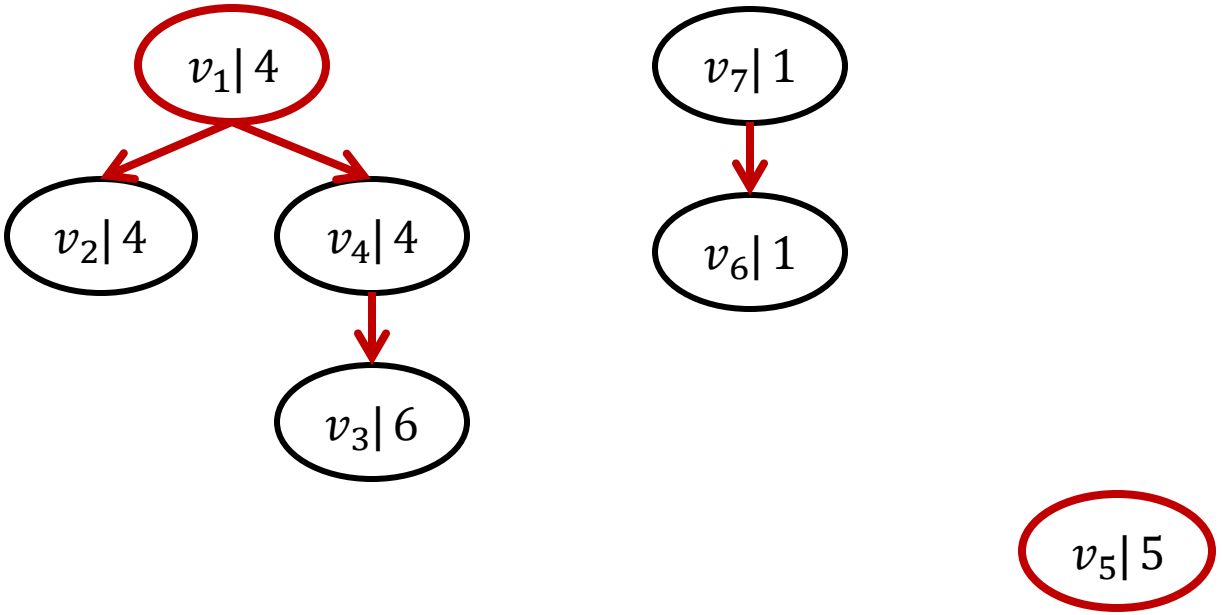
Binary tree network formation: example

scheduler
(v_1, v_2)
(v_4, v_3)
(v_7, v_6)
$\boldsymbol{(v_1, v_4)}$
(v_5, v_1)
(v_5, v_6)
(v_2, v_7)
(v_4, v_3)
(v_1, v_6)



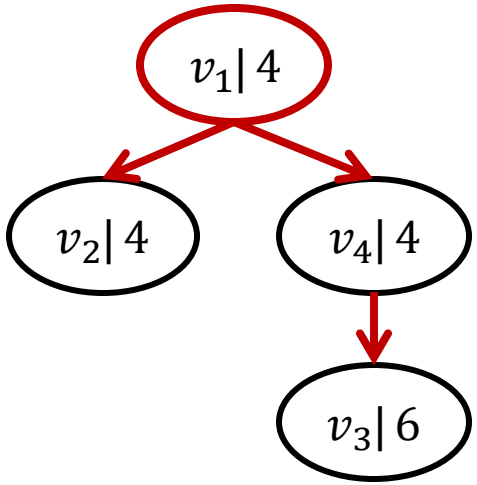
Binary tree network formation: example

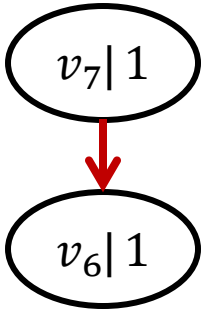
scheduler
(v_1, v_2)
(v_4, v_3)
(v_7, v_6)
(v_1, v_4)
$\boldsymbol{(v_5, v_1)}$
(v_5, v_6)
(v_2, v_7)
(v_4, v_3)
(v_1, v_6)



Binary tree network formation: example

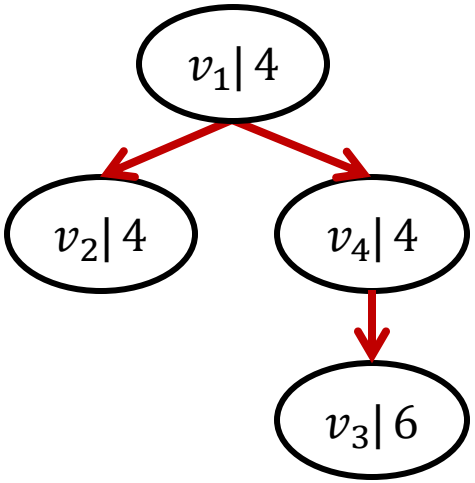
scheduler
(v_1, v_2)
(v_4, v_3)
(v_7, v_6)
(v_1, v_4)
$\boldsymbol{(v_5, v_1)}$
(v_5, v_6)
(v_2, v_7)
(v_4, v_3)
(v_1, v_6)

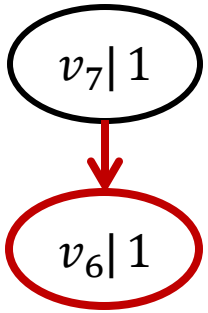




Binary tree network formation: example

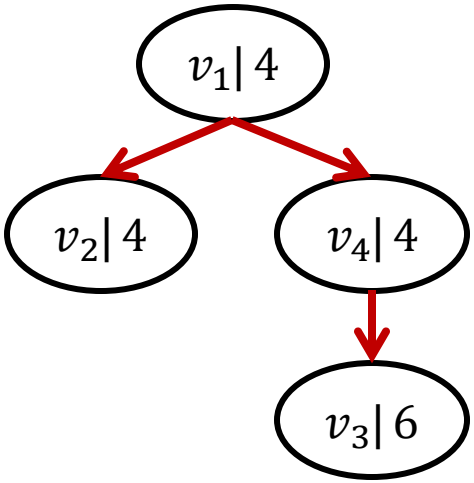
scheduler
(v_1, v_2)
(v_4, v_3)
(v_7, v_6)
(v_1, v_4)
(v_5, v_1)
$\boldsymbol{(v_5, v_6)}$
(v_2, v_7)
(v_4, v_3)
(v_1, v_6)

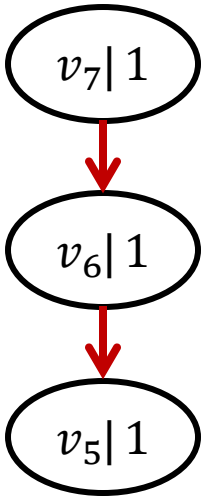




Binary tree network formation: example

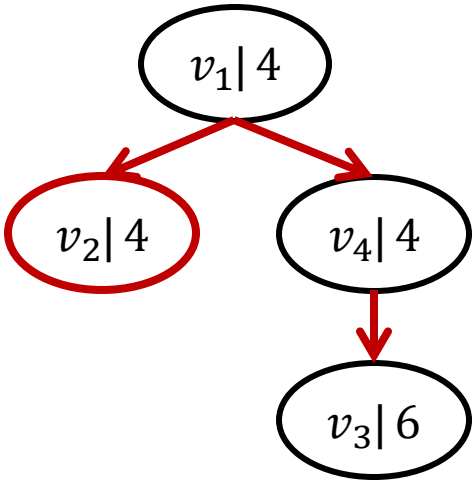
scheduler
(v_1, v_2)
(v_4, v_3)
(v_7, v_6)
(v_1, v_4)
(v_5, v_1)
$\boldsymbol{(v_5, v_6)}$
(v_2, v_7)
(v_4, v_3)
(v_1, v_6)

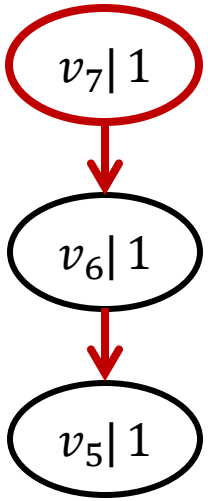




Binary tree network formation: example

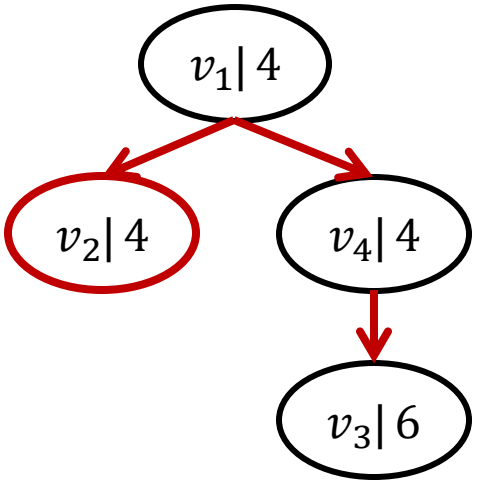
scheduler
(v_1, v_2)
(v_4, v_3)
(v_7, v_6)
(v_1, v_4)
(v_5, v_1)
(v_5, v_6)
$\boldsymbol{(v_2, v_7)}$
(v_4, v_3)
(v_1, v_6)

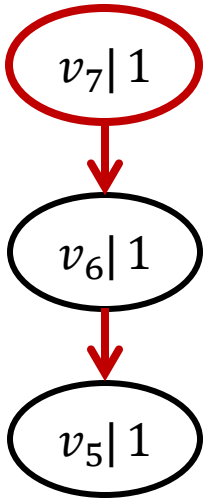


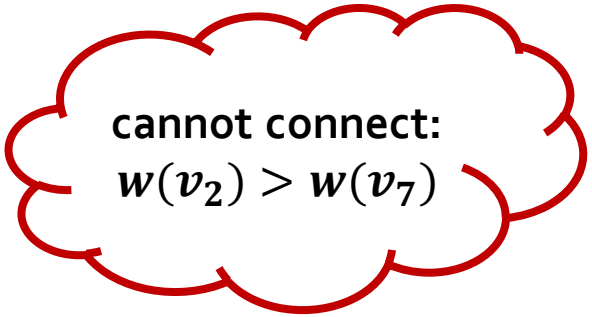


Binary tree network formation: example

scheduler
(v_1, v_2)
(v_4, v_3)
(v_7, v_6)
(v_1, v_4)
(v_5, v_1)
(v_5, v_6)
$\boldsymbol{(v_2, v_7)}$
(v_4, v_3)
(v_1, v_6)

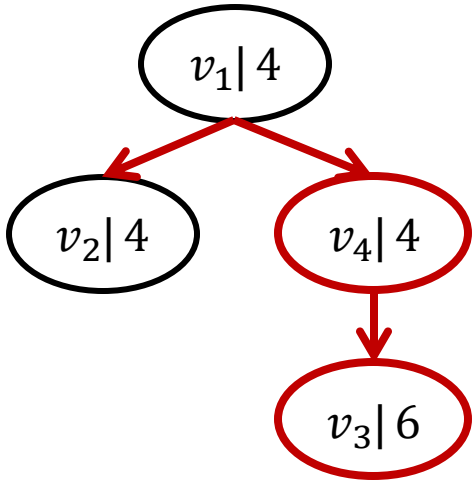


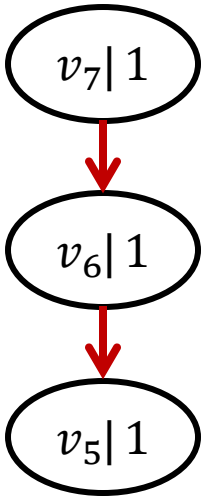




Binary tree network formation: example

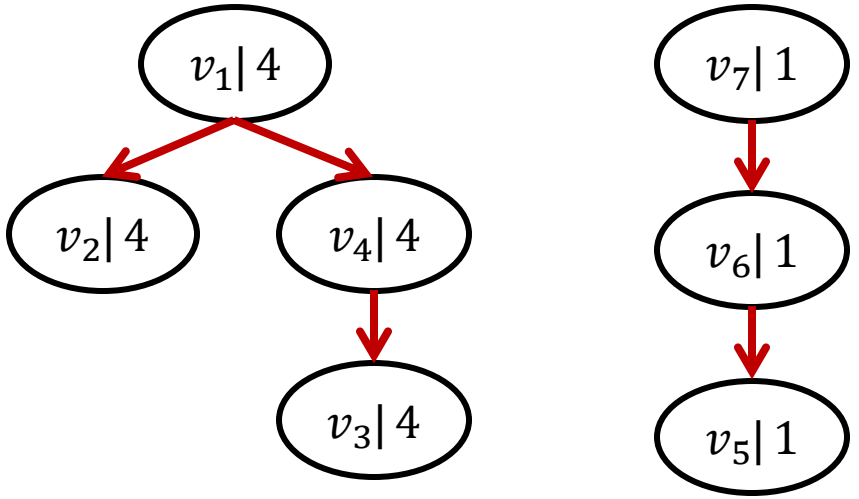
scheduler
(v_1, v_2)
(v_4, v_3)
(v_7, v_6)
(v_1, v_4)
(v_5, v_1)
(v_5, v_6)
(v_2, v_7)
$\boldsymbol{(v_4, v_3)}$
(v_1, v_6)





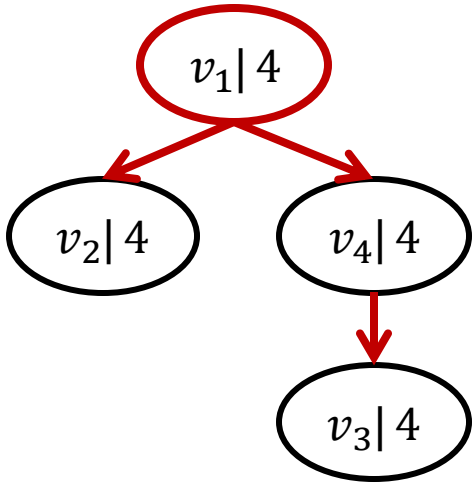
Binary tree network formation: example

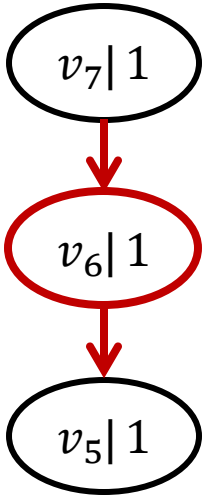
scheduler
(v_1, v_2)
(v_4, v_3)
(v_7, v_6)
(v_1, v_4)
(v_5, v_1)
(v_5, v_6)
(v_2, v_7)
$\boldsymbol{(v_4, v_3)}$
(v_1, v_6)



Binary tree network formation: example

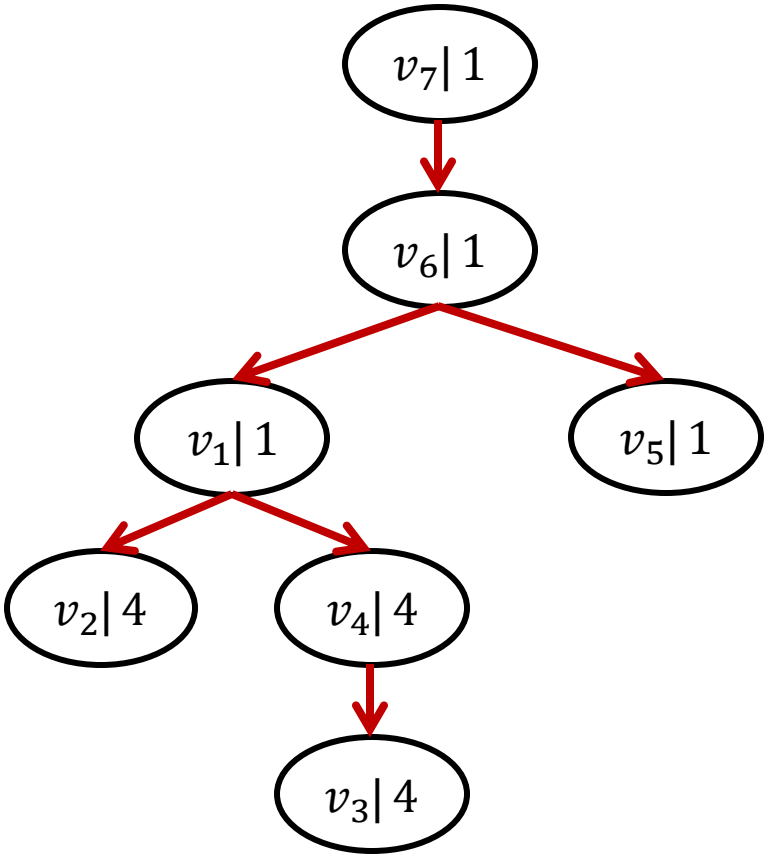
scheduler
(v_1, v_2)
(v_4, v_3)
(v_7, v_6)
(v_1, v_4)
(v_5, v_1)
(v_5, v_6)
(v_2, v_7)
(v_4, v_3)
$\boldsymbol{(v_1, v_6)}$





Binary tree network formation: example

scheduler
(v_1, v_2)
(v_4, v_3)
(v_7, v_6)
(v_1, v_4)
(v_5, v_1)
(v_5, v_6)
(v_2, v_7)
(v_4, v_3)
$\boldsymbol{(v_1, v_6)}$



Energy distributions

- Nodes of **lower depth** (closer to the root of the tree network) should have **more energy than** nodes of **higher depth**
- Motivation by **data propagation**
  - Nodes closer to the data sink are involved in almost all propagations and need more energy to operate properly
- **Exact distribution:** every node has **exactly twice** the energy of each of its children
- **Relaxed distribution:** every node has **at least twice** the energy of each of its children
- **Exact up to the root:** every node has **exactly twice** the energy of each of its children, **except** possibly for the root of the tree

Scenarios considered

- **Lossless case:** no energy is lost during any exchange
- **Lossy case:** some fraction of the energy that should be exchanged is lost
- **Scenario (FK):** full knowledge of network and total initial energy
- **Scenario (NK):** no knowledge
- **Scenario (PK):** knowledge of total initial energy only

Synopsis of theoretical results

- **Lossless scenario (FK):** *ideal-target protocol*
  - Guaranteed to converge to the unique exact distribution
- **Lossless scenario (NK):** *λ-exchange protocol*
  - Not guaranteed to converge to the exact distribution
  - Guaranteed to converge to a relaxed distribution when the network is a line (true in general via simulations)
- **Lossless scenario (PK):** *depth-target protocol* (for binary trees)
  - Guaranteed to converge to an exact up to the root distribution
- **Lossy scenarios (FK), (NK), (PK)**
  - Simulations

The ideal-target protocol

- Uses the tree network structure and the initial total energy (FK)
- Each node v can (locally) infer its **depth d_v** and the **height of the tree h**
- Sets as target energy

$$\gamma_v = 2^{h-d_v} \frac{E_{total}}{\sum_{d=0}^{h} n_d \cdot 2^{h-d}} = \gamma(d_v)$$

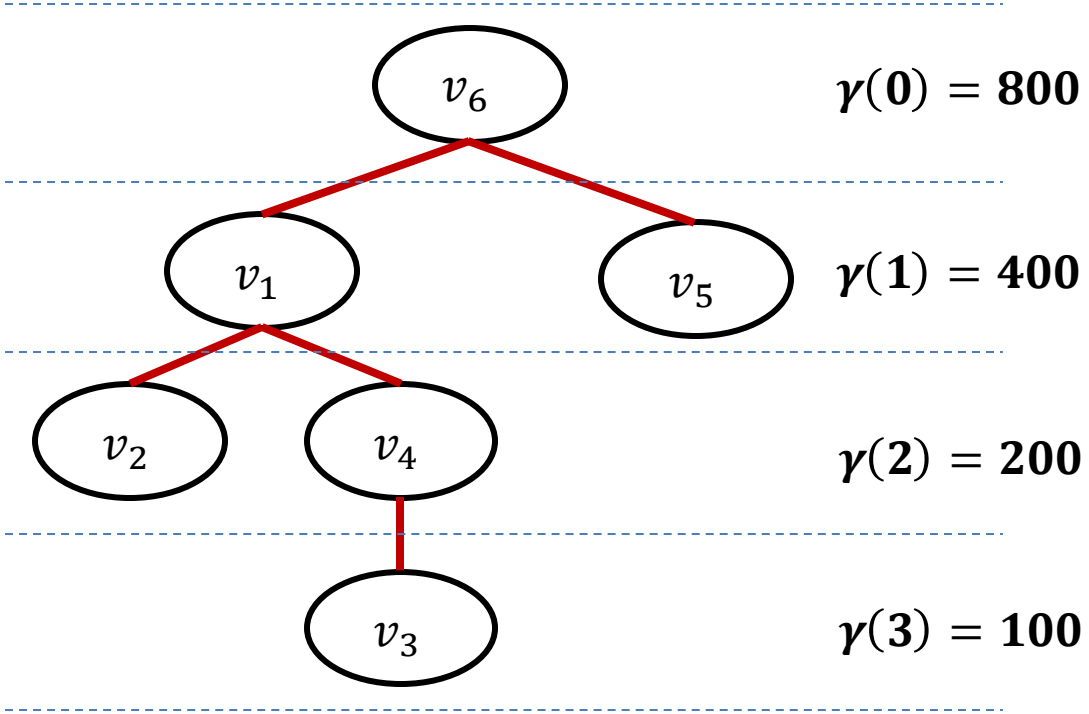
- Node v **requests** energy if its current is **strictly less than γ_v**
- Node v **can give** energy if its current is **strictly more than γ_v**

The ideal-target protocol: example

$E_{total} = 2100$

d	n_d
0	1
1	2
2	2
$h = 3$	1

v_1	500
v_2	100
v_3	150
v_4	400
v_5	350
v_6	600

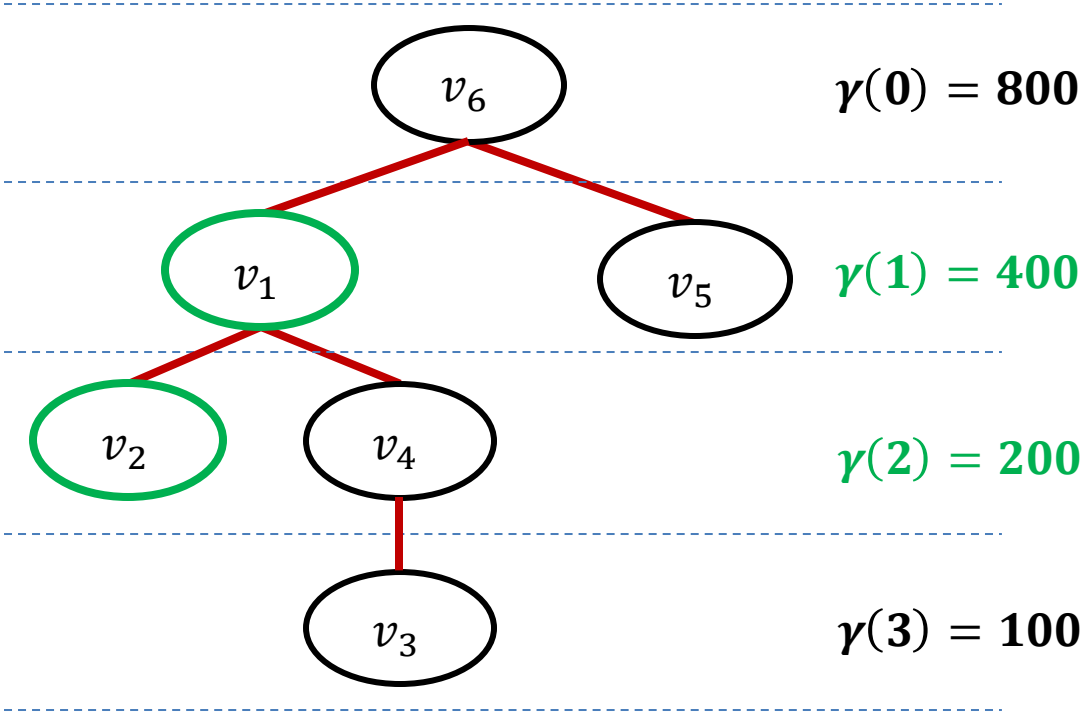


The ideal-target protocol: example

$E_{total} = 2100$

d	n_d
0	1
1	2
2	2
$h = 3$	1

v_1	500
v_2	100
v_3	150
v_4	400
v_5	350
v_6	600

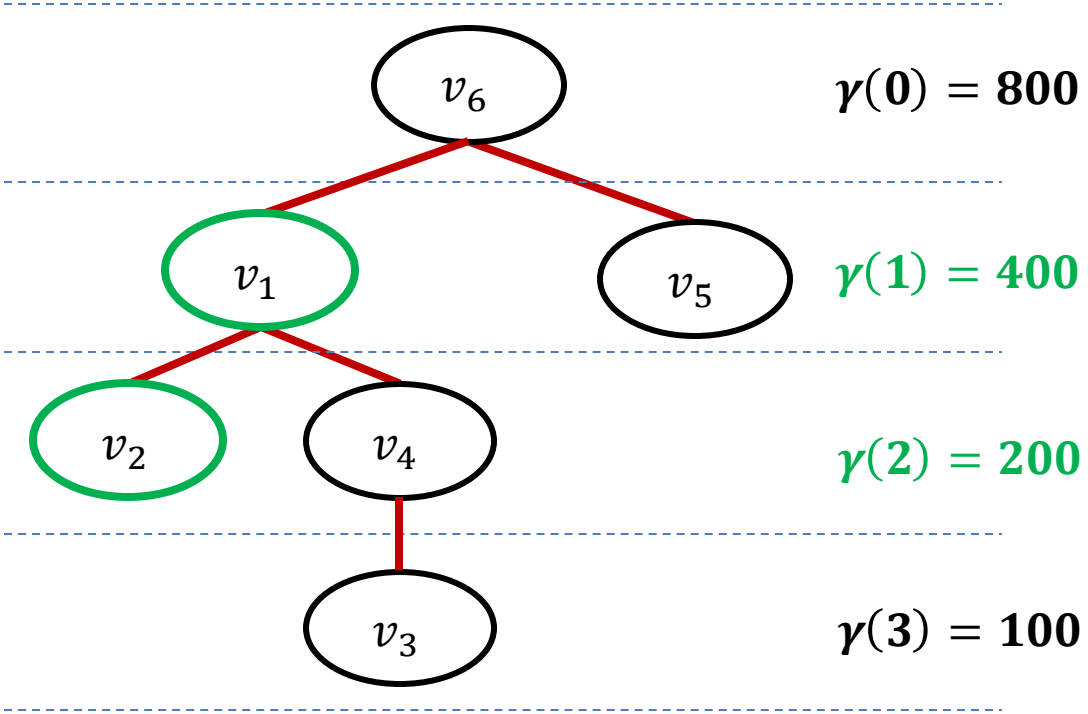


The ideal-target protocol: example

$E_{total} = 2100$

d	n_d
0	1
1	2
2	2
$h = 3$	1

v_1	**400**
v_2	**200**
v_3	150
v_4	400
v_5	350
v_6	600

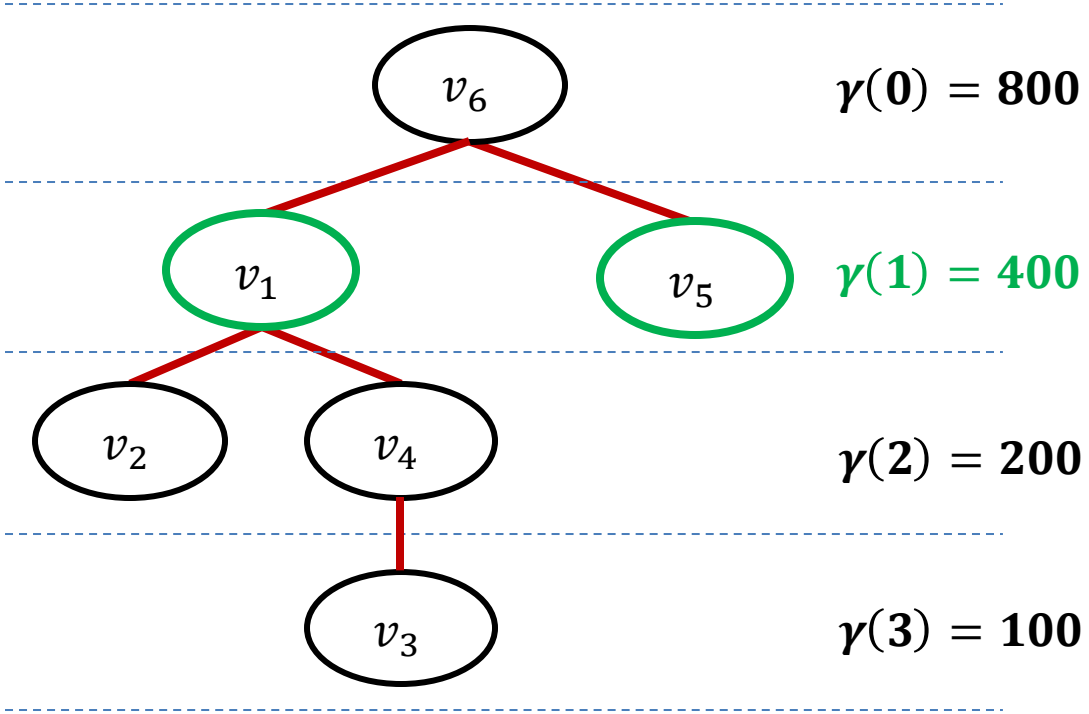


The ideal-target protocol: example

$E_{total} = 2100$

d	n_d
0	1
1	2
2	2
$h = 3$	1

v_1	400
v_2	200
v_3	150
v_4	400
v_5	350
v_6	600

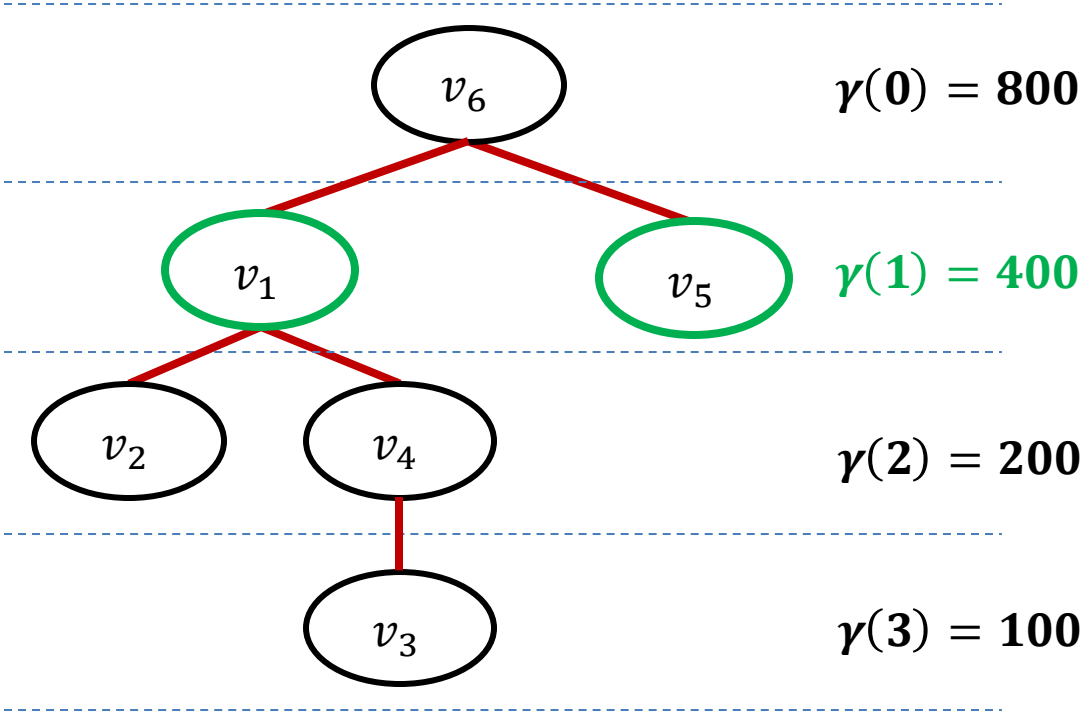


The ideal-target protocol: example

$E_{total} = 2100$

d	n_d
0	1
1	2
2	2
$h = 3$	1

v_1	400
v_2	200
v_3	150
v_4	400
v_5	350
v_6	600

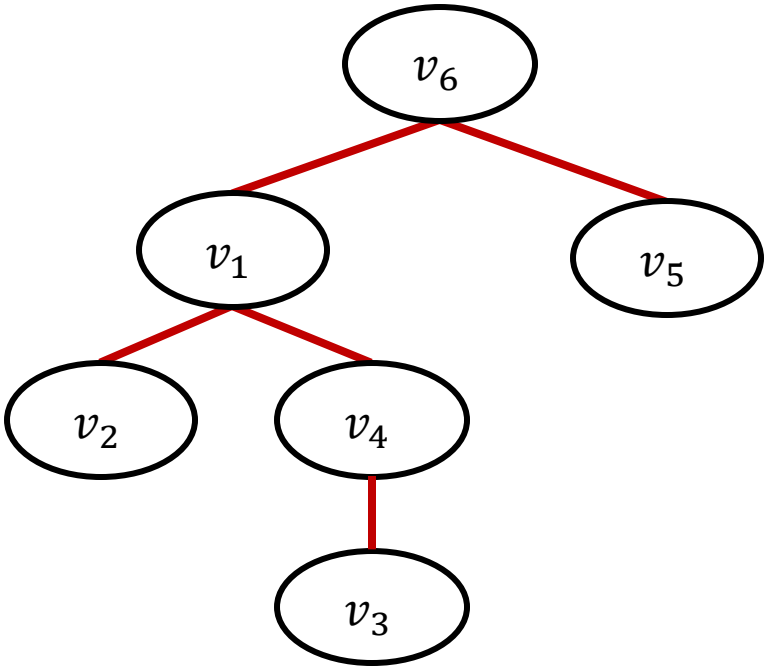


The λ-exchange protocol

- No global knowledge (NK)
- Energy can be exchanged only during interactions between parent-child pairs of nodes (p, c)
- If the energy of p is *strictly less* than λ times the energy of c, then their total energy is redistributed amongst them so that p has **exactly λ times** the energy of c

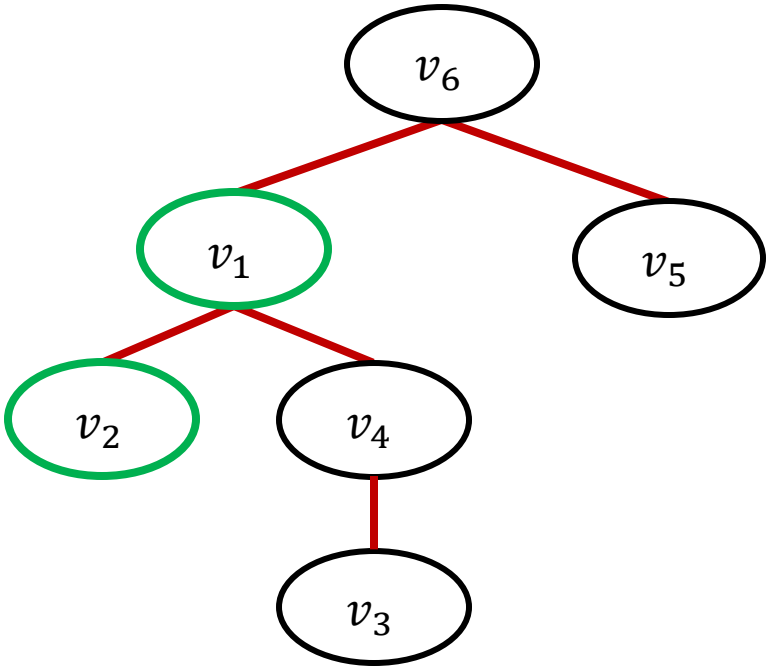
The 2-exchange protocol: example

v_1	500
v_2	100
v_3	150
v_4	400
v_5	350
v_6	600



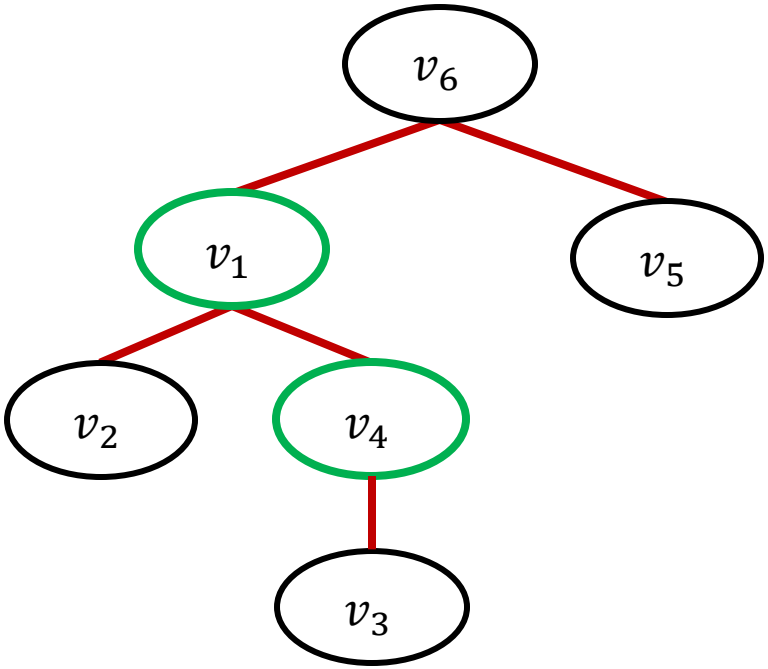
The 2-exchange protocol: example

v_1	**500**
v_2	**100**
v_3	150
v_4	400
v_5	350
v_6	600



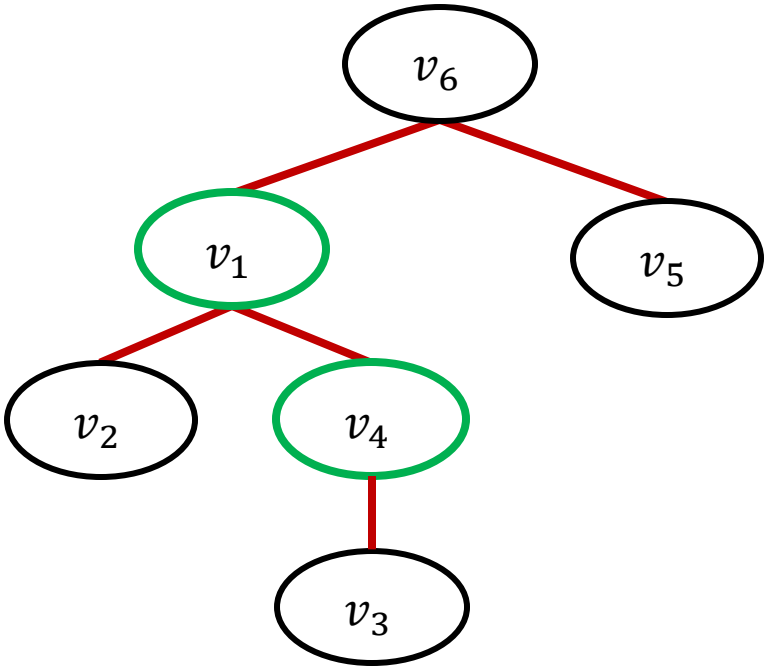
The 2-exchange protocol: example

v_1	**500**
v_2	100
v_3	150
v_4	**400**
v_5	350
v_6	600



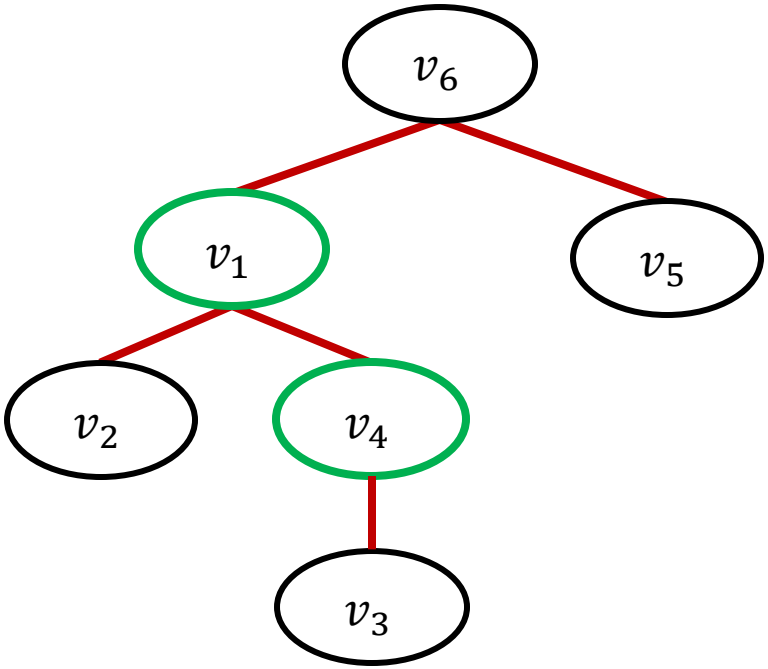
The 2-exchange protocol: example

v_1	**600**
v_2	100
v_3	150
v_4	**300**
v_5	350
v_6	600



The 2-exchange protocol: example

v_1	**600**
v_2	100
v_3	150
v_4	**300**
v_5	350
v_6	600



...

The depth-target protocol

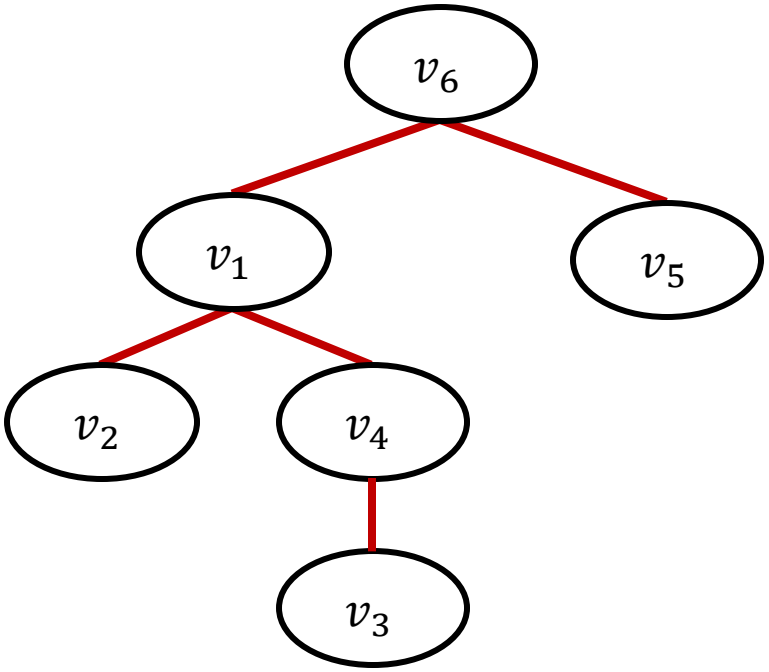
- Uses the total initial energy in the network (PK)
- Every non-root node v sets a target energy

$$\zeta_v = \frac{E_{total}}{2^{d_v}(h+1)}$$

- Node v requests energy if its current energy is less than ζ_v
- Node v can give energy if its current energy is more than ζ_v
- The root **can get or give energy**, depending on the needs of the other interacting node

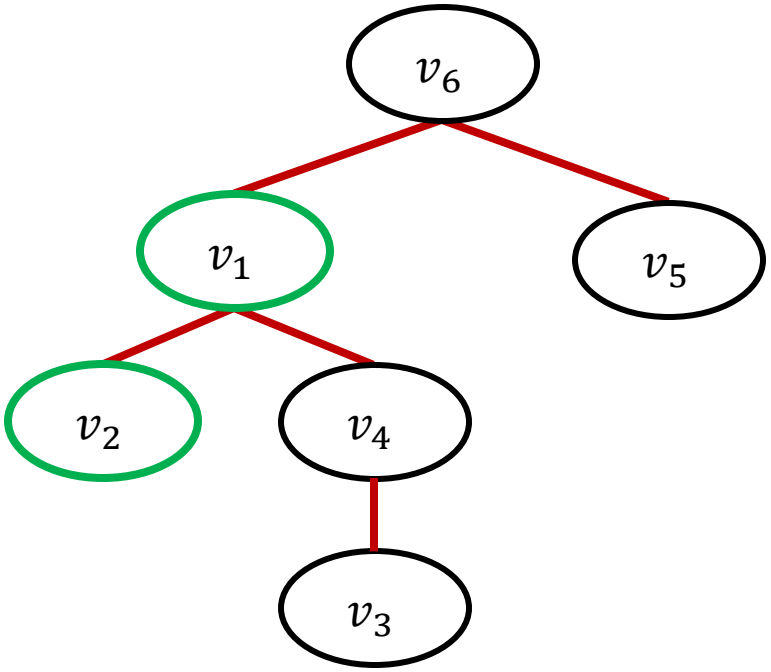
The depth-target protocol: example

	$E(t)$	ζ
v_1	500	262.5
v_2	100	131.25
v_3	150	65.625
v_4	400	131.25
v_5	350	262.5
v_6	600	–



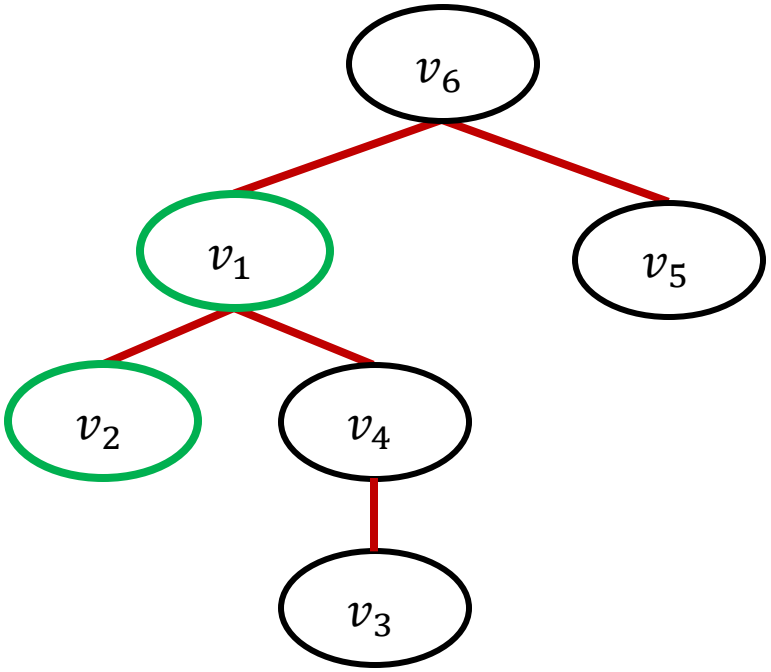
The depth-target protocol: example

	$E(t)$	ζ
v_1	**500**	**262.5**
v_2	**100**	**131.25**
v_3	150	65.625
v_4	400	131.25
v_5	350	262.5
v_6	600	–



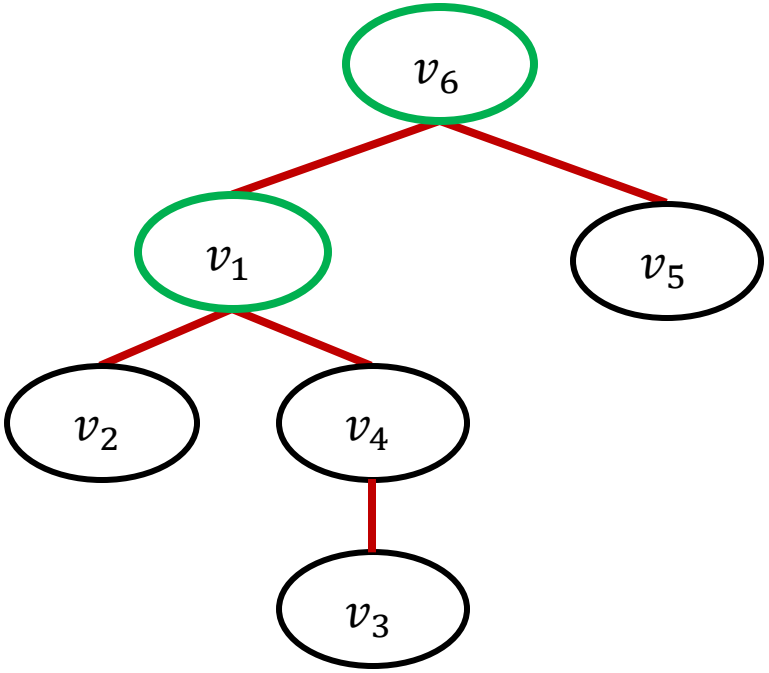
The depth-target protocol: example

	$E(t)$	ζ
v_1	**468.75**	**262.5**
v_2	**131.25**	**131.25**
v_3	150	65.625
v_4	400	131.25
v_5	350	262.5
v_6	600	–



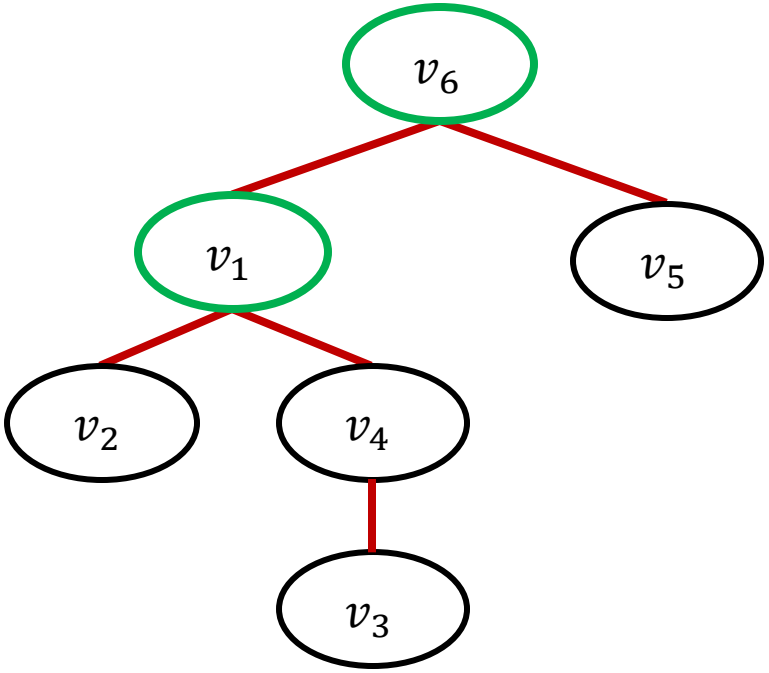
The depth-target protocol: example

	$E(t)$	ζ
v_1	**468.75**	**262.5**
v_2	131.25	131.25
v_3	150	65.625
v_4	400	131.25
v_5	350	262.5
v_6	**600**	–



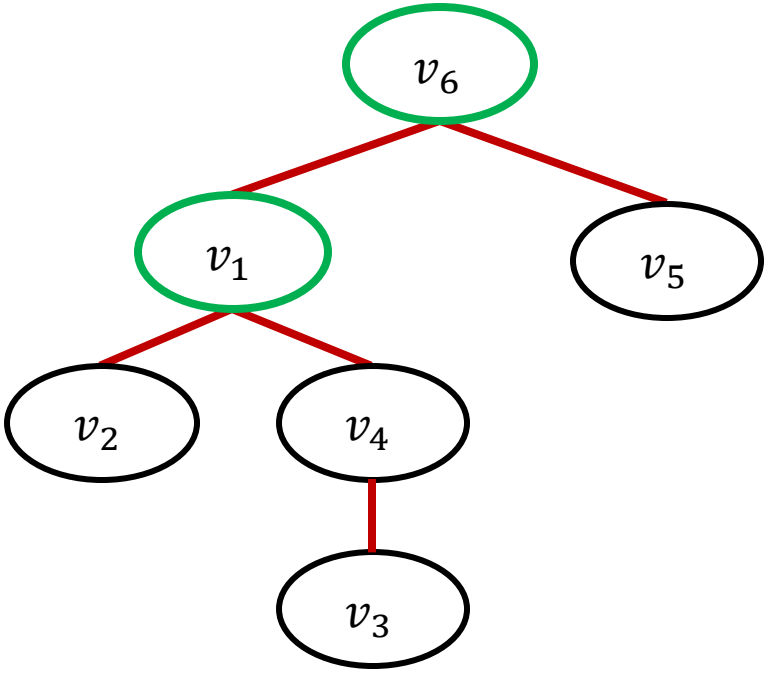
The depth-target protocol: example

	$E(t)$	ζ
v_1	**262.5**	**262.5**
v_2	131.25	131.25
v_3	150	65.625
v_4	400	131.25
v_5	350	262.5
v_6	**806.25**	–



The depth-target protocol: example

	$E(t)$	ζ
v_1	**262.5**	**262.5**
v_2	131.25	131.25
v_3	150	65.625
v_4	400	131.25
v_5	350	262.5
v_6	**806.25**	–



...

Simulation setup

- Populations of 10, 30, and 50 nodes
- Total initial energy analogous to the number of nodes and is split among them randomly
- The energy that is lost during every exchange is a random variable following $N(0.2,0.05)$
- Each simulation was repeated 100 times (statistical smoothness)

Distribution and energy distance

- DD measures the total energy that has to be redistributed at time t in order to achieve a relaxed distribution

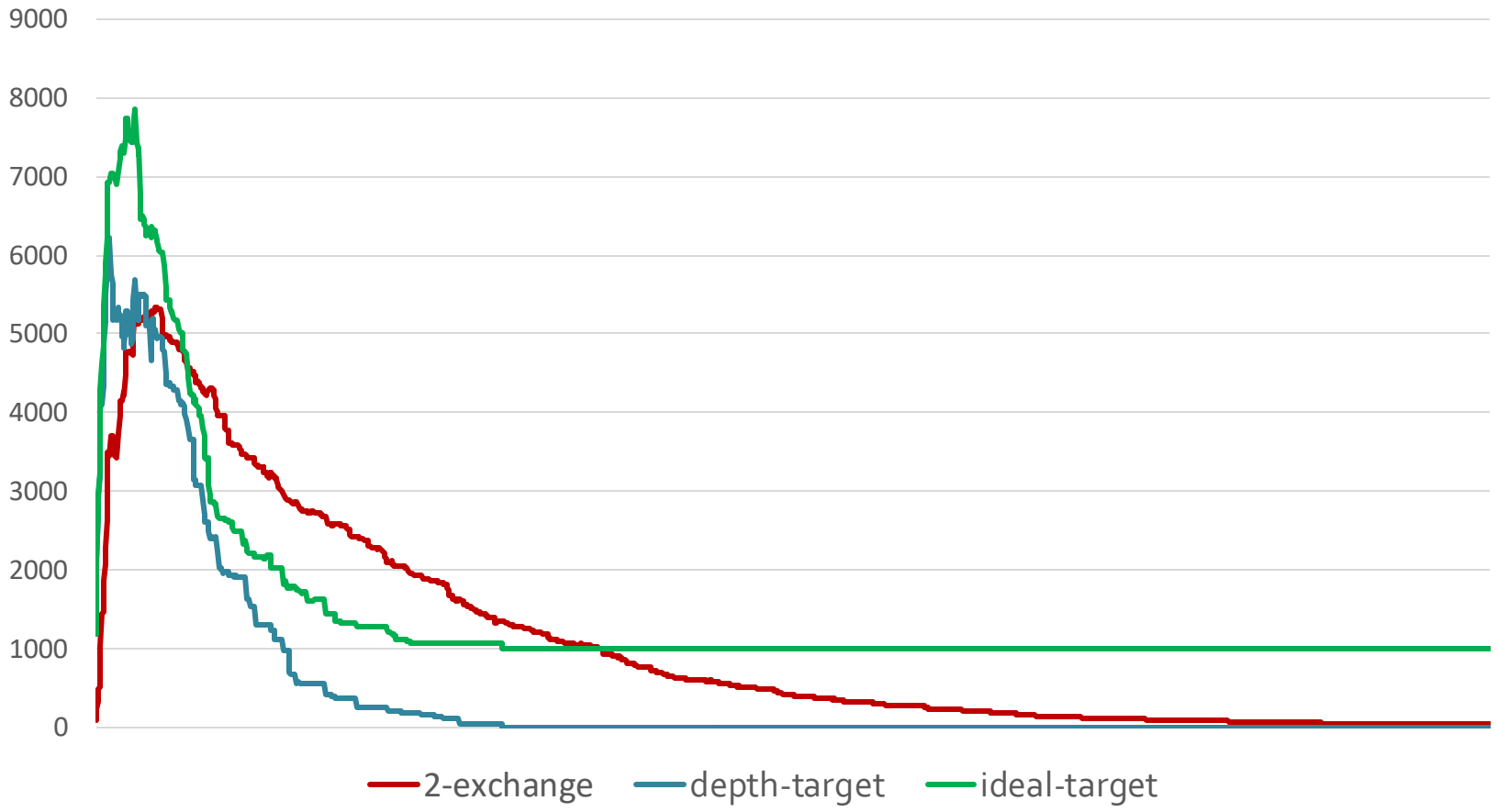
$$DD(t) = \sum_{(p,c)} \left(2E_c(t) - E_p(t)\right) \mathbf{1}\{E_p(t) < 2E_c(t)\}$$

- The convergence time of λ-exchange is the smallest τ such that $DD(\tau) = 0$
- The convergence time of ideal-target and depth-target is the smallest τ such that $DD(\tau) = DD(t)$ for any $t \geq \tau$

- ED measures the total energy that has been misplaced in the final distribution

$$ED = \frac{1}{2} \sum_{v} |E_v(\tau) - \gamma_v|$$

Simulation results



Distribution distance for $n = 10$

Simulation results

	$n = 10$	$n = 30$	$n = 50$
ideal-target	5.77 %	8.99 %	**10.77 %**
2-exchange	**5.36 %**	**7.14 %**	12.09 %
depth-target	32.85 %	42.63 %	49.54 %

Energy distance

Possible extensions

- Energy distributions with additive more energy for nodes of lower depth (instead of multiplicative more energy)
- Formation of more predictable tree networks (e.g. balanced trees)
- Nodes cannot store arbitrary levels of energy: what is going on if we assume that every node has a battery limit?
- Other network structures and corresponding energy distributions

Thank you!