

# Oblivious Interference Scheduling

**Alexander Fanghänel**

**Thomas Keßelheim**

**Harald Räcke**

**Berthold Vöcking**

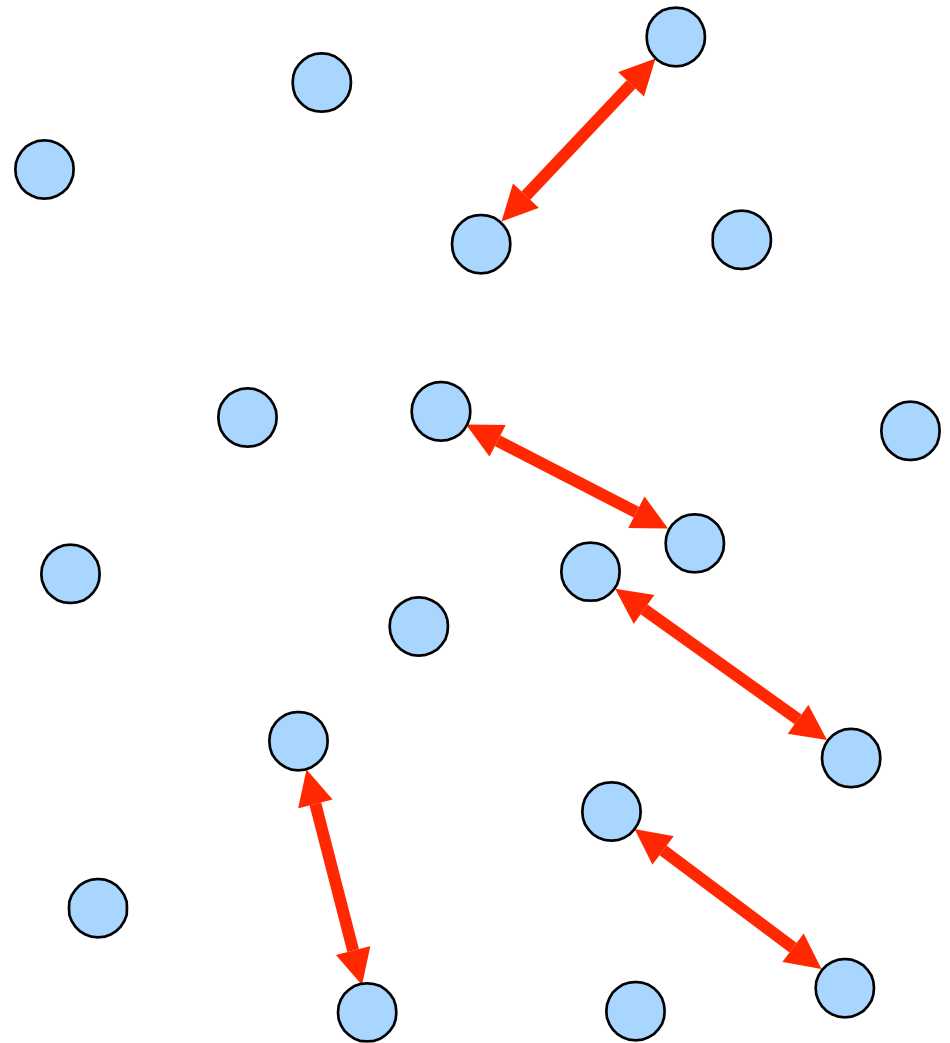
# Problem Definition

a) given nodes in a wireless network

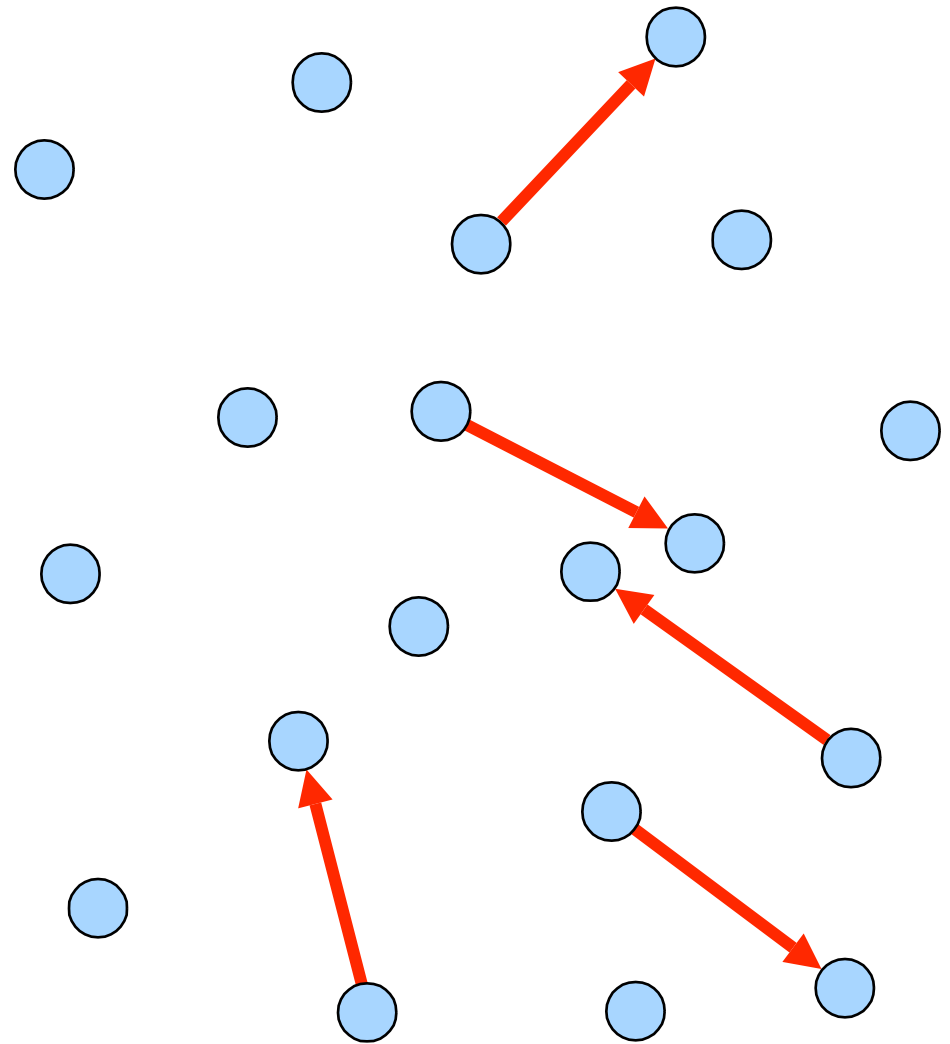
b) pairs of nodes indicate communication requests

c) fulfil all requests as quickly as possible

Two versions: Requests are either **unidirectional** or **bidirectional**



# Problem Definition

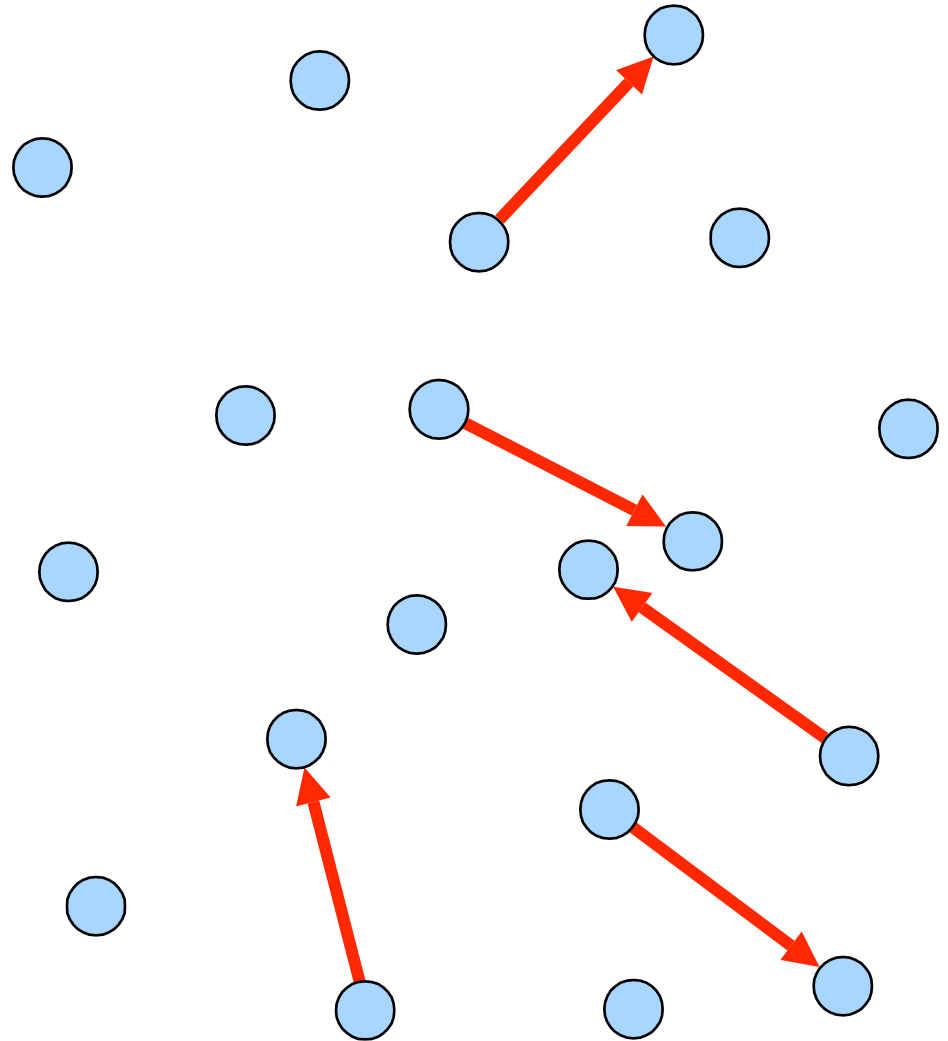


# Problem Definition

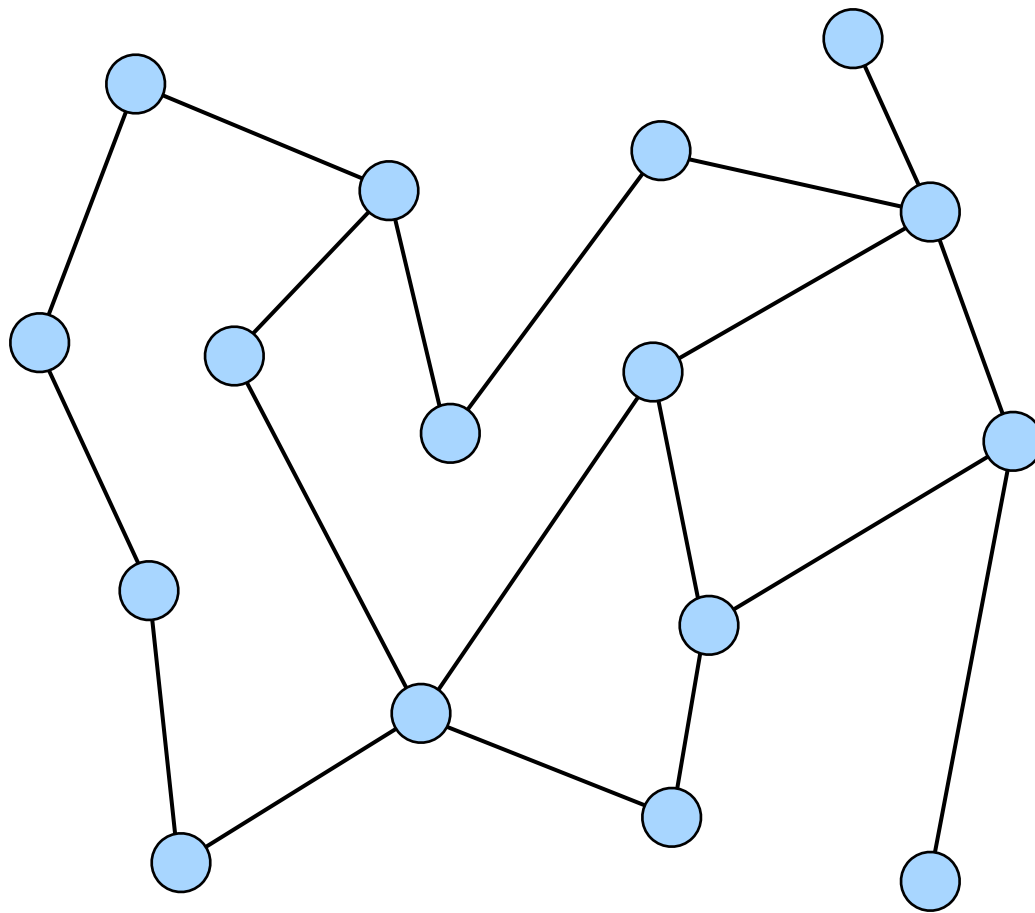
Wireless communication causes **interference**.

a) a node may not be able to receive its signal because too many other nodes are sending at the same time

**Goal:** Schedule requests in rounds such that requests within the same round do not interfere.



# Modelling Interference: The graph model



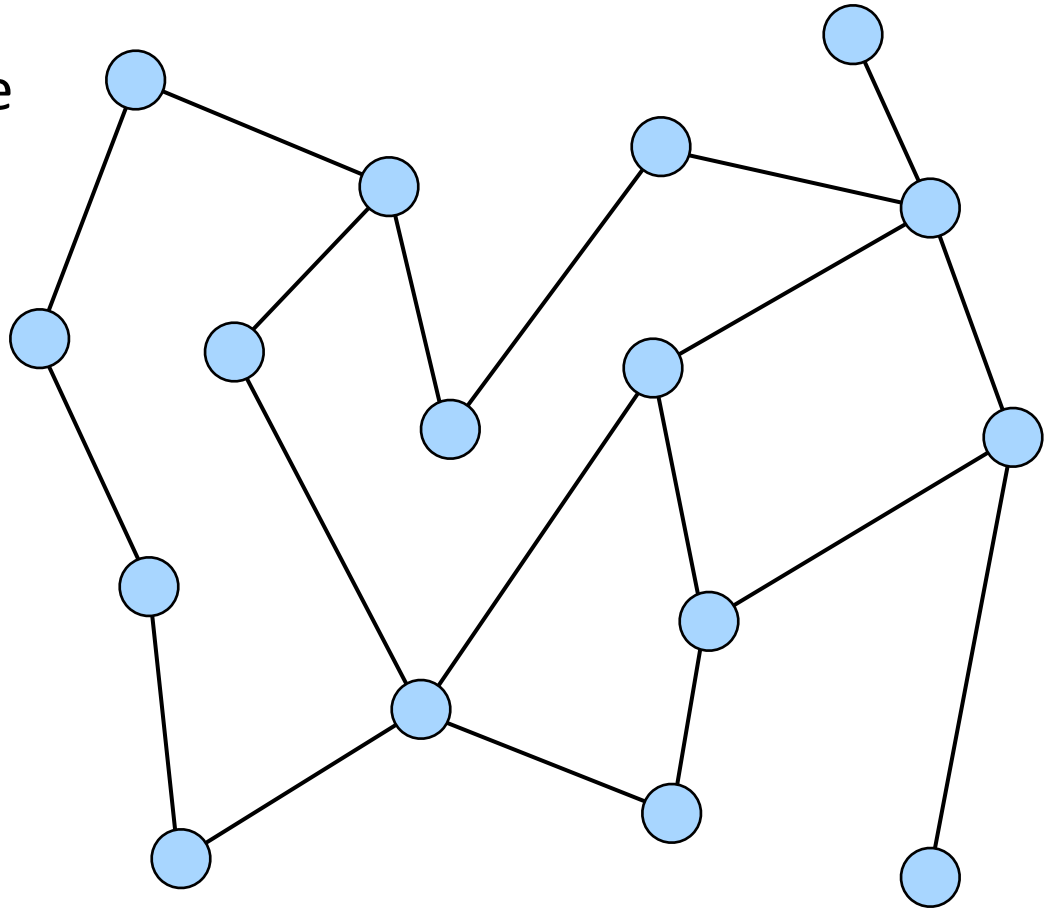
# Modelling Interference: The graph model

Model neighbourhoods by a graph.

a) a node can only receive if at most one of its neighbours is active.

## **Problem:**

Neighbourhoods cannot realistically be modelled by  $\{0,1\}$ -variable.



# Modelling Interference: SINR-constraints

# Modelling Interference: SINR-constraints

a) Metric space  $(V, d)$

b) the strength of a signal at node  $v$  send by node  $u$   
with power level  $p_u$  is

$$\frac{p_u}{d(u, v)^\alpha}$$

In order to decode a signal at node  $v_i$

# Modelling Interference: SINR-constraints

a) Metric space  $(V, d)$

b) the strength of a signal at node  $v$  send by node  $u$   
with power level  $p_u$  is

$$\frac{p_u}{d(u, v)^\alpha} \leftarrow \text{path loss exponent}$$

usually:  $2 \leq \alpha \leq 5$

In order to decode a signal at node  $v_i$

# Modelling Interference: SINR-constraints

a) Metric space  $(V, d)$

b) the strength of a signal at node  $v$  send by node  $u$   
with power level  $p_u$  is

$$\text{decay} \longrightarrow \frac{p_u}{d(u, v)^\alpha} \longleftarrow \text{path loss exponent}$$

usually:  $2 \leq \alpha \leq 5$

In order to decode a signal at node  $v_i$

$$\frac{p_i}{d(u_i, v_i)^\alpha} \geq \beta \left( \sum_{j \neq i} \frac{p_j}{d(u_j, v_i)^\alpha} + \nu \right)$$

# Modelling Interference: SINR-constraints

a) Metric space  $(V, d)$

b) the strength of a signal at node  $v$  send by node  $u$   
with power level  $p_u$  is

$$\text{decay} \longrightarrow \frac{p_u}{d(u, v)^\alpha} \longleftarrow \text{path loss exponent}$$

usually:  $2 \leq \alpha \leq 5$

In order to decode a signal at node  $v_i$

$$\underbrace{\frac{p_i}{d(u_i, v_i)^\alpha}}_{\text{signal at } v_i} \geq \beta \left( \sum_{j \neq i} \frac{p_j}{d(u_j, v_i)^\alpha} + \nu \right)$$

# Modelling Interference: SINR-constraints

a) Metric space  $(V, d)$

b) the strength of a signal at node  $v$  send by node  $u$   
with power level  $p_u$  is

$$\text{decay} \longrightarrow \frac{p_u}{d(u, v)^\alpha} \longleftarrow \text{path loss exponent}$$

usually:  $2 \leq \alpha \leq 5$

In order to decode a signal at node  $v_i$

$$\underbrace{\frac{p_i}{d(u_i, v_i)^\alpha}}_{\text{signal at } v_i} \geq \beta \left( \underbrace{\sum_{j \neq i} \frac{p_j}{d(u_j, v_i)^\alpha}}_{\text{interference at } v_i} + \nu \right)$$

# Modelling Interference: SINR-constraints

a) Metric space  $(V, d)$

b) the strength of a signal at node  $v$  send by node  $u$   
with power level  $p_u$  is

$$\text{decay} \longrightarrow \frac{p_u}{d(u, v)^\alpha} \longleftarrow \text{path loss exponent}$$

usually:  $2 \leq \alpha \leq 5$

In order to decode a signal at node  $v_i$

$$\underbrace{\frac{p_i}{d(u_i, v_i)^\alpha}}_{\text{signal at } v_i} \geq \underbrace{\beta}_{\text{gain}} \left( \underbrace{\sum_{j \neq i} \frac{p_j}{d(u_j, v_i)^\alpha}}_{\text{interference at } v_i} + \nu \right)$$

# Modelling Interference: SINR-constraints

a) Metric space  $(V, d)$

b) the strength of a signal at node  $v$  send by node  $u$   
with power level  $p_u$  is

$$\text{decay} \longrightarrow \frac{p_u}{d(u, v)^\alpha} \longleftarrow \text{path loss exponent}$$

usually:  $2 \leq \alpha \leq 5$

In order to decode a signal at node  $v_i$

$$\underbrace{\frac{p_i}{d(u_i, v_i)^\alpha}}_{\text{signal at } v_i} \geq \beta \left( \underbrace{\sum_{j \neq i} \frac{p_j}{d(u_j, v_i)^\alpha}}_{\text{interference at } v_i} + \nu \right)$$

gain noise

# Modelling Interference: SINR-constraints

$$w \in \{u_i, v_i\}$$

$$\frac{p_i}{d(u_i, v_i)^\alpha} \geq \beta \left( \sum_{j \neq i} \left( \frac{p_j}{d(u_j, w)^\alpha} + \frac{p_j}{d(v_j, w)^\alpha} \right) + \nu \right)$$

# Modelling Interference: SINR-constraints

a) Bidirectional constraints for all  $i$  and  $w \in \{u_i, v_i\}$

$$\frac{p_i}{d(u_i, v_i)^\alpha} \geq \beta \left( \sum_{j \neq i} \left( \frac{p_j}{d(u_j, w)^\alpha} + \frac{p_j}{d(v_j, w)^\alpha} \right) + \nu \right)$$

f) we assume that the ambient noise  $\nu$  is 0.

# Interference Scheduling Problem

# Interference Scheduling Problem

Given a set of requests (pairs of points in a metric space):

- a) assign a power level to each request
- b) colour requests such that requests in a colour class fulfil the SINR-constraints.

**Goal:** minimize the number of colours.

# Example



# Example

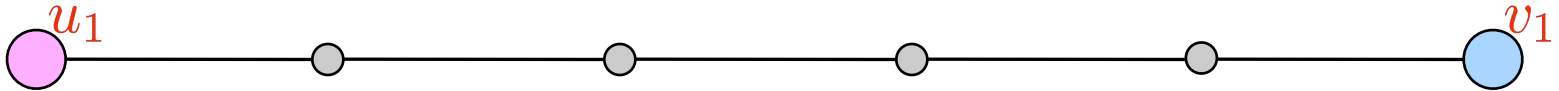
Linear network with 6 nodes.



**Decays** for the case  $\alpha=2$ .

# Example

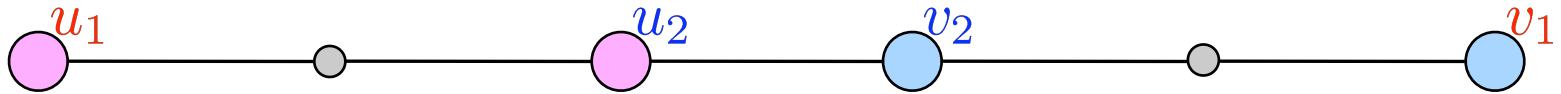
Linear network with 6 nodes.



**Decays** for the case  $\alpha=2$ .

# Example

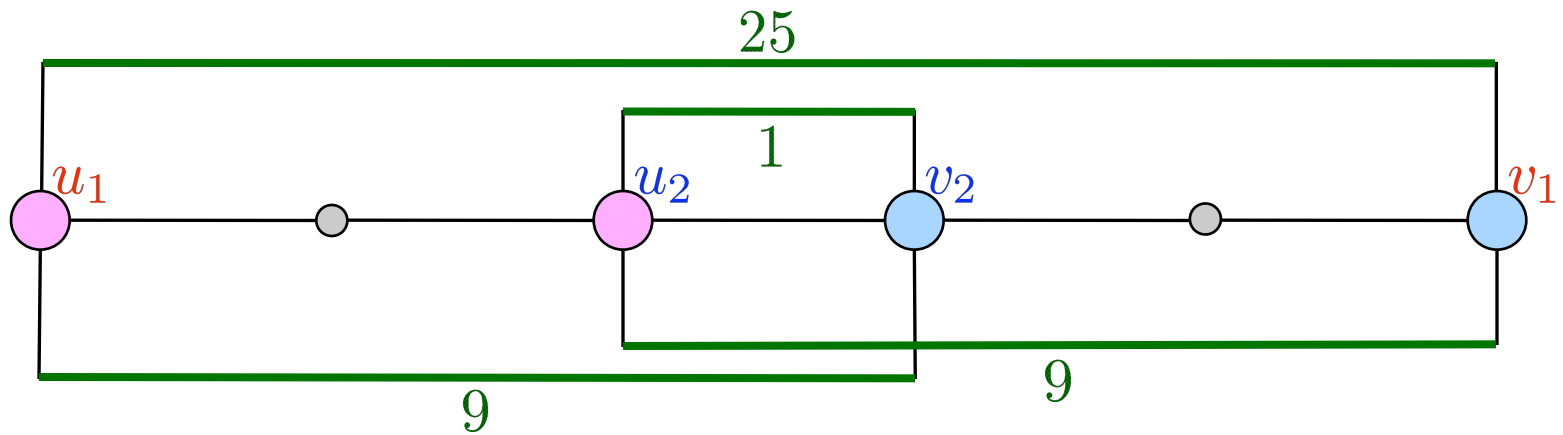
Linear network with 6 nodes.



**Decays** for the case  $\alpha=2$ .

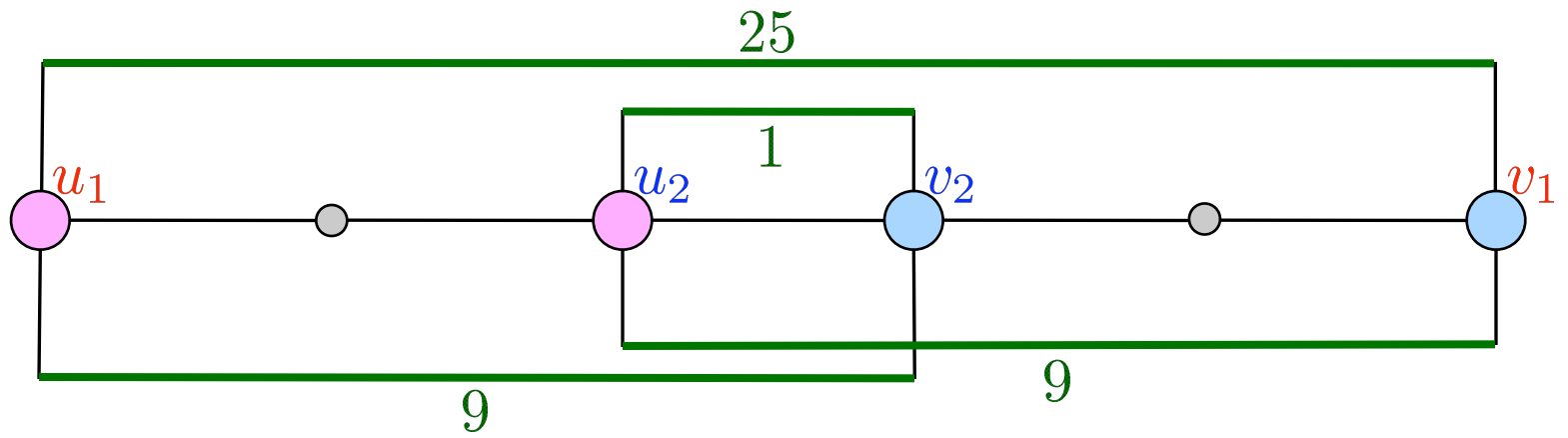
# Example

Linear network with 6 nodes.



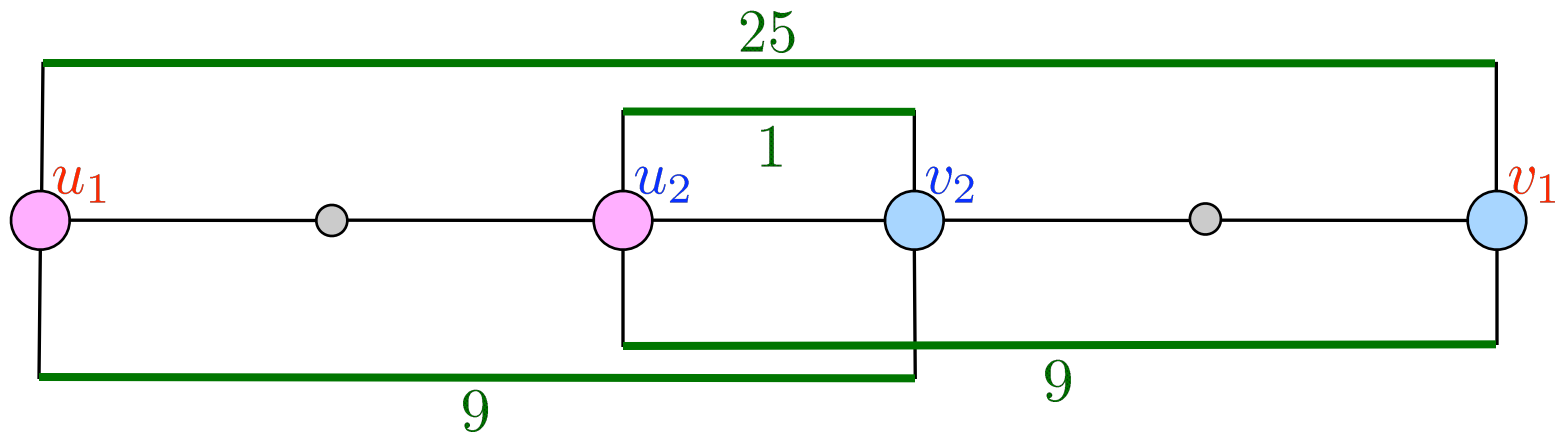
**Decays** for the case  $\alpha=2$ .

# Example



# Example

Linear network with 6 nodes. Uniform power assignment.

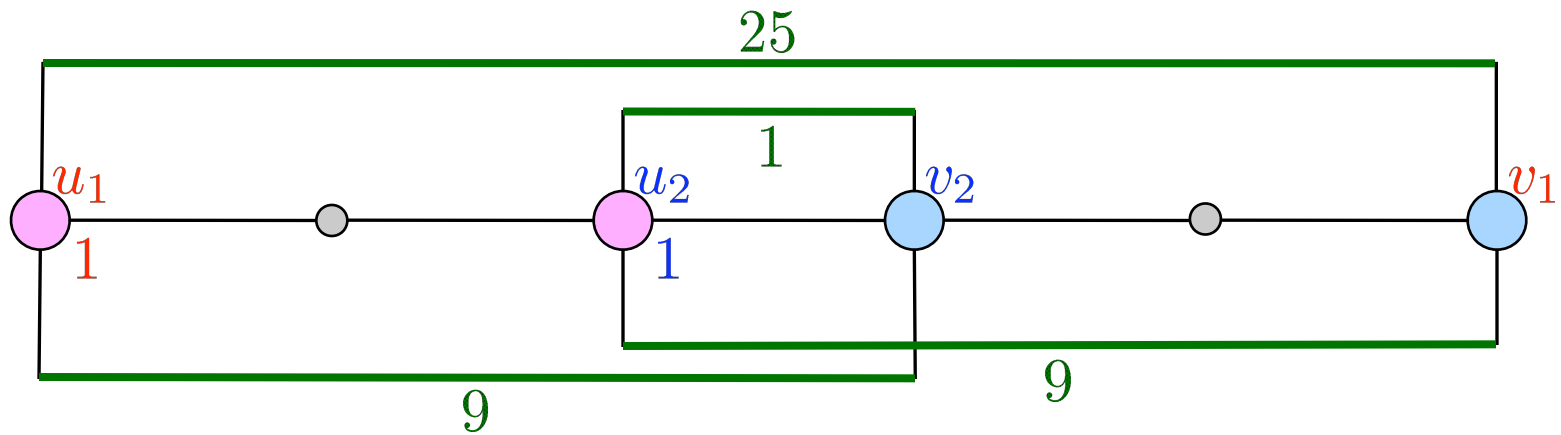


**Decays** for the case  $\alpha=2$ .

**We need two rounds to schedule both pairs.**

# Example

Linear network with 6 nodes. Uniform power assignment.

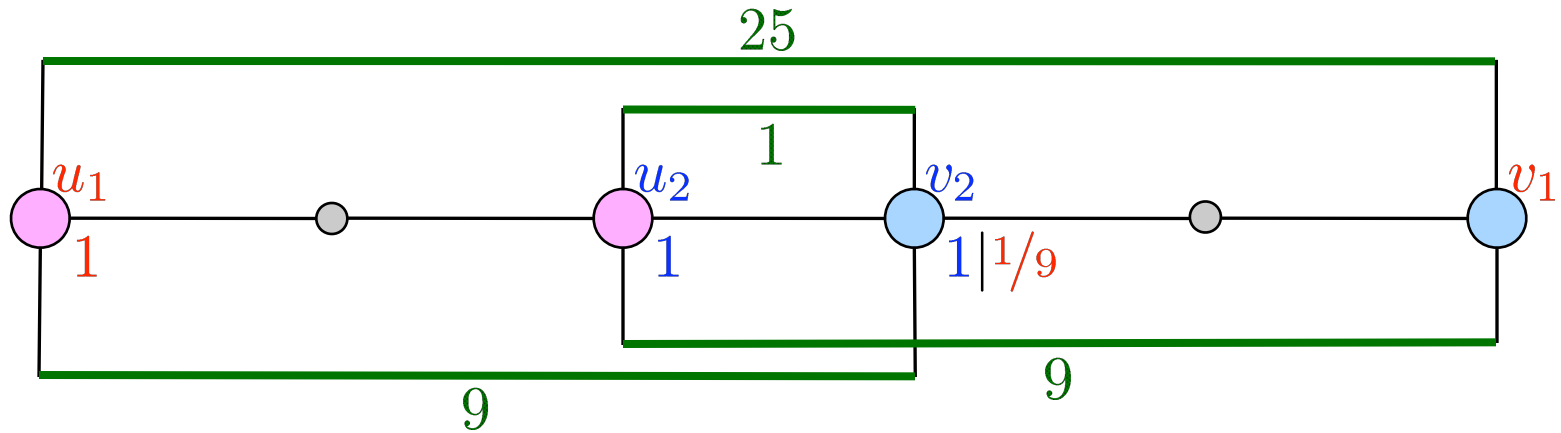


**Decays** for the case  $\alpha=2$ .

**We need two rounds to schedule both pairs.**

# Example

Linear network with 6 nodes. Uniform power assignment.

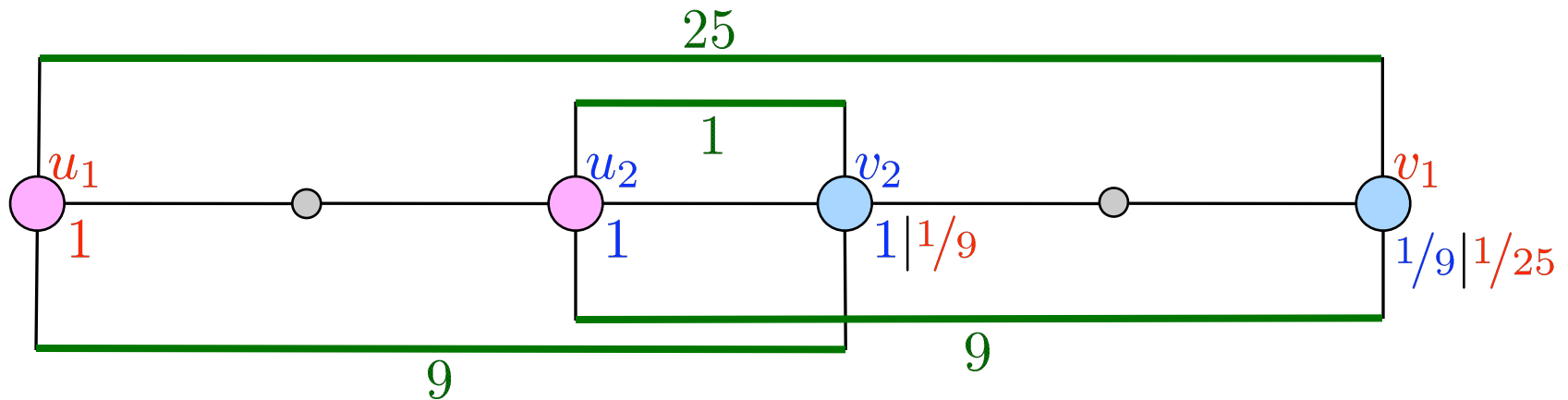


**Decays** for the case  $\alpha=2$ .

**We need two rounds to schedule both pairs.**

# Example

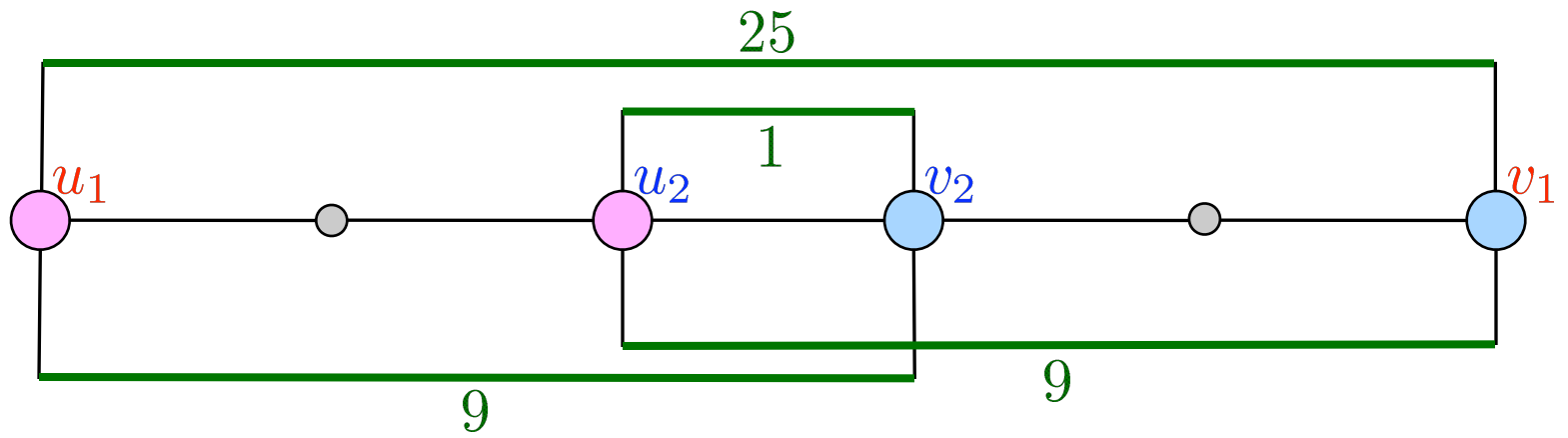
Linear network with 6 nodes. Uniform power assignment.



**Decays** for the case  $\alpha=2$ .

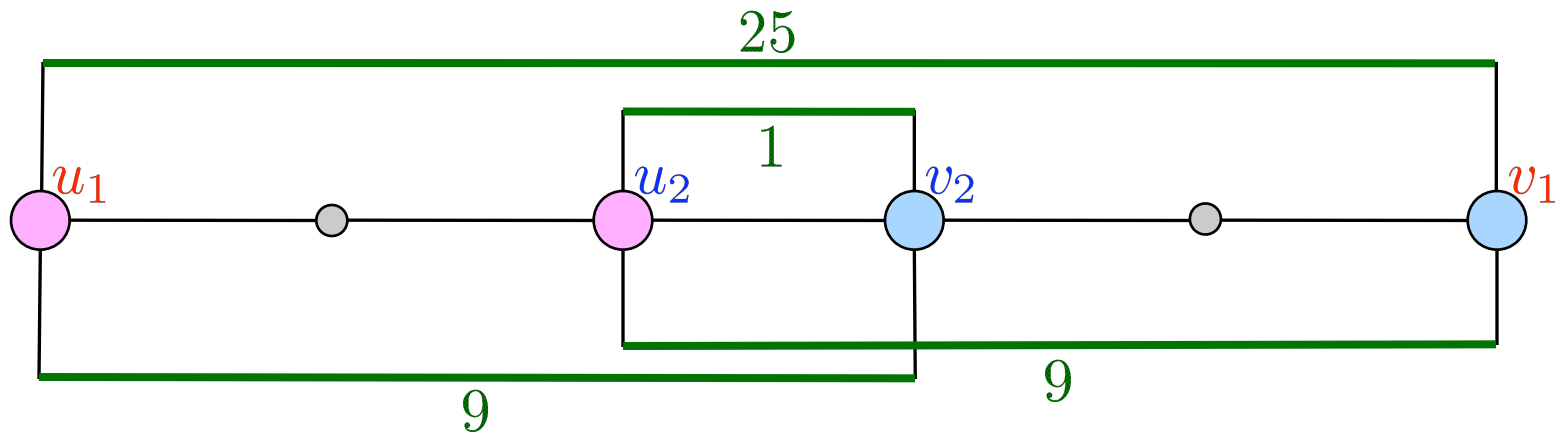
**We need two rounds to schedule both pairs.**

# Example



# Example

Linear network with 6 nodes. Linear power assignment.

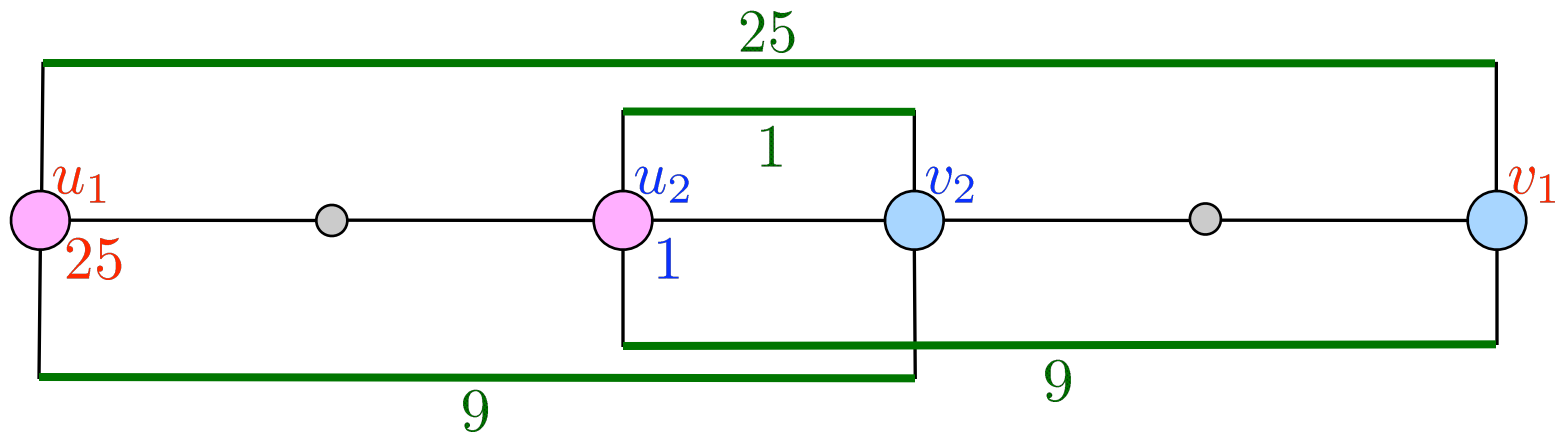


**Decays** for the case  $\alpha=2$ .

**We need two rounds to schedule both pairs.**

# Example

Linear network with 6 nodes. Linear power assignment.

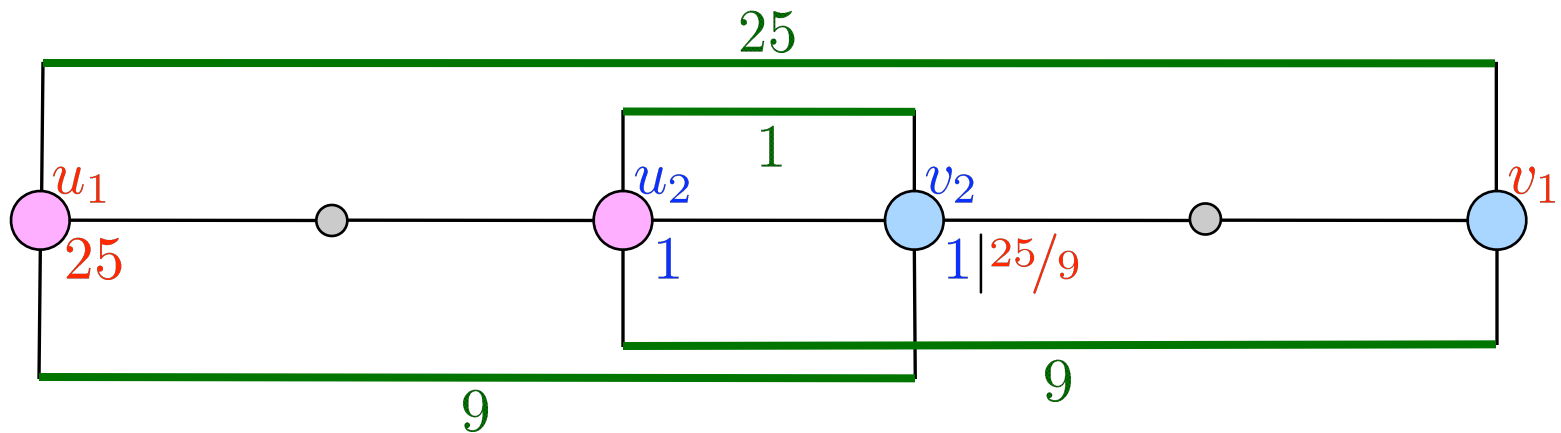


**Decays** for the case  $\alpha=2$ .

**We need two rounds to schedule both pairs.**

# Example

Linear network with 6 nodes. Linear power assignment.

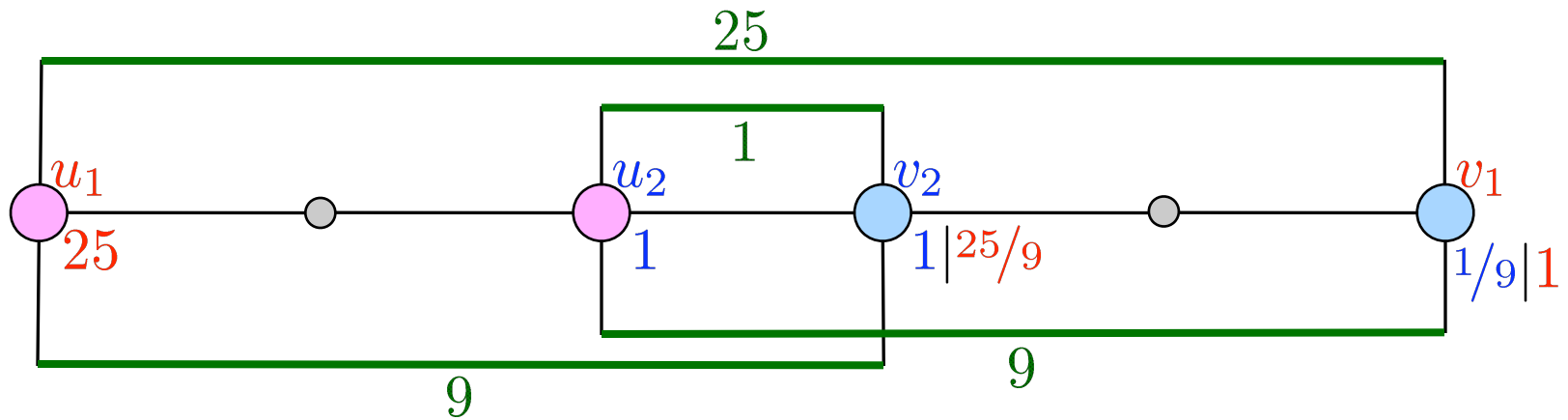


**Decays** for the case  $\alpha=2$ .

**We need two rounds to schedule both pairs.**

# Example

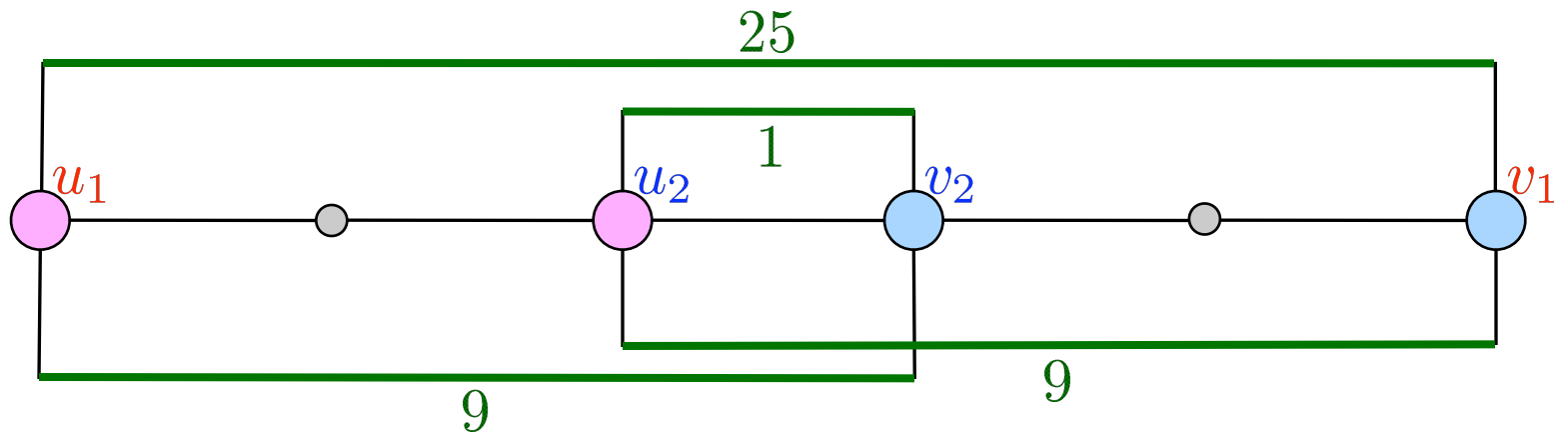
Linear network with 6 nodes. Linear power assignment.



**Decays** for the case  $\alpha=2$ .

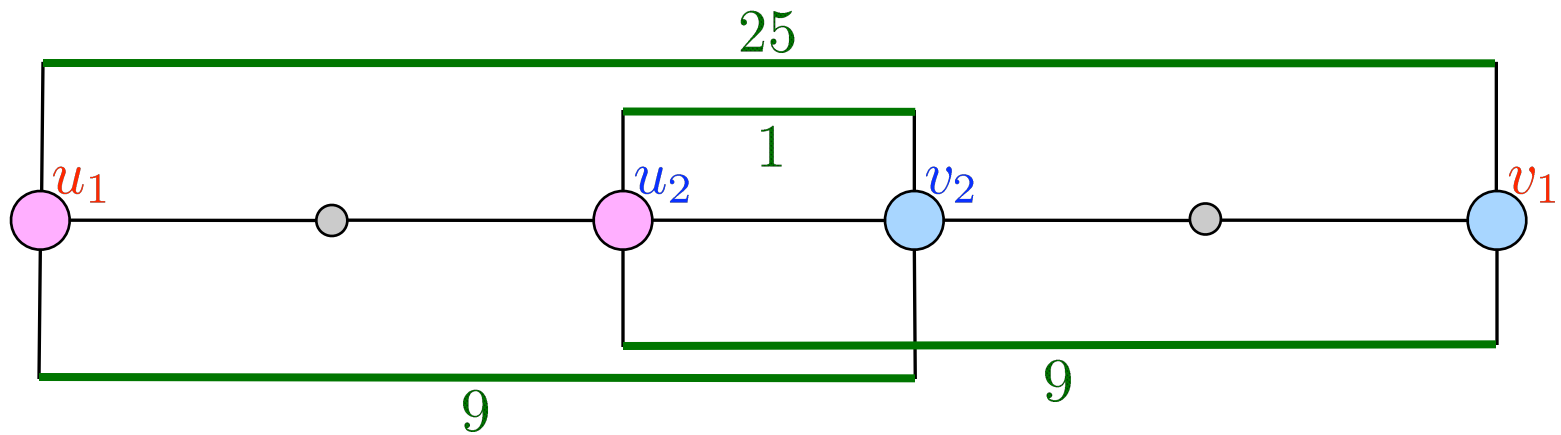
**We need two rounds to schedule both pairs.**

# Example



# Example

Linear network with 6 nodes. Square root power assignment.

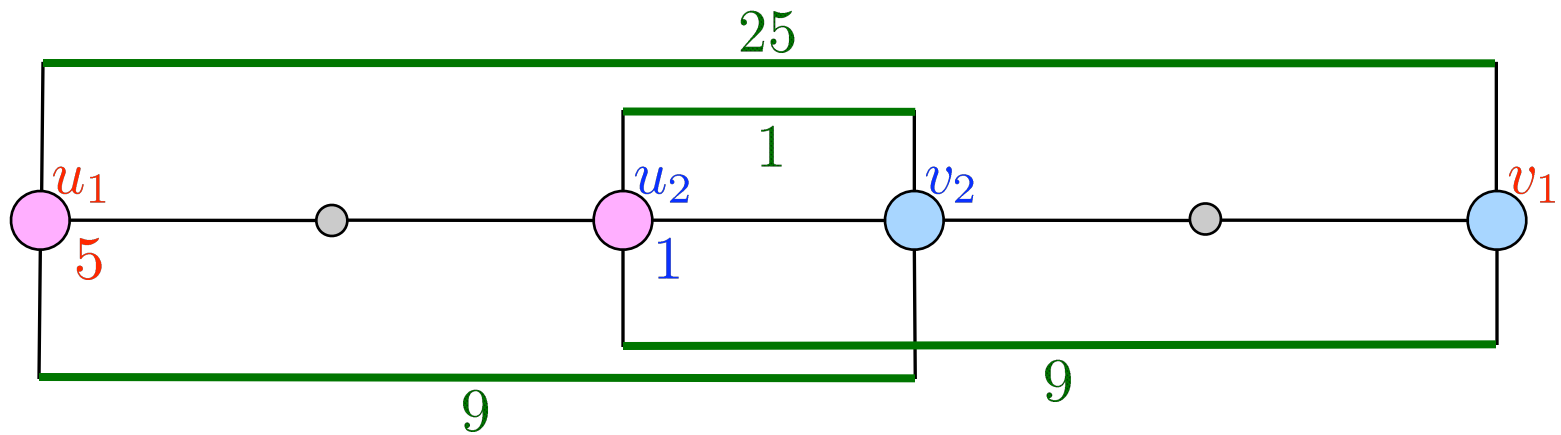


**Decays** for the case  $\alpha=2$ .

**Both pairs can be scheduled concurrently.**

# Example

Linear network with 6 nodes. Square root power assignment.

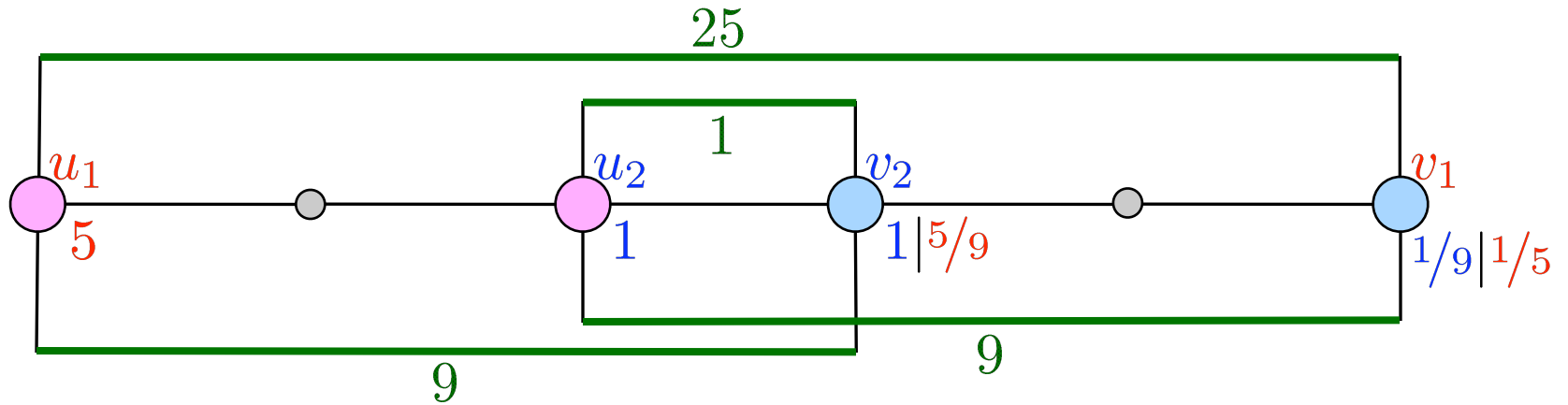


**Decays** for the case  $\alpha=2$ .

**Both pairs can be scheduled concurrently.**

# Example

Linear network with 6 nodes. Square root power assignment.



**Decays** for the case  $\alpha=2$ .

**Both pairs can be scheduled concurrently.**

# Interference Scheduling Problem

# Interference Scheduling Problem

We investigate **oblivious power assignments**. The power level for a request may only depend on the distance/decay for this request.

This means there is a function  $f : \mathbb{R}^+ \longrightarrow \mathbb{R}^+$  that specifies the power level to use for a given distance/decay.

Question:

**Is there an oblivious power assignment that is always good?**

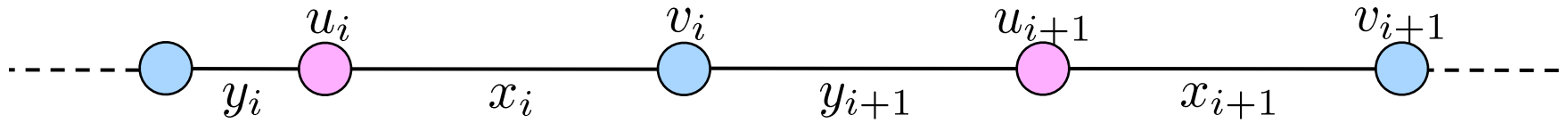
This means for any set of request pairs there exists a coloring with an almost optimal number of colors such that the oblivious power assignment fulfils the SINR-constraints.

# Interference Scheduling Problem

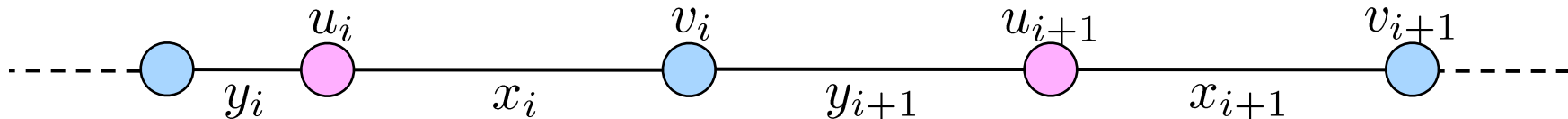
## Theorem I.

In the **unidirectional** version of the problem for any oblivious power assignment there exist an input such that this power assignment is far away from optimal.

# Unidirectional is Difficult



# Unidirectional is Difficult



Assume that  $f : \mathbb{R}^+ \rightarrow \mathbb{R}^+$  is asymptotically unbounded.

Assume  $y_j$ 's,  $x_j$ 's have been chosen for  $j < i$

b) set  $y_i = 2(y_{i-1} + x_{i-1})$

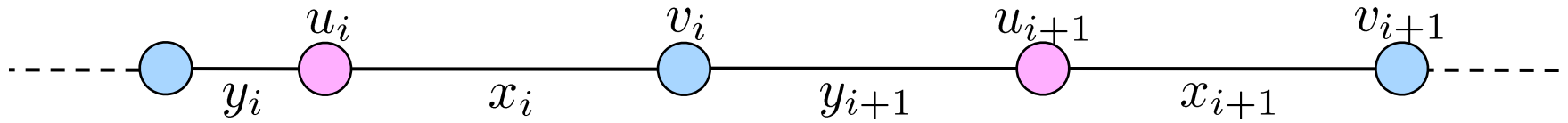
c) choose  $x_i$  such that  $y_i \leq x_i$  and

$$\forall j < i : f(x_i) \geq y_i^\alpha \frac{f(x_j)}{x_j^\alpha}$$

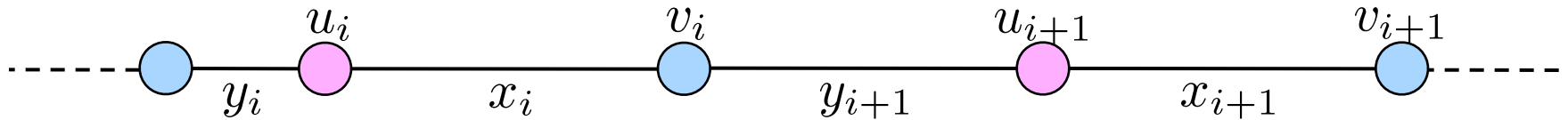
f) the  $y_i$ 's are exponentially increasing. Therefore

$$\forall j < i : d(v_j, u_i)^\alpha \approx y_i^\alpha$$

# Unidirectional is Difficult



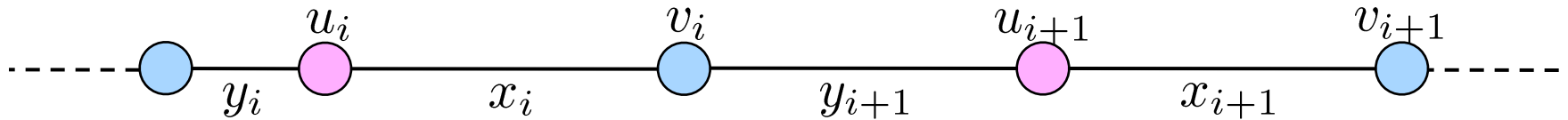
# Unidirectional is Difficult



SINR-constraint at node  $j$  for the square root power assignment

Linear interference  $\rightarrow$  only a constant number of pairs can be scheduled in a single round.

# Unidirectional is Difficult



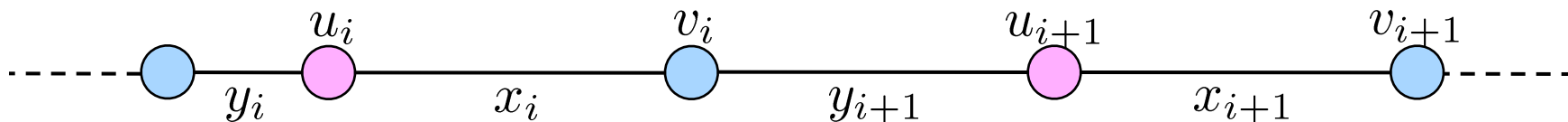
SINR-constraint at node  $j$  for the square root power assignment

$$\underbrace{\frac{f(x_j)}{x_j^\alpha}}_{\text{signal at } v_j} \geq$$

**signal at  $v_j$**

Linear interference  $\rightarrow$  only a constant number of pairs can be scheduled in a single round.

# Unidirectional is Difficult

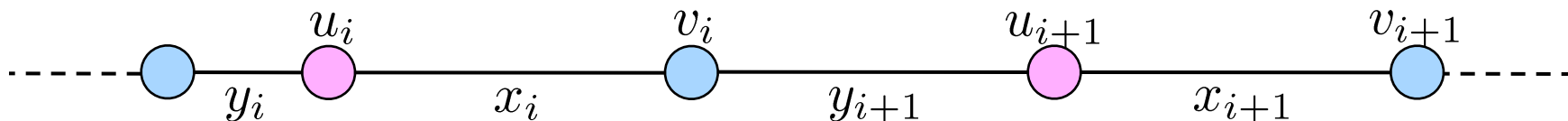


SINR-constraint at node  $j$  for the square root power assignment

$$\underbrace{\frac{f(x_j)}{x_j^\alpha}}_{\text{signal at } v_j} \geq \beta \underbrace{\sum_{i>j} \frac{p(u_i)}{d(u_i, v_k)^\alpha}}_{\text{interference at } v_j \text{ from } i>j}$$

Linear interference  $\rightarrow$  only a constant number of pairs can be scheduled in a single round.

# Unidirectional is Difficult

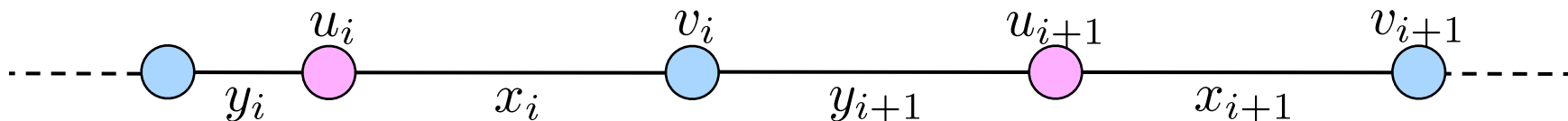


SINR-constraint at node  $j$  for the square root power assignment

$$\underbrace{\frac{f(x_j)}{x_j^\alpha}}_{\text{signal at } v_j} \geq \beta \underbrace{\sum_{i>j} \frac{p(u_i)}{d(u_i, v_k)^\alpha}}_{\text{interference at } v_j \text{ from } i>j} \approx \beta \sum_{i>j} \frac{y_i^\alpha \frac{f(x_j)}{x_j^\alpha}}{y_i^\alpha}$$

Linear interference  $\rightarrow$  only a constant number of pairs can be scheduled in a single round.

# Unidirectional is Difficult

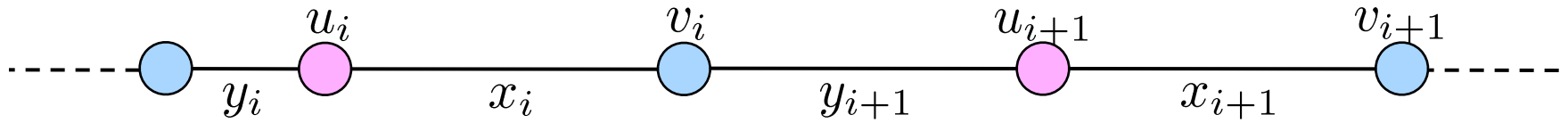


SINR-constraint at node  $j$  for the square root power assignment

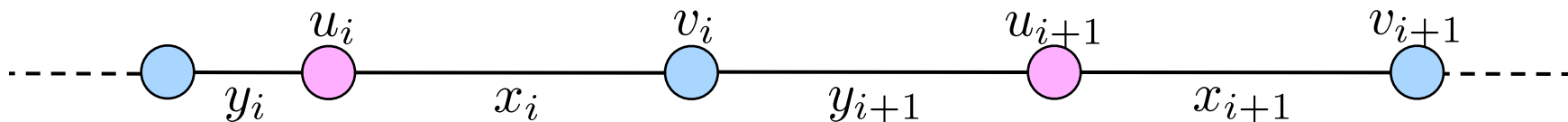
$$\underbrace{\frac{f(x_j)}{x_j^\alpha}}_{\text{signal at } v_j} \geq \beta \underbrace{\sum_{i>j} \frac{p(u_i)}{d(u_i, v_k)^\alpha}}_{\text{interference at } v_j \text{ from } i>j} \approx \beta \sum_{i>j} \frac{y_i^\alpha \frac{f(x_j)}{x_j^\alpha}}{y_i^\alpha} \geq \beta(n-j) \frac{f(x_j)}{x_j^\alpha}$$

Linear interference  $\rightarrow$  only a constant number of pairs can be scheduled in a single round.

# Unidirectional is Difficult



# Unidirectional is Difficult



a) OPT can choose  $p(u_i) = \sqrt{2^i}$

b) Fix  $j$ . Interference from  $i > j$  is at most

$$\sum_{i>j} \frac{p(u_i)}{d(u_i, v_j)^\alpha} \lesssim \sum_{i>j} \frac{\sqrt{2^i}}{y_i^\alpha} \lesssim \Omega(1)$$

f) Interferences from  $i < j$  is at most

$$\sum_{i<j} \frac{p(u_i)}{d(u_i, v_j)^\alpha} \lesssim \sum_{i<j} \frac{\sqrt{2^i}}{y_j^\alpha} \lesssim \Omega(1)$$

# Interference Scheduling Problem

## Theorem II.

In the **bidirectional** version of the problem the square root power assignment always allows a colouring such that the number of colours is  $\leq \sqrt{2} \Delta$ .

# Interference Scheduling Problem

## Theorem II.

In the **bidirectional** version of the problem the square root power assignment always allows a colouring such that the number of colours is  $O(\text{polylog}n \cdot c_{\text{opt}})$ .

## Motivation:

The Media Access Control (MAC) layer provides single-hop full-duplex communication channels in multipoint networks to higher levels of the protocol stack.

# Bidirectional Requests

## Observations (Scaling the gain):

Given a set  $U$  of requests that fulfil a gain of  $\gamma$ . Sampling requests with probability  $\frac{1}{\gamma}$  and afterwards deleting all requests for which one of the end-points does not fulfil the SINR-constraints with gain  $\gamma$ , results in a set  $U'$  of requests with

.

# Bidirectional Requests

## Observations (Scaling the gain):

Given a set  $U$  of requests that fulfil a gain of  $\beta' < \beta$ . Sampling requests with probability  $\Theta(\frac{\beta'}{\beta})$  and afterwards deleting all requests for which one of the end-points does not fulfil the SINR-constraints with gain  $\beta$ , results in a set  $U'$  of requests with

$$|U'| \geq \Omega(\frac{\beta'}{\beta})|U|.$$

# Bidirectional Requests

# Bidirectional Requests

## Problem simplification:

- a) replace a pair  $(u,v)$  by a two nodes  $u$  and  $v$ , and assign a **loss-parameter**  $\ell_u = \ell_v = d(u, v)^\alpha$  to each node.
- b) a set  $U$  of nodes is called  $\beta$ -feasible w.r.t.  $\mathbf{p}$  if

$$\forall v : I_p(v \leftarrow U) = \sum_{u \in U, u \neq v} \frac{p_u}{d(u, v)^\alpha} \leq \frac{p_v}{\beta \ell_v}$$

**interference at v from U**                      **1/β\*signal-strength at v**

- g) note that one colour class in a solution gives a  $\beta$ -feasible set on the corresponding terminals.

# Bidirectional Requests

# Bidirectional Requests

## Proof technique:

a) suppose you are given an optimal solution that requires one round.

b) terminals give a  $\beta$ -feasible set  $U$ . (for some power assignment  $p$ )

c) find a  $\beta'$ -feasible subset  $U'$  that is nearly as large. ( $p = \sqrt{\cdot}$ )

For example  $\beta' \approx \frac{\beta}{\log^2 n}$  and  $|U'| \geq \Omega(1 - \frac{\beta'}{\beta})|U|$

d) since  $U'$  is very large it must contain many terminal pairs.

e) scale the gain to obtain a subset  $U''$  of requests that can be scheduled with gain  $\beta$  in a single round using the square root power assignment.

# Bidirectional Requests

## Proof technique:

a) suppose you are given an optimal solution that requires one round.

b) terminals give a  $\beta$ -feasible set  $U$ . (for some power assignment  $p$ )

c) find a  $\beta'$ -feasible subset  $U'$  that is nearly as large. ( $p = \sqrt{\cdot}$ )

For example  $\beta' \approx \frac{\beta}{\log^2 n}$  and  $|U'| \geq \Omega(1 - \frac{\beta'}{\beta})|U|$

d) since  $U'$  is very large it must contain many terminal pairs.

e) scale the gain to obtain a subset  $U''$  of requests that can be scheduled with gain  $\beta$  in a single round using the square root power assignment.

# Interference Scheduling Problem

## Star lemma.

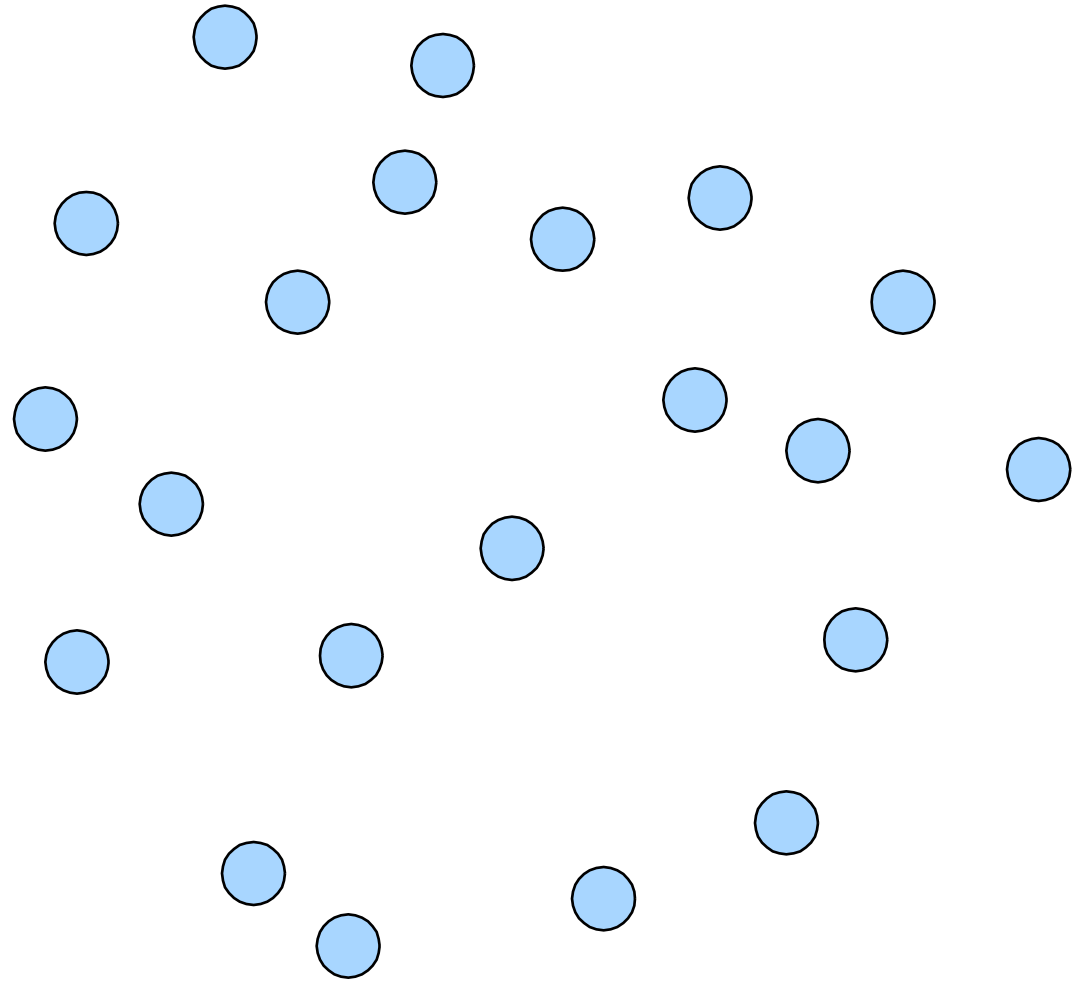
Given any set  $U$  of node-loss pairs from a **star metric** that is  $\alpha$ -feasible for some power assignment  $p$ . Then there exists a subset  $U'$  with  $|U'| \geq \frac{1}{2}|U|$  that is  $\alpha$ -feasible for the **square-root power assignment**, where

# Interference Scheduling Problem

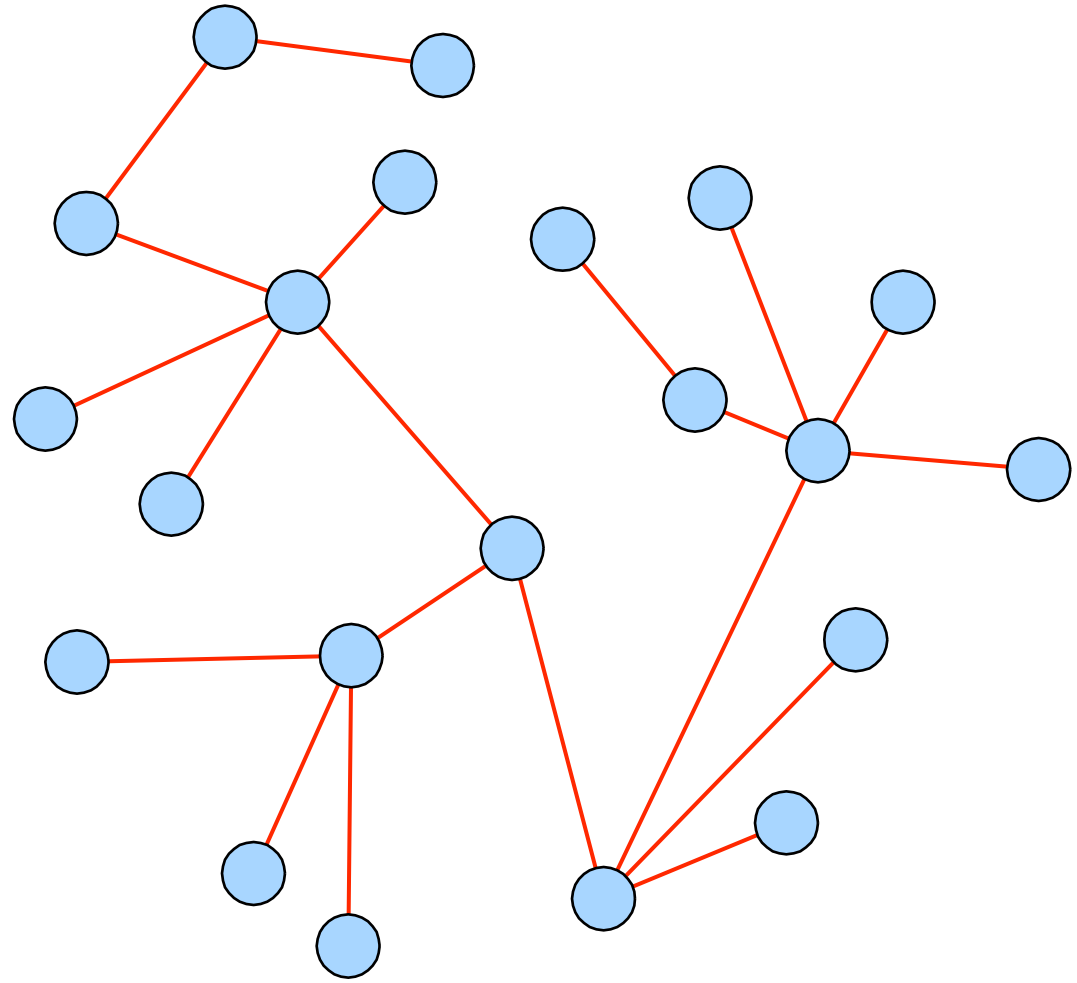
## Star lemma.

Given any set  $U$  of node-loss pairs from a **star metric** that is  $\beta$ -feasible for some power assignment  $p$ . Then there exists a subset  $U' \subset U$  with  $|U'| \geq (1 - \frac{1}{\text{polylog}n})|U|$  that is  $\beta'$ -feasible for the **square-root power assignment**, where  $\beta' = \beta^{2/3} / \text{polylog}n$

# From Stars to Trees



# From Stars to Trees



# From Stars to Trees

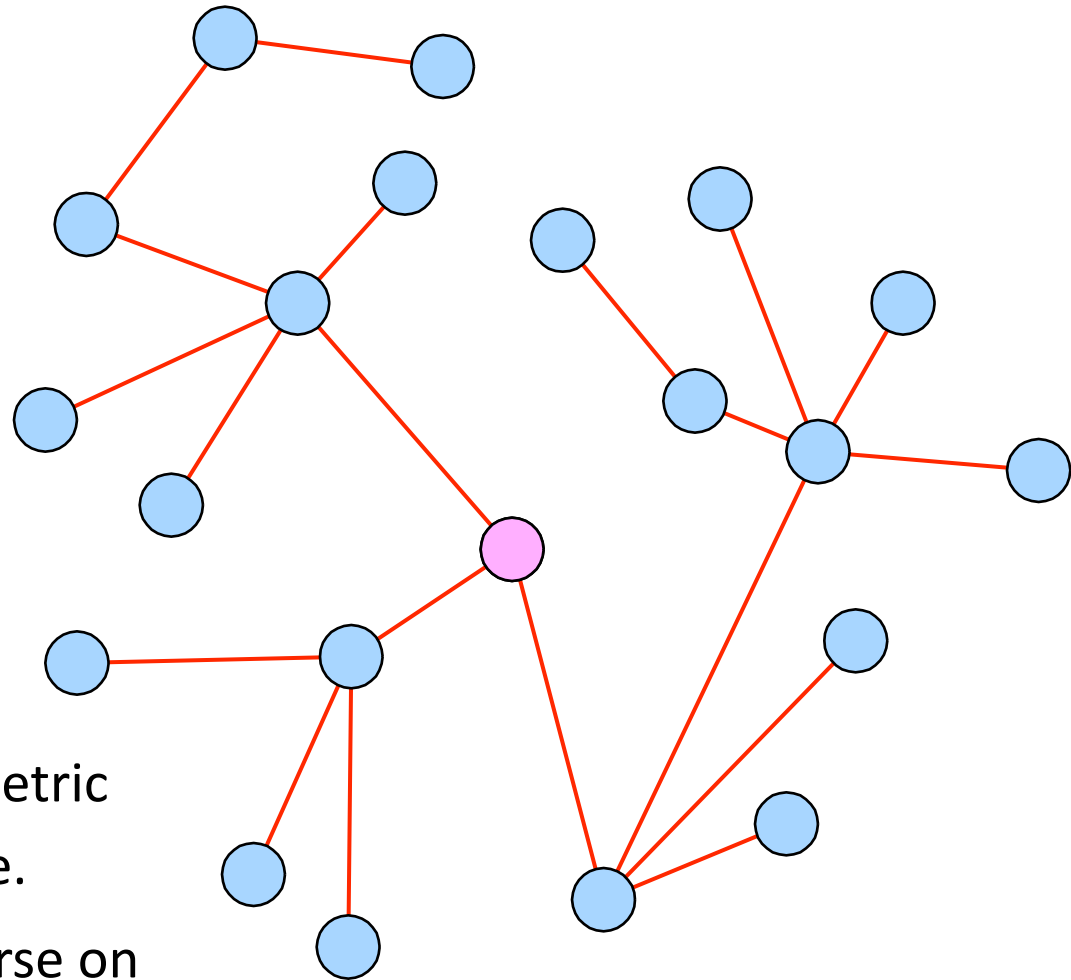
a) choose a center  $c$ , such that after the removal of  $c$  every part contains less than half of the nodes.

b) consider the star in which all nodes are connected to the center  $c$ .

c) solve the problem for this star.

d) maybe you accepted to many nodes since the star-metric under-estimates interference.

e) split tree into parts and recurse on forest



# From Stars to Trees

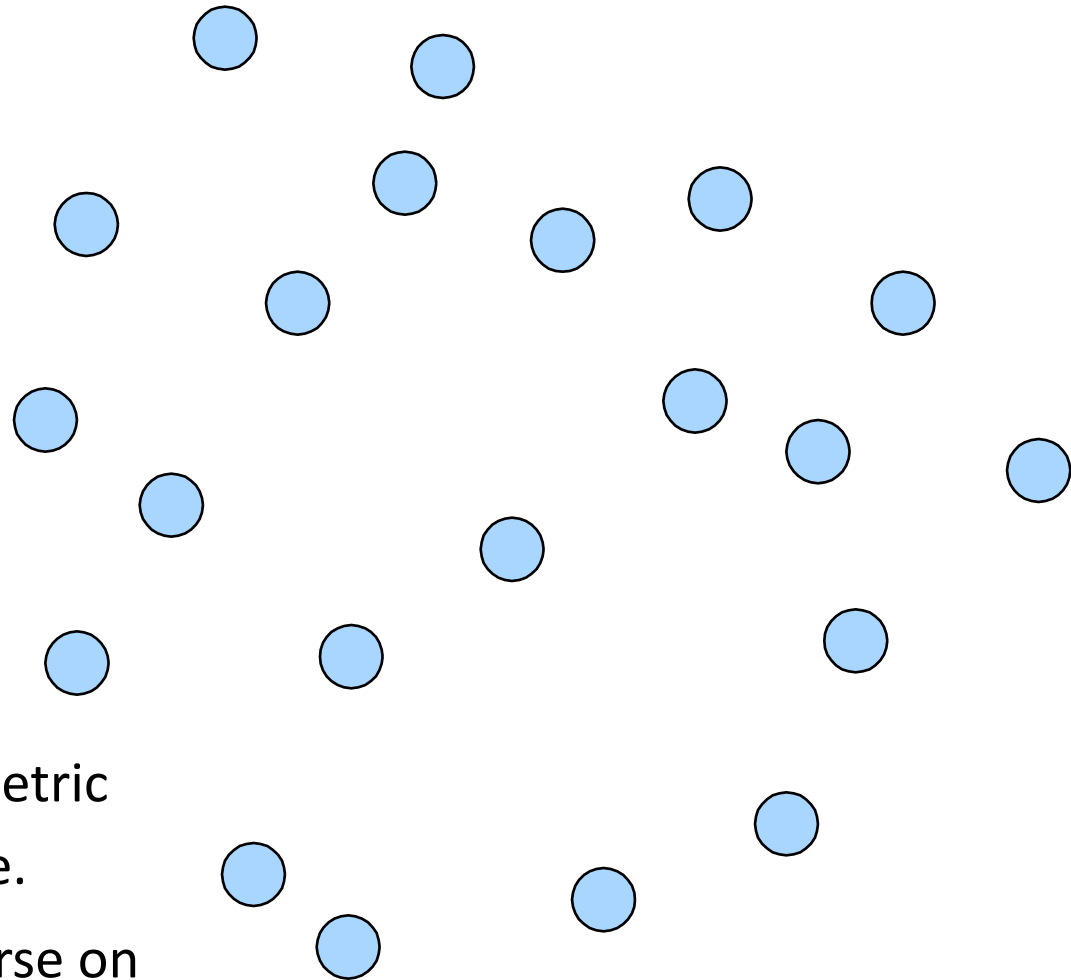
a) choose a center  $c$ , such that after the removal of  $c$  every part contains less than half of the nodes.

b) consider the star in which all nodes are connected to the center  $c$ .

c) solve the problem for this star.

d) maybe you accepted to many nodes since the star-metric under-estimates interference.

e) split tree into parts and recurse on forest



# From Stars to Trees

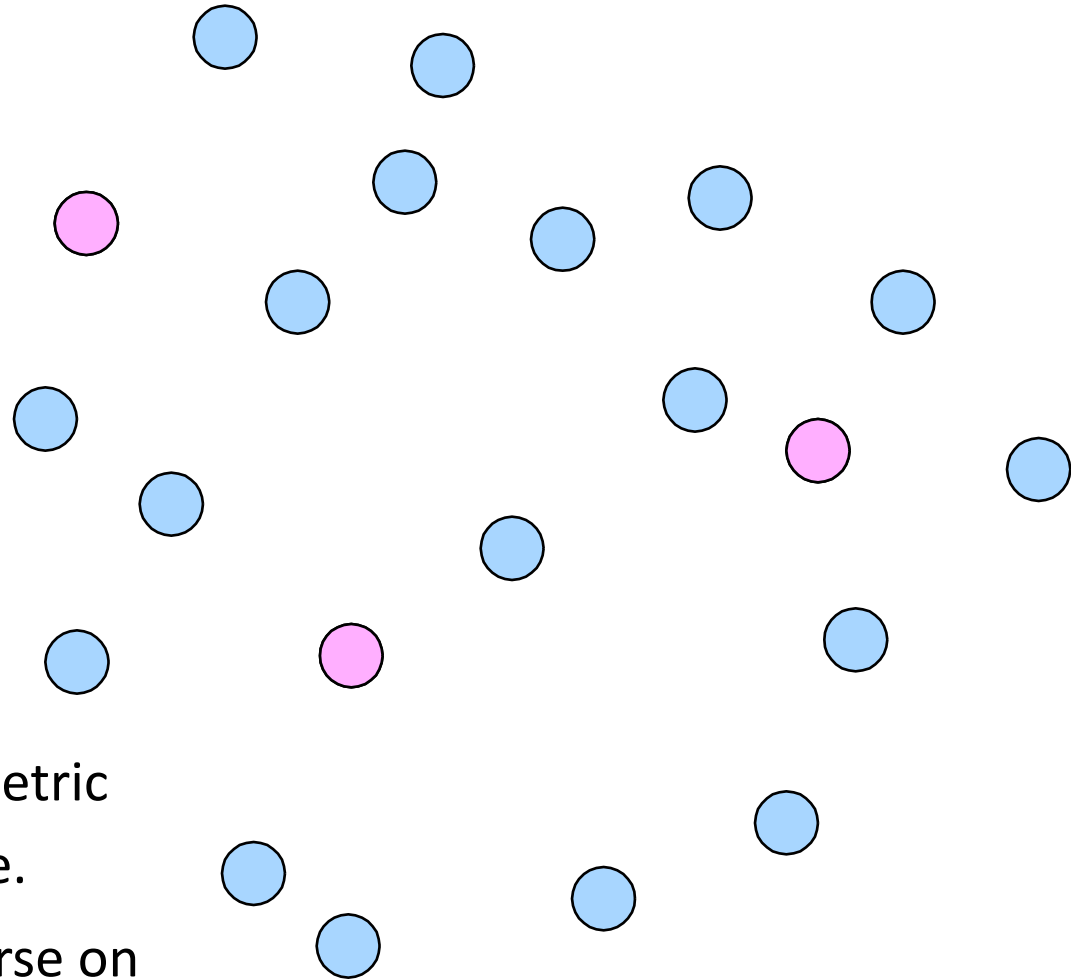
a) choose a center  $c$ , such that after the removal of  $c$  every part contains less than half of the nodes.

b) consider the star in which all nodes are connected to the center  $c$ .

c) solve the problem for this star.

d) maybe you accepted to many nodes since the star-metric under-estimates interference.

e) split tree into parts and recurse on forest



# From Stars to Trees

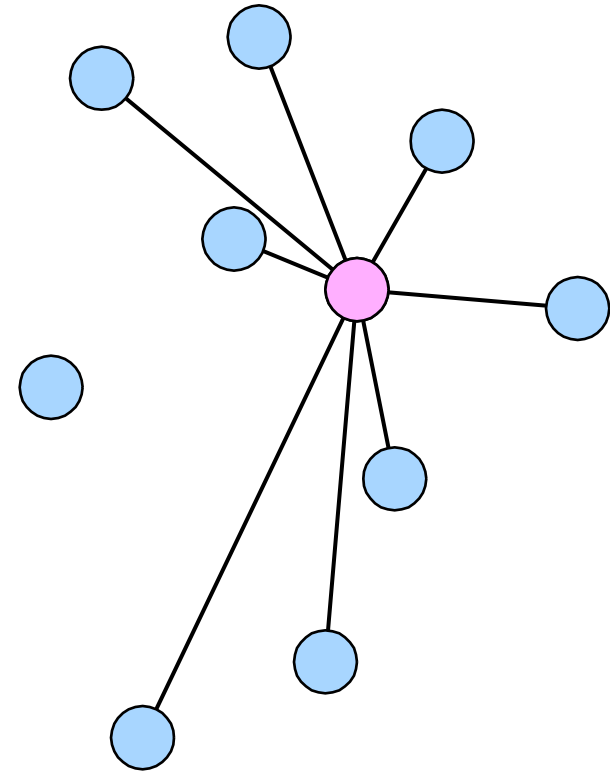
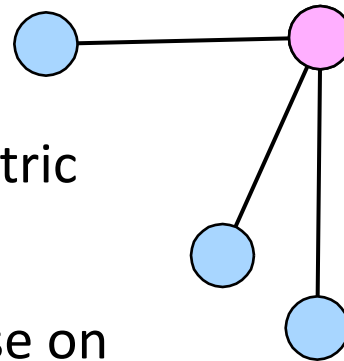
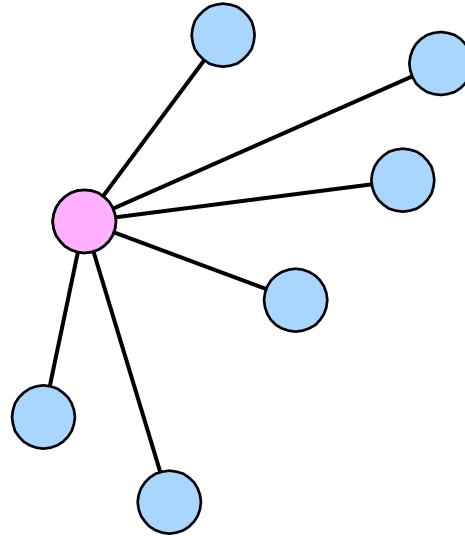
a) choose a center  $c$ , such that after the removal of  $c$  every part contains less than half of the nodes.

b) consider the star in which all nodes are connected to the center  $c$ .

c) solve the problem for this star.

d) maybe you accepted to many nodes since the star-metric under-estimates interference.

e) split tree into parts and recurse on forest



# From Stars to Trees

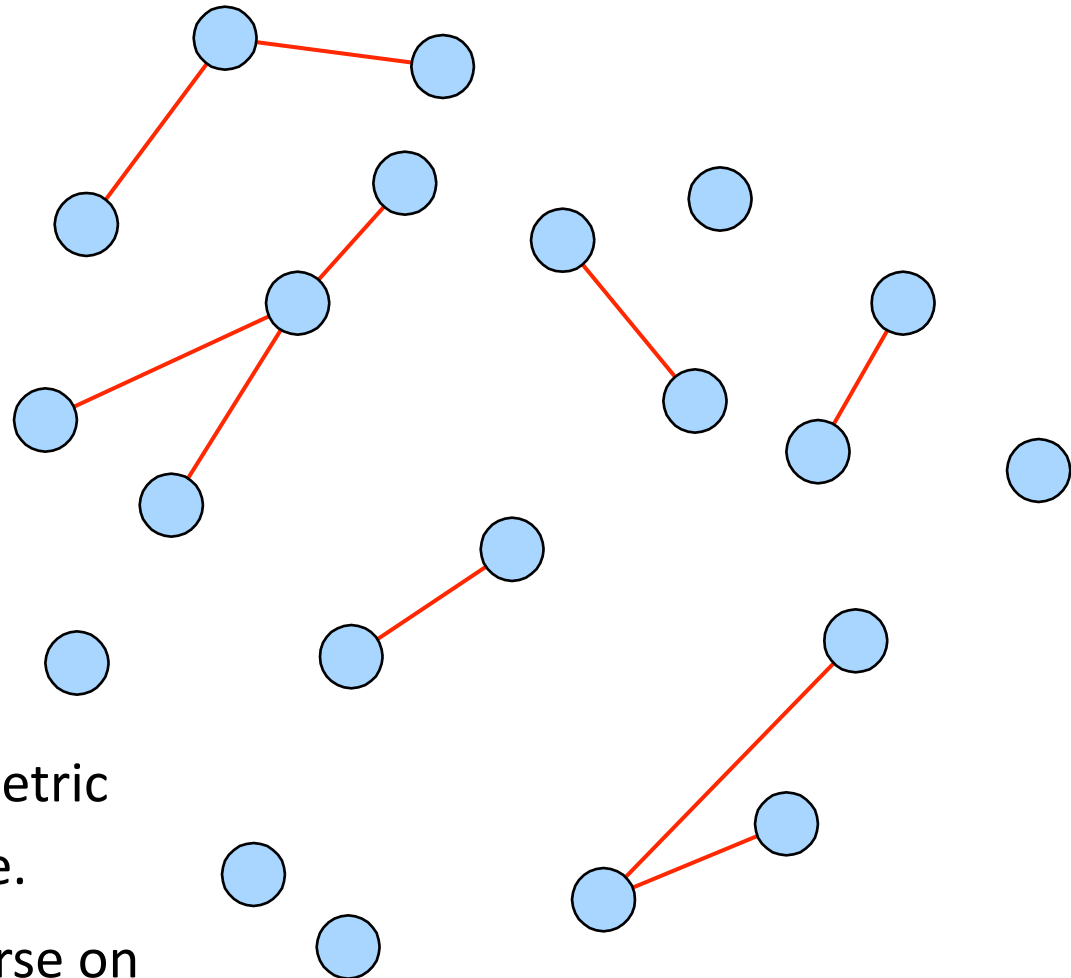
a) choose a center  $c$ , such that after the removal of  $c$  every part contains less than half of the nodes.

b) consider the star in which all nodes are connected to the center  $c$ .

c) solve the problem for this star.

d) maybe you accepted to many nodes since the star-metric under-estimates interference.

e) split tree into parts and recurse on forest



# From Stars to Trees

# From Stars to Trees

- c) For every pair of nodes  $(u,v)$  there exists one forest such that the distance between  $u$  and  $v$  is correct in this forest.
- d) Let  $U$  denote the set of nodes pairs that are scheduled in **every** forest. The total interference induced by  $U$  on a node  $v$  (summed over all forests) is an **upper bound** on the real interference induced when scheduling  $U$  in the original metric.
- e) This interference is only  $O(\log n) \frac{1}{\beta' \sqrt{\ell_v}}$ .
- f) This means that the set is  $\Omega(\frac{\beta'}{\log n})$ -feasible.

# From Trees to General Metrics

## Theorem [Gupta, Hajiaghayi, R 2006]

Given a finite metric space  $(V, d)$  there exist  $r = O(\log |V|)$  edge-weighted trees  $T_1, \dots, T_r$  with node set  $V$  such that the following holds.

a) For every pair  $(u, v) \in V^2$  and for every tree  $T_i$   
 $d(u, v) \leq d_{T_i}(u, v)$

d) For every node  $v$  there exists a subset  $\mathcal{T}_v \subseteq \{T_1, \dots, T_r\}$  with  $|\mathcal{T}_v| \geq \frac{9}{10}r$  such that the pairwise distances involving  $v$  are only stretched by a logarithmic factor, i.e.,

$$\forall T \in \mathcal{T}_v, u \in V : d_T(u, v) \leq O(\log |V|) \cdot d(u, v)$$

# From Trees to General Metrics

## Theorem [Gupta, Hajiaghayi, R 2006]

Given a finite metric space  $(V, d)$  there exist  $r = O(\log |V|)$  edge-weighted trees  $T_1, \dots, T_r$  with node set  $V$  such that the following holds.

a) For every pair  $(u, v) \in V^2$  and for every tree  $T_i$   
$$d(u, v) \leq d_{T_i}(u, v)$$

d) For every node  $v$  there exists a subset  $\mathcal{T}_v \subseteq \{T_1, \dots, T_r\}$  with  $|\mathcal{T}_v| \geq \frac{9}{10}r$  such that the pairwise distances involving  $v$  are only stretched by a logarithmic factor, i.e.,

$$\forall T \in \mathcal{T}_v, u \in V : d_T(u, v) \leq O(\log |V|) \cdot d(u, v)$$

# From Trees to General Metrics

# From Trees to General Metrics

## Observation.

Let for a tree  $T_i$ , the core  $C_i$  be the subset of nodes whose distances are not stretched too much.

- a) there must exist a tree that contains at least 9/10 of the original  $\beta$ -feasible node-set  $U$  in the core. Let  $V$  denote the intersection between  $U$  and the core.
- b) for this set the tree-algorithm will find an  $\Omega(\frac{\beta'}{\log n})$ -feasible subset that contains at least  $\Omega(1 - \frac{1}{\text{polylog}n})|V|$  nodes.
- c) being feasible on the tree means that one is “nearly” feasible in the original metric since the tree underestimates decays only by a factor  $\log^\alpha n$

# Interference Scheduling Problem

## Theorem III.

There is a polynomial time colouring algorithm for the square root power assignment that gives an  $\sqrt{2}$ -approximation to the optimum number of colours required for  $\mathcal{G}$ .

# Interference Scheduling Problem

## Theorem III.

There is a polynomial time colouring algorithm for the square root power assignment  $\bar{p}$  that gives an  $O(\log n)$ -approximation to the optimum number of colours required for  $\bar{p}$ .

# Open Problems

# Open Problems

## Observation.

- a) what happens if we allow different power levels for both directions in the bidirectional version
- b) so far only colouring and power assignment is considered. How to incorporate multi-hop routing into the model?