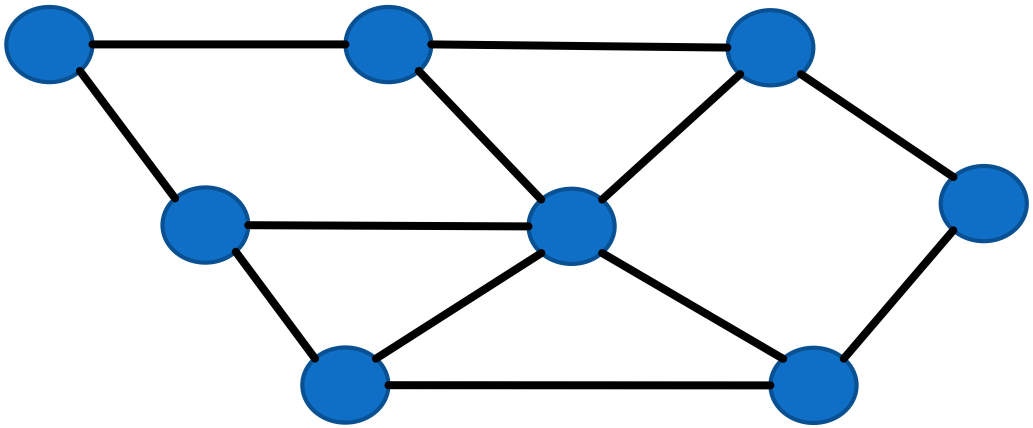
Distributed Approximation for Tree Augmentation

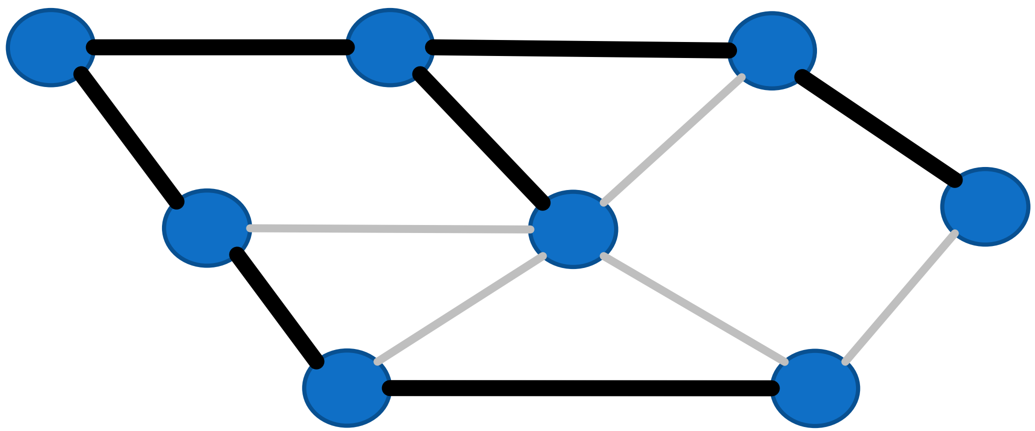
Michal Dory, Technion

Joint work with: Keren Censor-Hillel, Technion

Consider a Communication Network

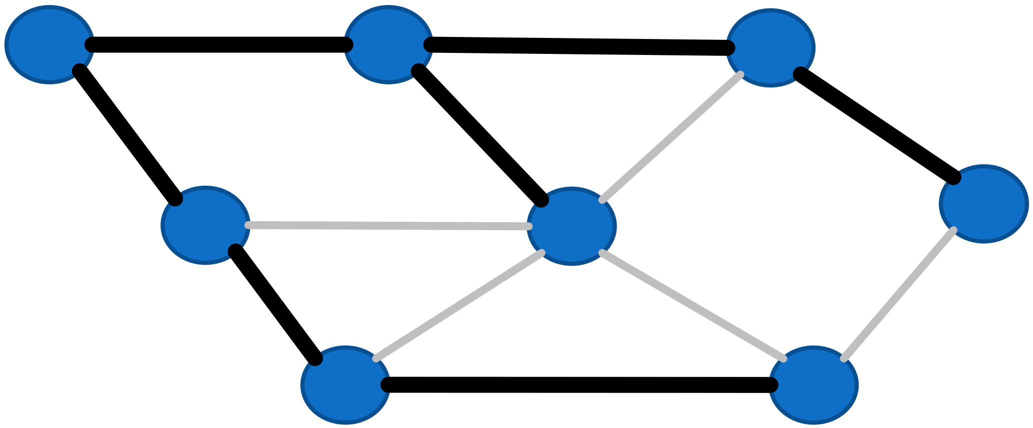


We can communicate over a spanning tree

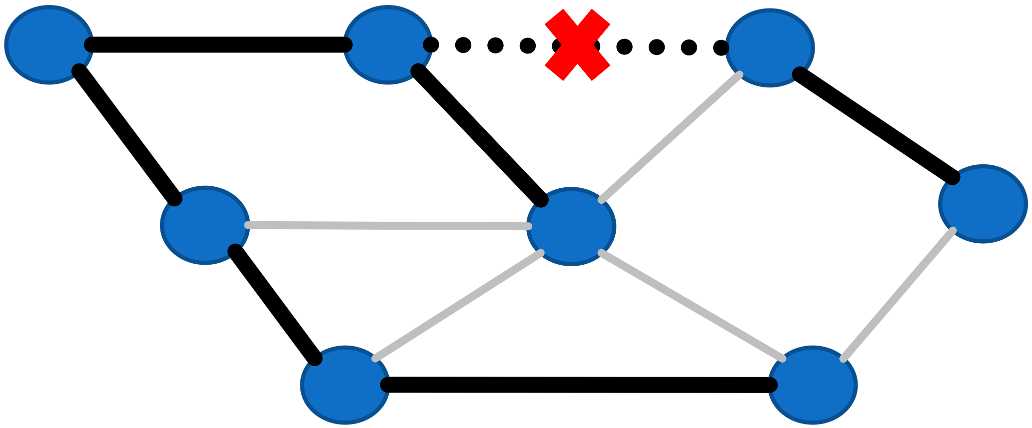


There are many constructions of minimum spanning trees (MST)

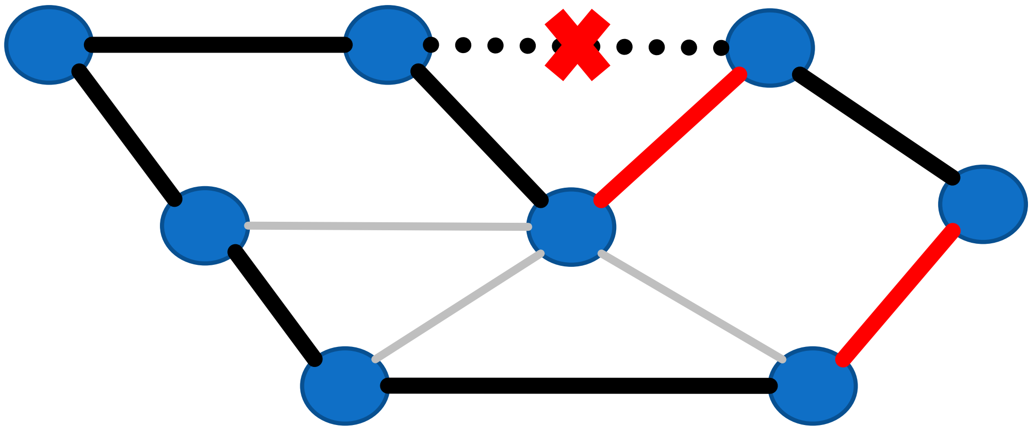
[Gallager, Humblet and Spira 83, Kutten and Peleg 95, Garay, Kutten and Peleg 98, Pandurangan, Robinson and Scquizzato 17, Elkin 17,...]



Trees cannot survive a link failure

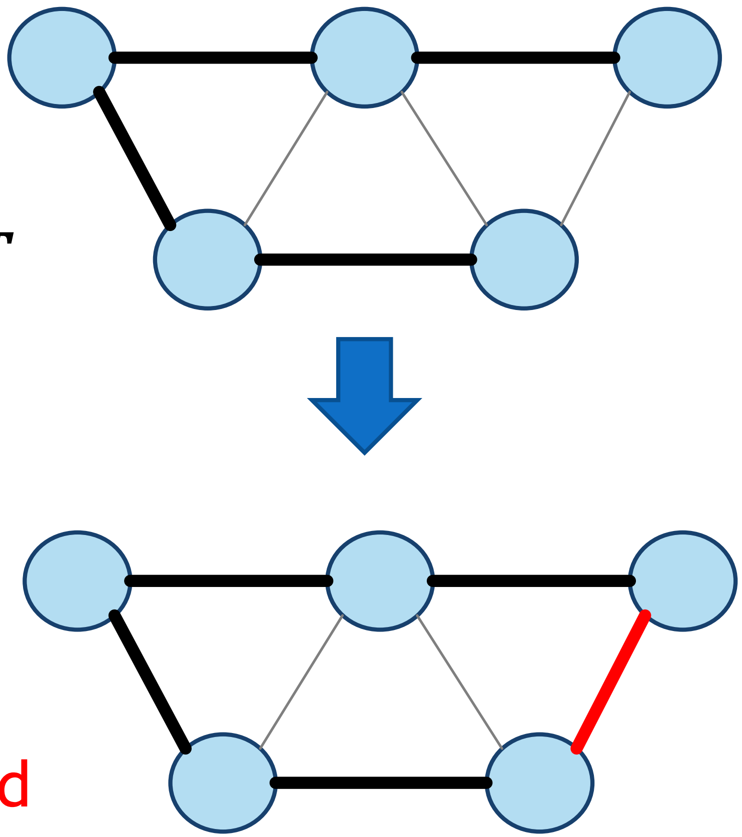


Trees cannot survive a link failure



The Tree Augmentation Problem (TAP)

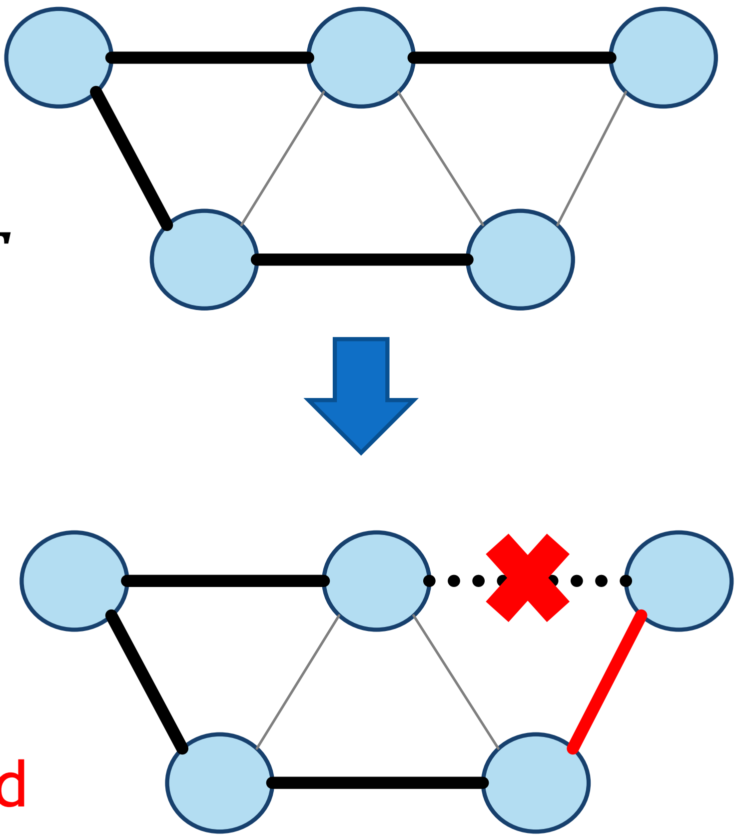
Input:
a graph G
a spanning tree $\boldsymbol{T}$



Goal:
augment $\boldsymbol{T}$ to be
2-edge-connected

The Tree Augmentation Problem (TAP)

<u>Input:</u>
a graph G
a spanning tree $\boldsymbol{T}$



<u>Goal:</u>
augment $\boldsymbol{T}$ to be
2-edge-connected

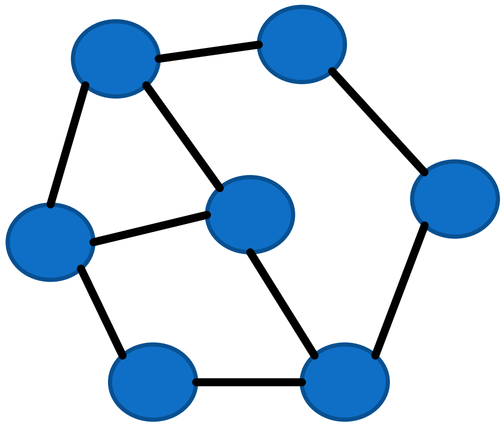
The Tree Augmentation Problem (TAP)

- A central problem in network design
- Has many **sequential** algorithms:

  [Frederickson and JáJá 81, Khuller and Thurimella 93, Goemans et al. 94, Jain 01, Kortsarz and Nutov 16, Adjiashvili 17…].
- Goal: solve TAP in the **distributed** setting.

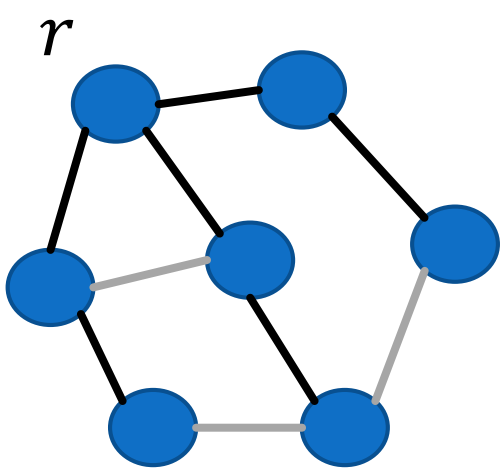
The CONGEST model

- Communication network with n processors
- Synchronous rounds
- Messages of $O(\log n)$ bits
- Time = number of rounds
- The input and output are local



The CONGEST model

- Communication network with n processors
- Synchronous rounds
- Messages of $O(\log n)$ bits
- Time = number of rounds
- The input and output are local

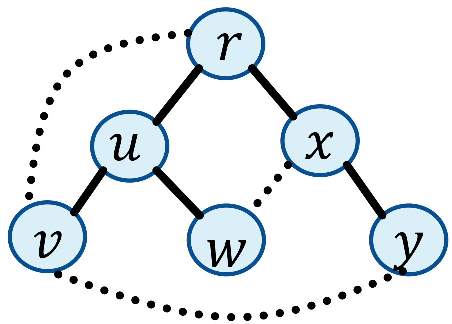


Sequential Approximation for TAP
[Khuller and Thurimella 93]

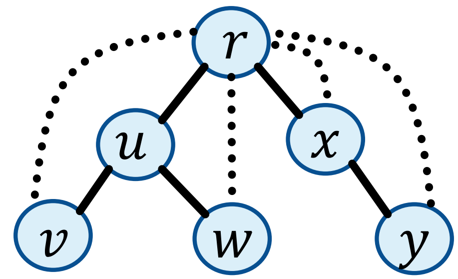
- Builds a new **virtual graph** G'
- Finds a **directed MST** in G'
- This gives a 2-approximation in G

Distributed Approximation for TAP

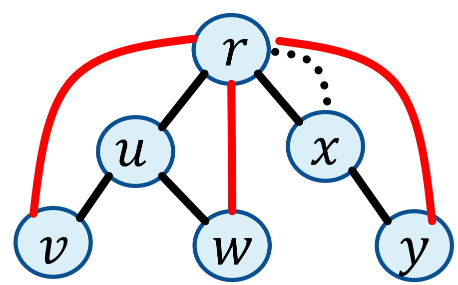
The input graph G



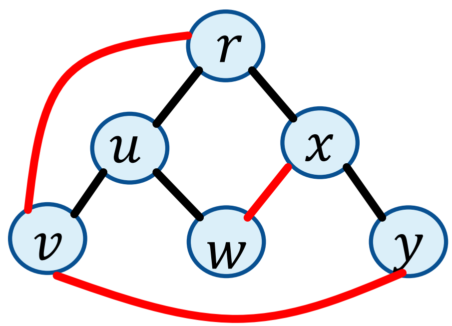
A virtual graph G'



Optimal solution in G'



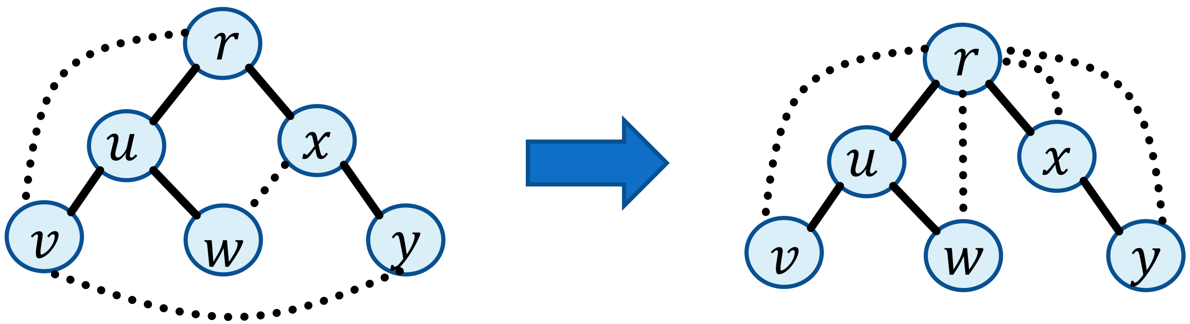
2-approximation in G



The Graph G'

- All the non-tree edges in G' are between ancestors to descendants:

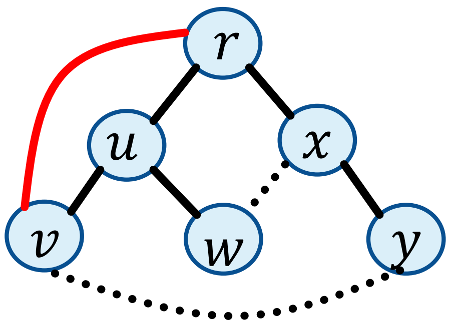
The input graph G A virtual graph G'



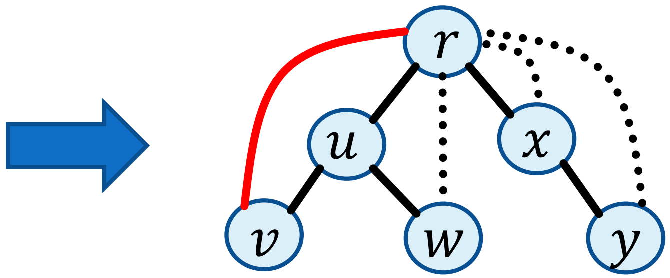
The Graph G'

- All the non-tree edges in G' are between ancestors to descendants:

The input graph G



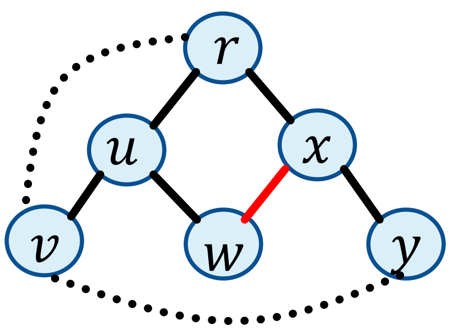
A virtual graph G'



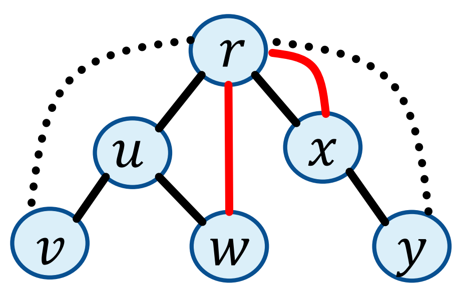
The Graph G'

- All the non-tree edges in G' are between ancestors to descendants:

The input graph G



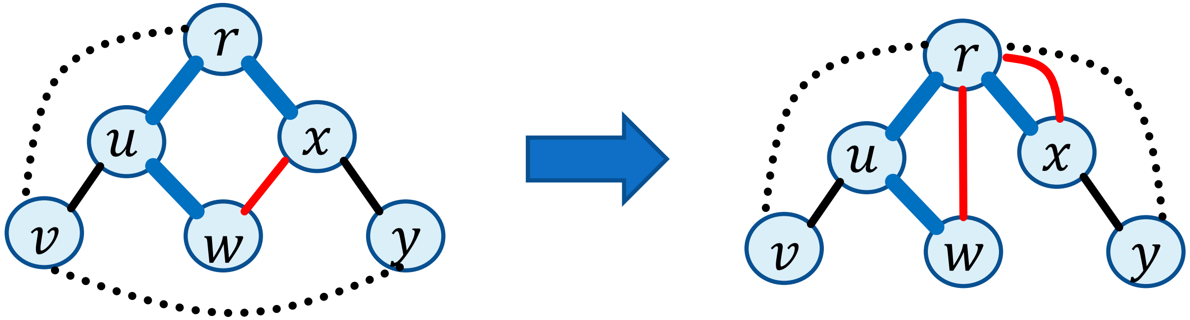
A virtual graph G'



The Graph G'

- The corresponding edges in G' *cover* exactly the same tree edges:

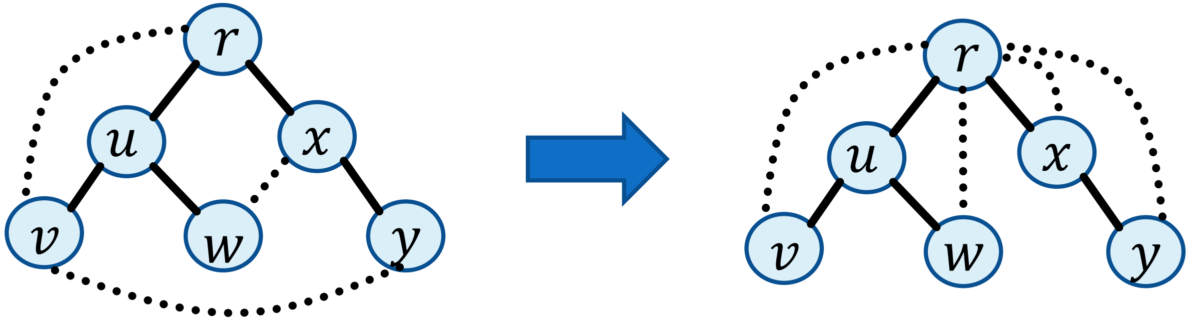
The input graph G A virtual graph G'



The Graph G'

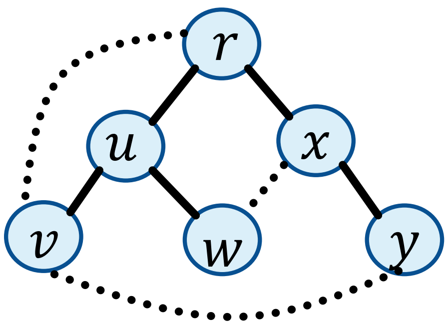
- We can build G' in $O(h)$ rounds using LCA labels of $O(\log n)$ bits [Alstrup et al. 2004]

The input graph G A virtual graph G'

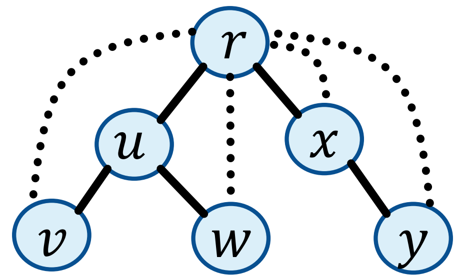


Distributed Approximation for TAP

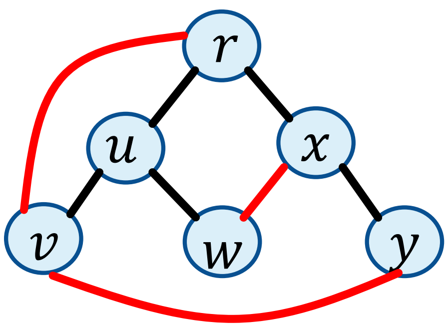
The input graph G



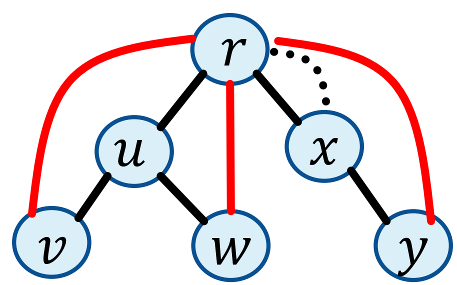
A virtual graph G'



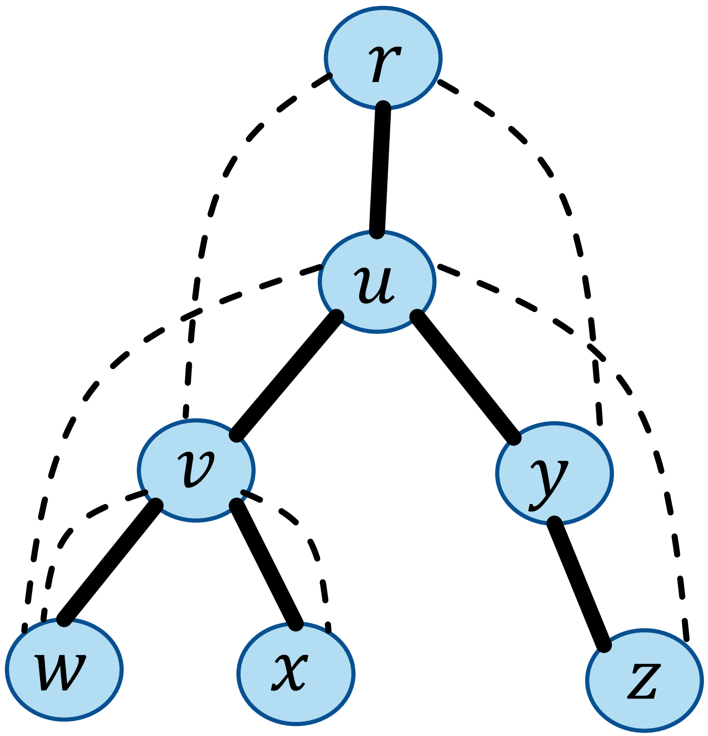
2-approximation in G



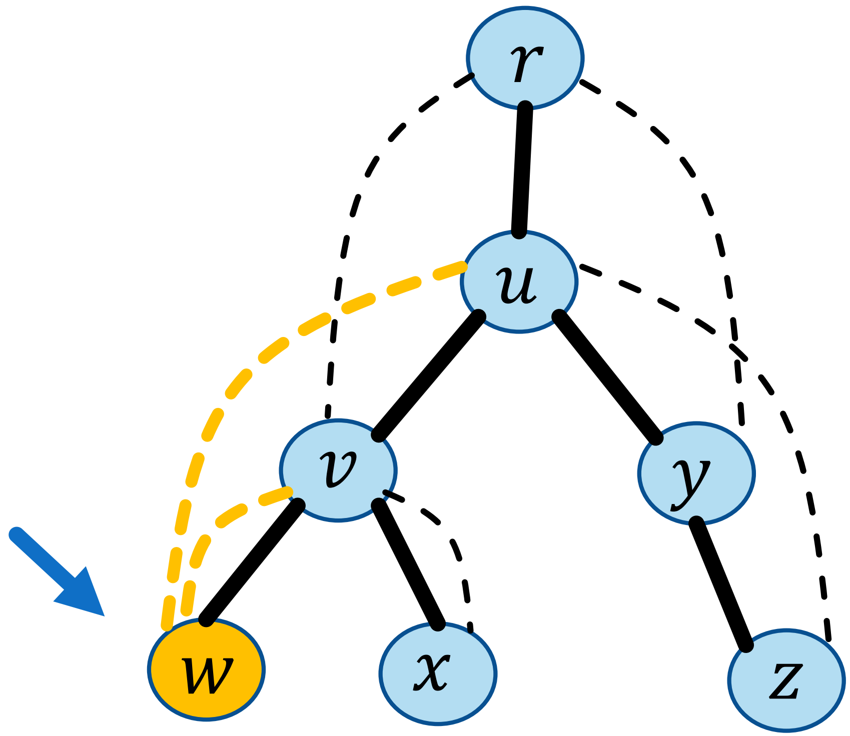
Optimal solution in G'



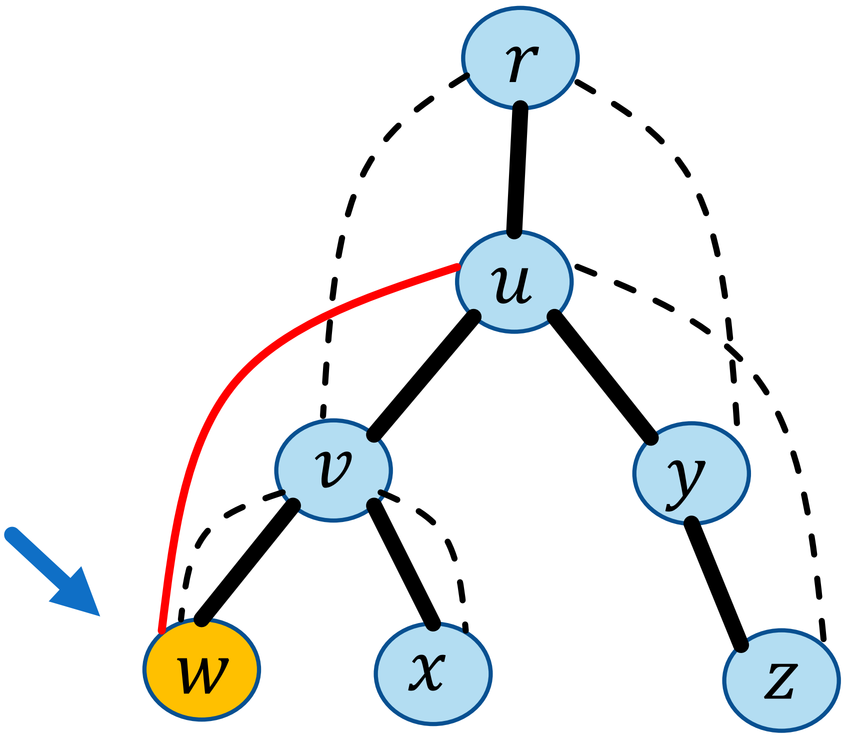
Finding an Optimal Augmentation in G'



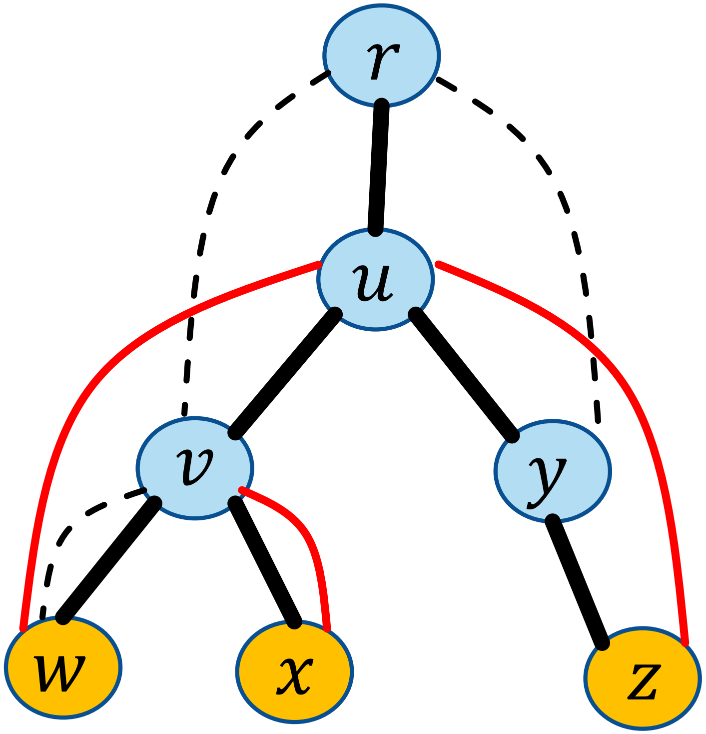
Finding an Optimal Augmentation in G'



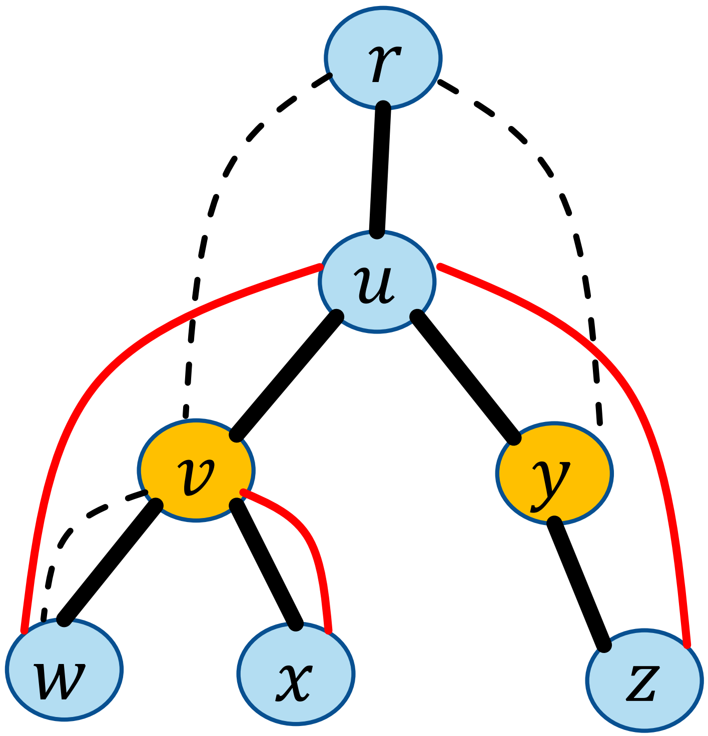
Finding an Optimal Augmentation in G'



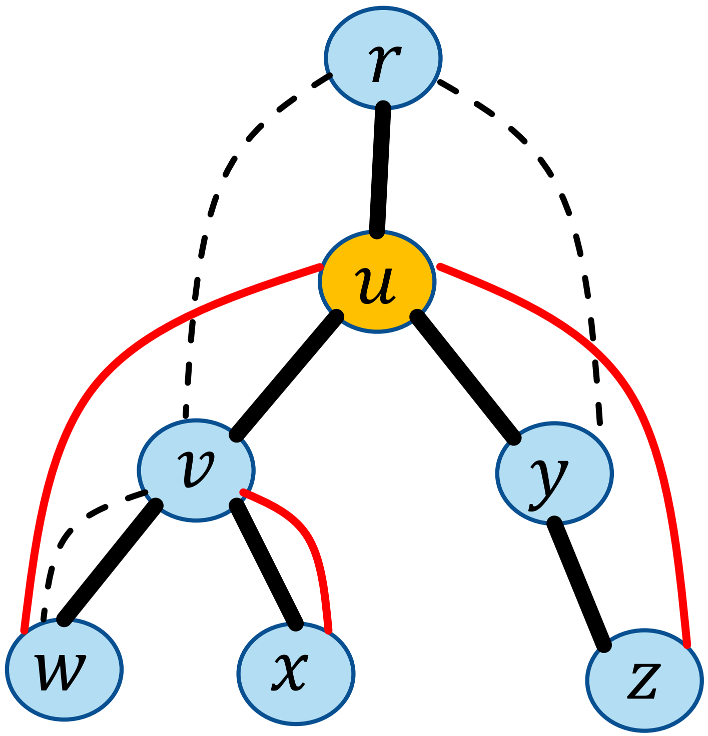
Finding an Optimal Augmentation in G'



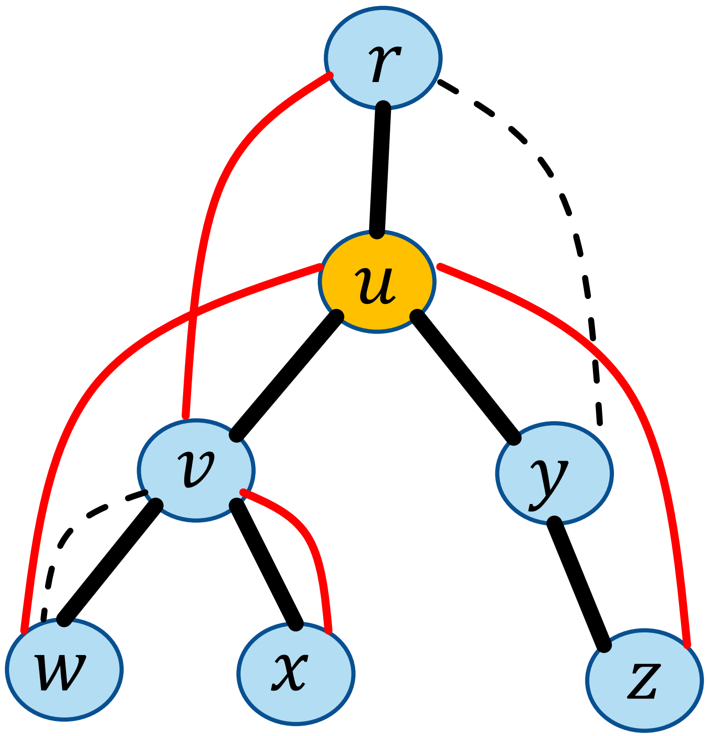
Finding an Optimal Augmentation in G'



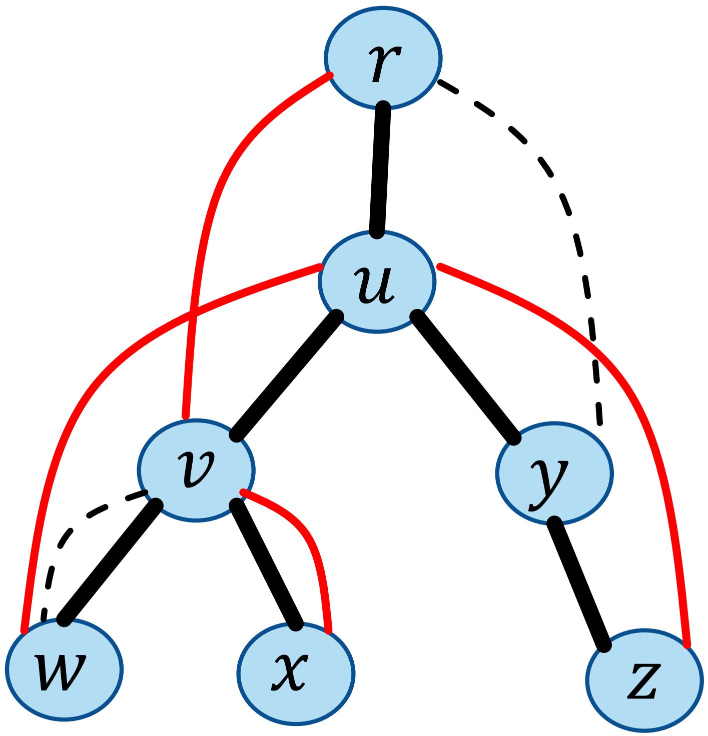
Finding an Optimal Augmentation in G'



Finding an Optimal Augmentation in G'



Finding an Optimal Augmentation in G'



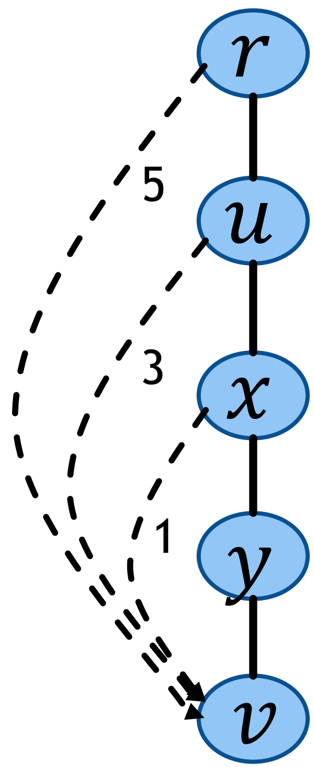
Conclusion

- There is a **2-approximation** for *unweighted* TAP in $\boldsymbol{O(h)}$ rounds, where h = height of T.
- What about the weighted case?

Weighted TAP

<u>Problem:</u>

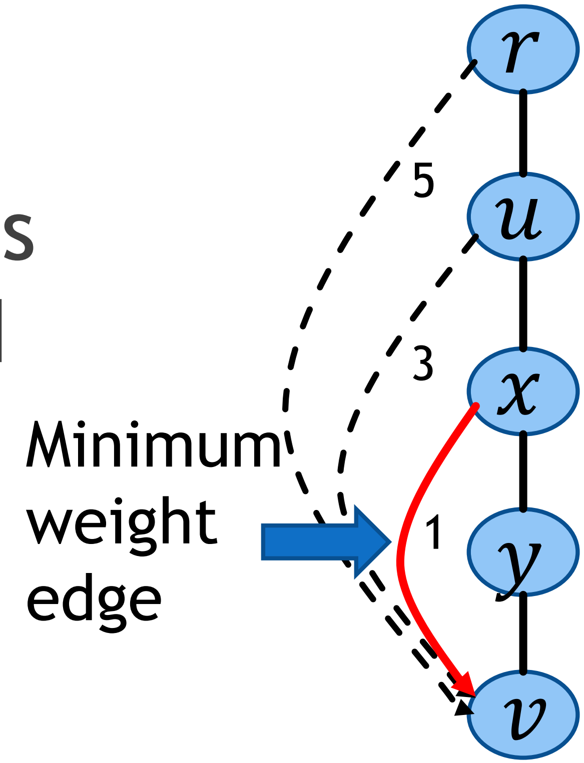
how to compare edges?



Weighted TAP

<u>Solution:</u>

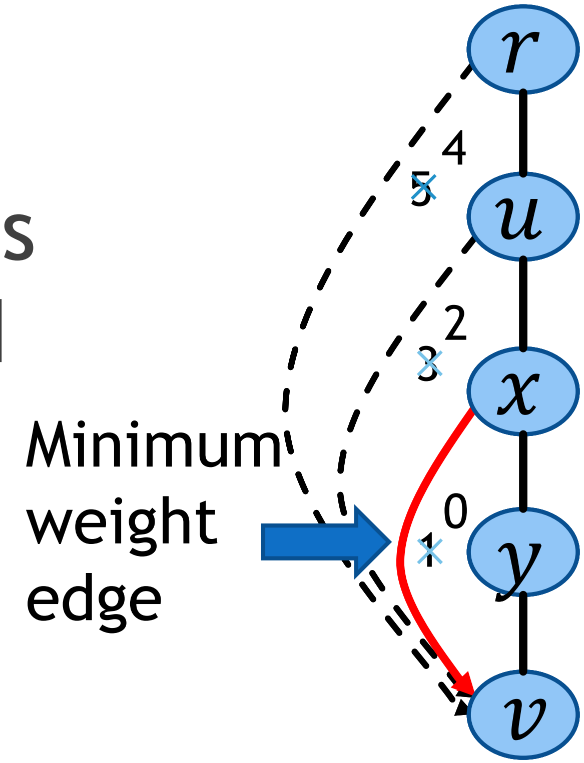
sending edges with reduced weights



Weighted TAP

Solution:

sending edges with reduced weights



Conclusion

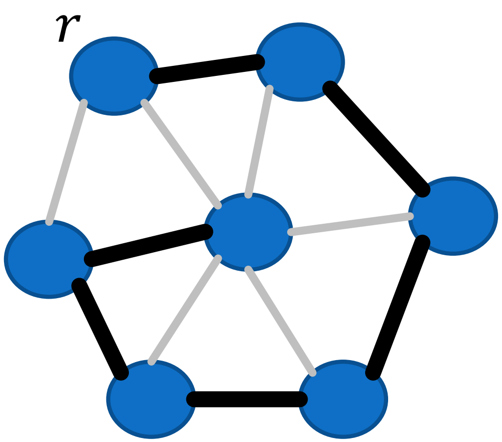
- There is a **2-approximation** for *weighted* TAP in $\boldsymbol{O(h)}$ rounds, where h = height of T.

Is This Optimal?

- TAP is a global problem which requires $\boldsymbol{\Omega(D)}$ rounds, where D = diameter of G
- If $h = O(D)$ our algorithms are optimal up to a constant factor

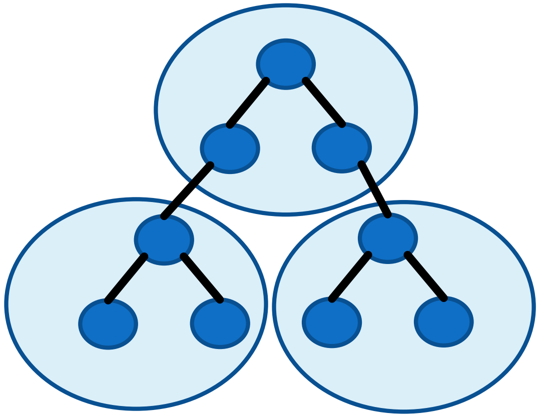
What about the case $h = \omega(D)$?

- We show an $\boldsymbol{\Omega(h)}$ lower bound for *weighted* TAP when $D \approx \log n$, $h \approx \sqrt{n}$



What about the case $h = \omega(D)$?

- We show a **4-approximation** for unweighted TAP in $\tilde{O}(\sqrt{n} + D)$ rounds

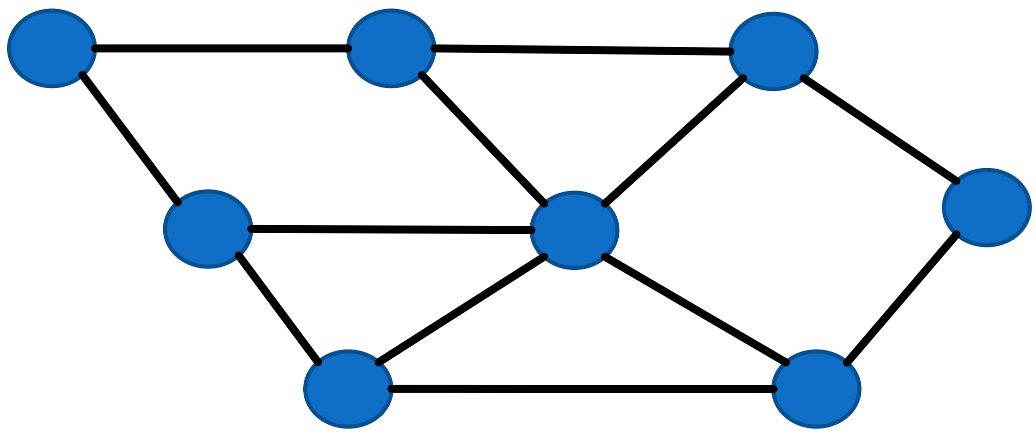


Main Results

- **2-approximation** for unweighted or weighted TAP in $\boldsymbol{O(h)}$ rounds, where h = height of T
- **4-approximation** for unweighted TAP in $\boldsymbol{\tilde{O}(\sqrt{n} + D)}$ rounds, where D = diameter of G
- $\boldsymbol{\Omega(D)}$ lower bound
- $\boldsymbol{\Omega(h)}$ lower bound for weighted TAP when $D \approx \log n$, $h \approx \sqrt{n}$

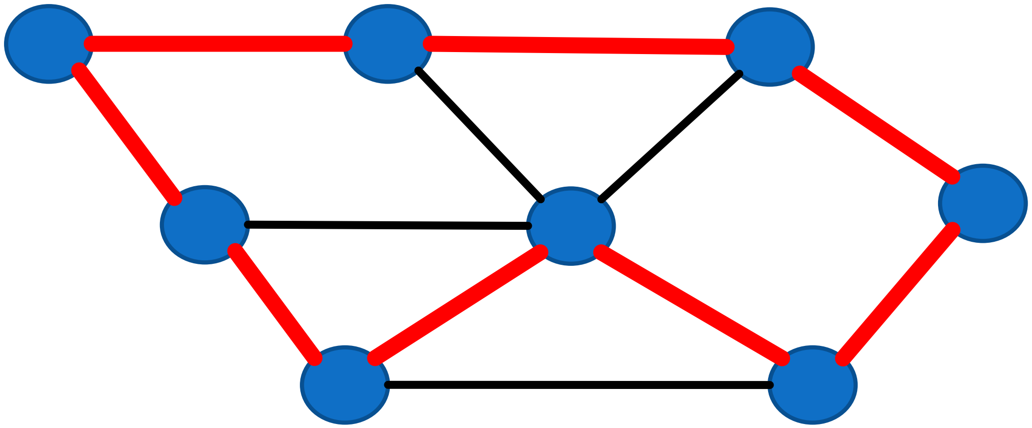
<u>Application:</u> minimum size 2-Edge-Connected Subgraph (2-ECSS)

- <u>Goal:</u> find the minimum size 2-ECSS
- No spanning tree is given



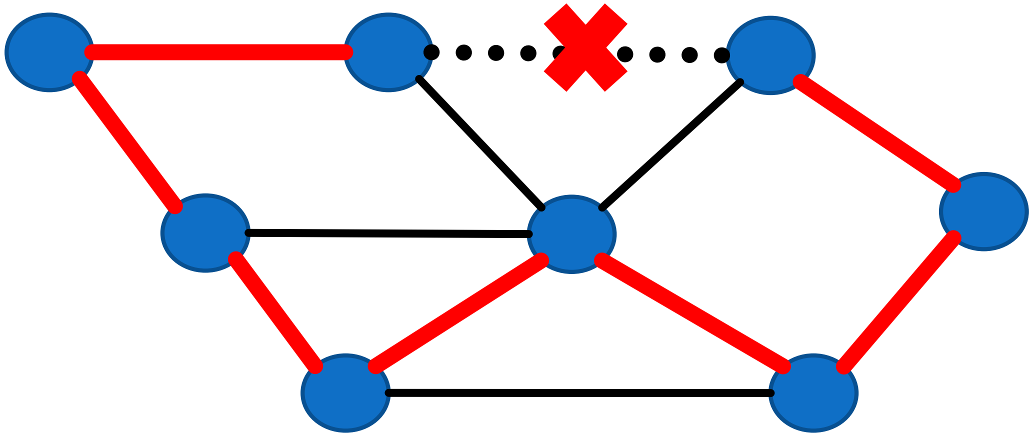
Application: minimum size 2-Edge-Connected Subgraph (2-ECSS)

- Goal: find the minimum size 2-ECSS
- No spanning tree is given



Application: minimum size 2-Edge-Connected Subgraph (2-ECSS)

- Goal: find the minimum size 2-ECSS
- No spanning tree is given



Application: minimum size 2-Edge-Connected Subgraph (2-ECSS)

Previous algorithms:

- $\frac{3}{2}$-approximation in $O(n)$ rounds [Krumke et al. 07]
- 2-approximation in $\tilde{O}(D+\sqrt{n})$ rounds [Thurimella 95]

Application: minimum size 2-Edge-Connected Subgraph (2-ECSS)

Previous algorithms:

- $\frac{3}{2}$-approximation in $O(n)$ rounds [Krumke et al. 07]
- 2-approximation in $\tilde{O}(D+\sqrt{n})$ rounds [Thurimella 95]

Using our TAP algorithm:

- 2-approximation in $O(D)$ rounds

Application: minimum ***weight*** 2-Edge-Connected Subgraph (2-ECSS)

Previous algorithms:

- 3-approximation in $O(n\log n)$ rounds

  [Krumke et al. 07]

Using our TAP algorithm:

- 3-approximation in $\tilde{O}(h_{MST} + \sqrt{n})$ rounds
- Lower bound of $\tilde{\Omega}(D + \sqrt{n})$ rounds for any polynomial approximation

Future Work

- Design efficient algorithms for *weighted* TAP and *weighted* 2-ECSS.
- Design **distributed** algorithms for additional **connectivity** problems:
  - Higher connectivity
  - Vertex connectivity