Optimum Broadcasting in Complete Weighted-Vertex Graphs

Shahin Kamali

UNIVERSITY OF Waterloo

David Cheriton School of Computer Science

University of Waterloo, Waterloo, ON

CANADA

Contents

- Problem definition
  - Weighted-Vertex Model
  - Previous Results

- Optimum broadcasting in complete weighted-vertex graphs
  - Polynomial Algorithm
  - Proof of the correctness (scheme)

Introduction

- Information dissemination problems
  - *Broadcasting (one-to-all communication)*
  - *Multicasting (one-to-many communication)*
  - *Gossiping (all-to-all communication)*
- Broadcasting:
  - A single message is sent from originator to all other members

Introduction

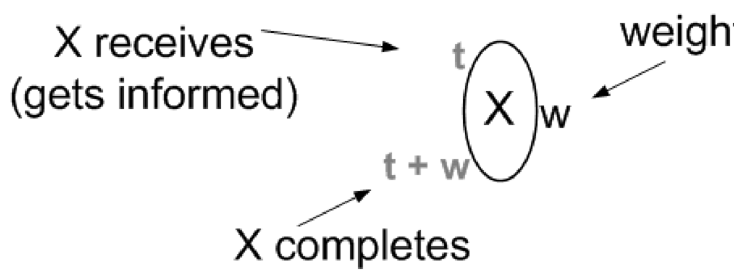
- Network is modelled by a graph $G = (V, E)$
- Communication occurs in discrete *rounds*
- The goal is to find a scheme which completes in minimum number of rounds
- Communication model
  - defines what may happens in a round

Introduction

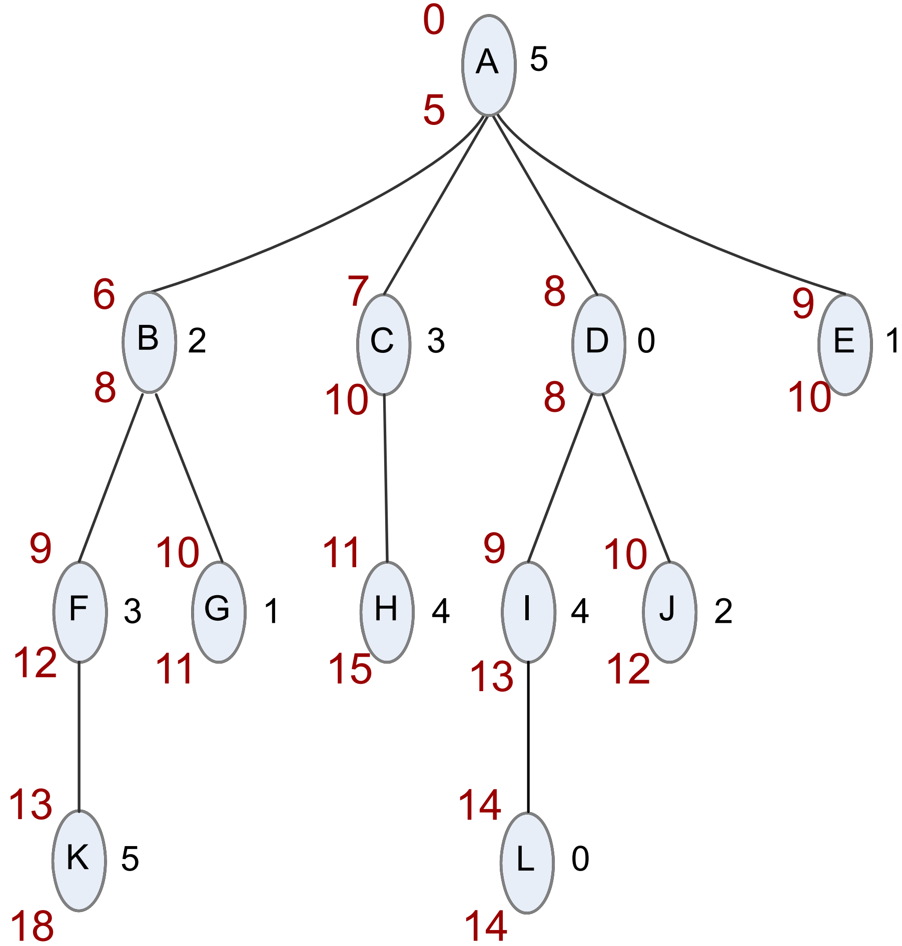
- Classical model of broadcasting
  - Each call involves one informed node and one of its uninformed neighbors
  - A node can participate in only one call per unit of time
  - In one unit of time, many calls can be performed in parallel
- Finding optimum scheme is NP_complete

Weighted-Vertex Model

- Each node needs to pass an internal delay, denoted by its weight, before passing the message to its neighbours
- The weights are non-negative integers
- Applications in parallel machines, and Satellite-Terrestrial (ST) networks
- The broadcasting completes when all nodes complete their internal delay



Weighted-Vertex Model

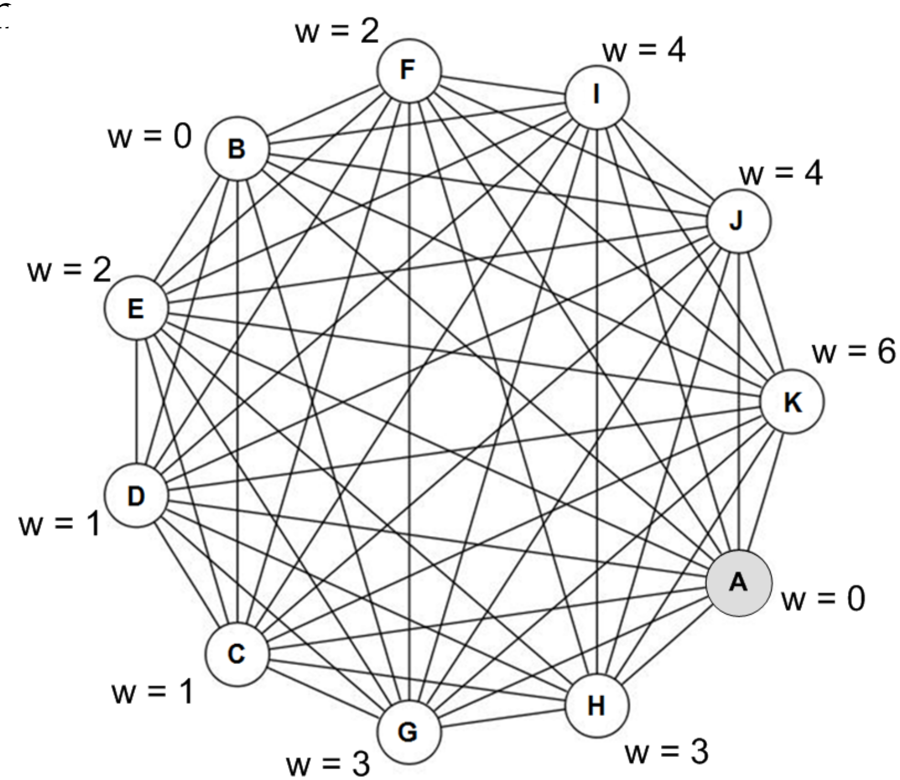


Existing Results

- NP-hardness
- Linear algorithm for trees (represent schemes with trees)
- Approximation results:
  - $O(b^* \log n + n^{1/2})$
  - $O(b^* \log n + n^{1/2})$ (b^* the optimum broadcast time)
- Better approximations exist for specific types of weighted graphs

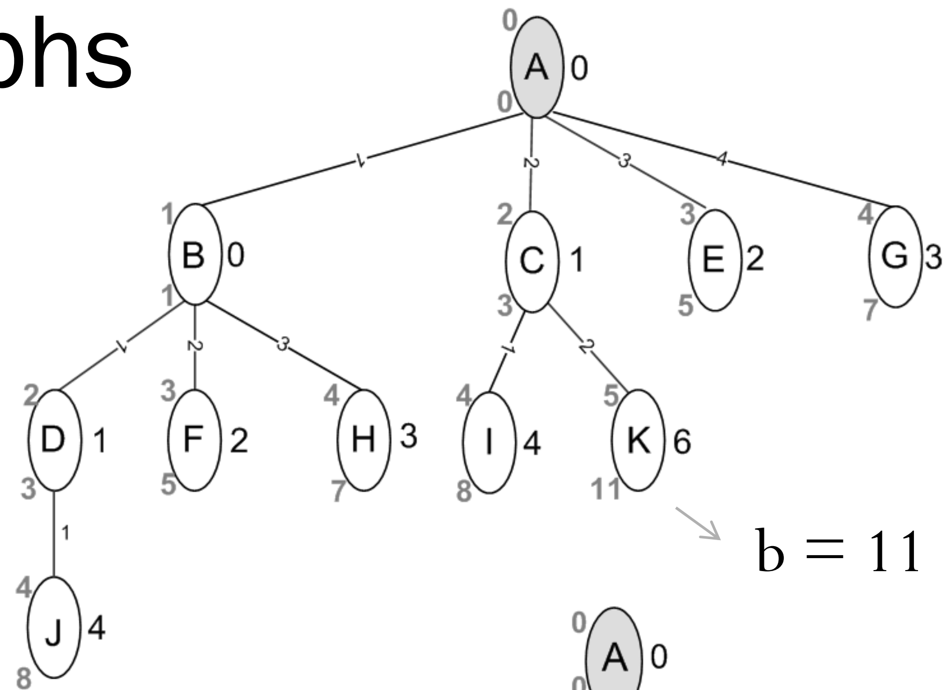
Broadcasting in Weighted-Vertex Complete Graphs

- No symmetry !
- Find a polynomial algorithm for finding t optimum scheme

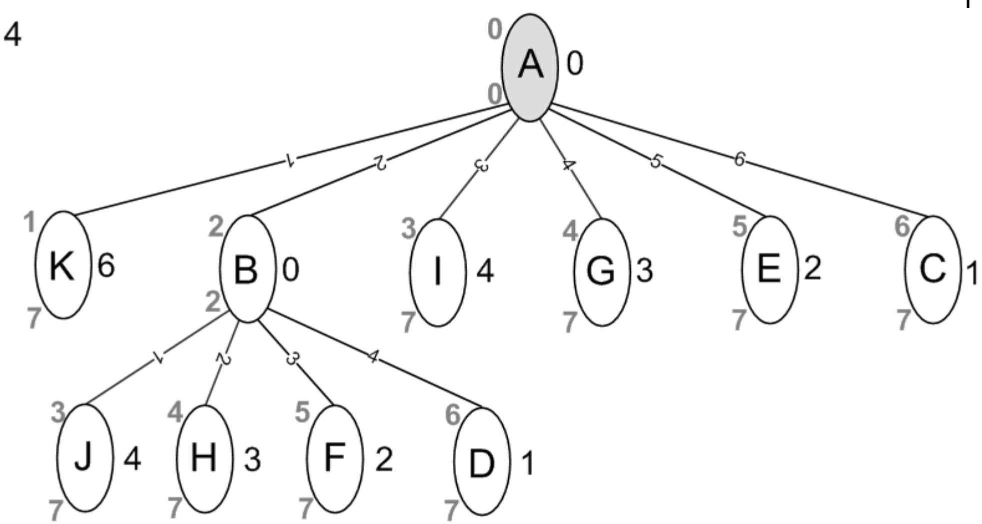


Broadcasting in Weighted-Vertex Complete Graphs

- Informing neighbours with minimum weight does Not work !



b = 11



b = 7

Algorithm Outline

- Algorithm *BroadGuess*
  - Input: instance of problem + guess value for broadcast time *br*
  - Output: A broadcast scheme (tree) completes in time *br* OR *BadGuess*
- Find the best scheme using a binary search approach
  - Upper bound for optimum broadcast time: $br_{opt} \leq n + 1 + W$
  - Call the algorithm with smaller or larger *br*

Output is a scheme which completes within *br*

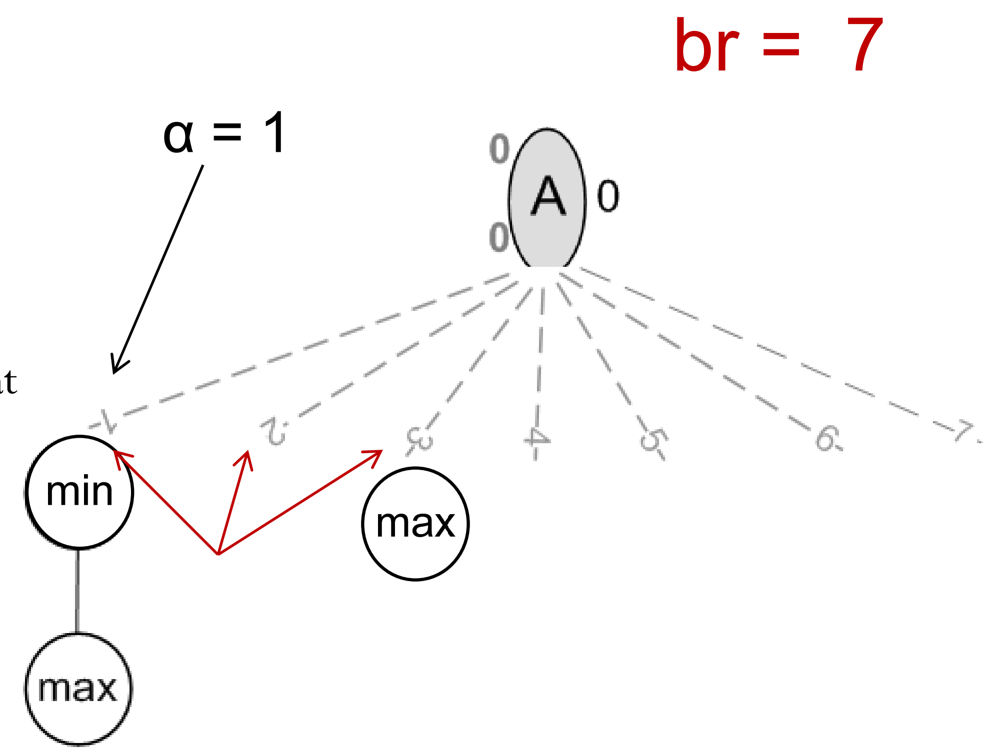
Output is BadGuess

BroadGuess

- Start with a broadcast tree initialized by originator
- Iteratively 'Stick' all other vertices to the tree
- In each iteration
  - Stick the uninformed vertex with minimum weight *(min)* or maximum weight *(max)*

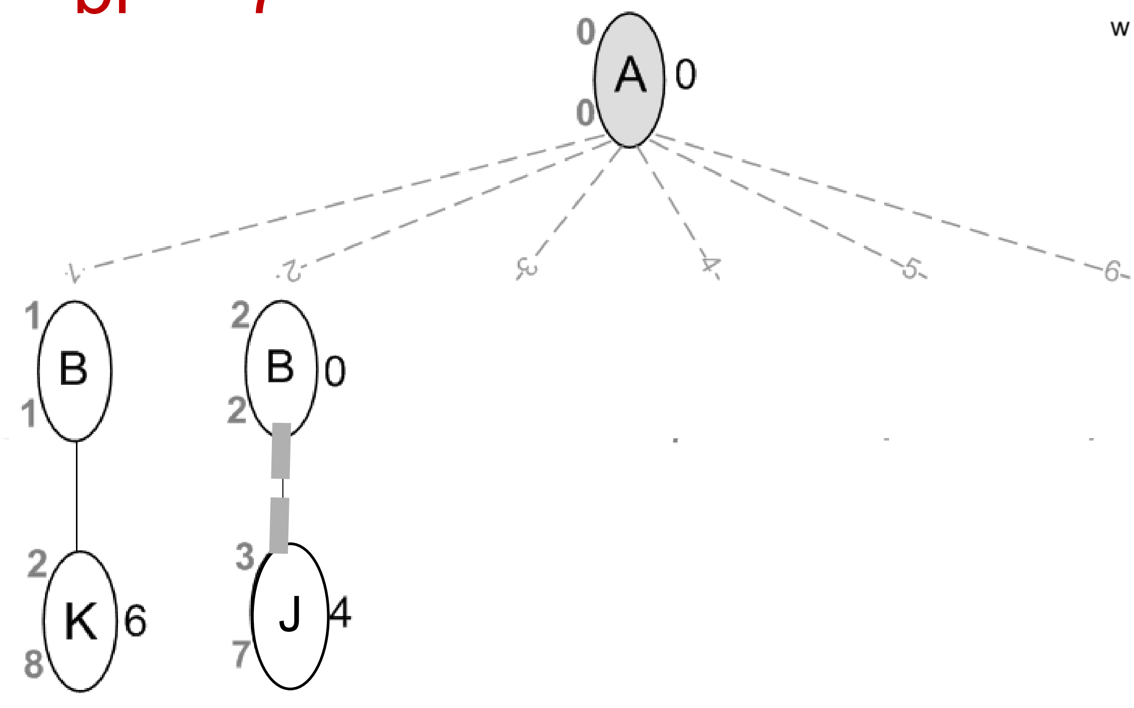
BroadGuess

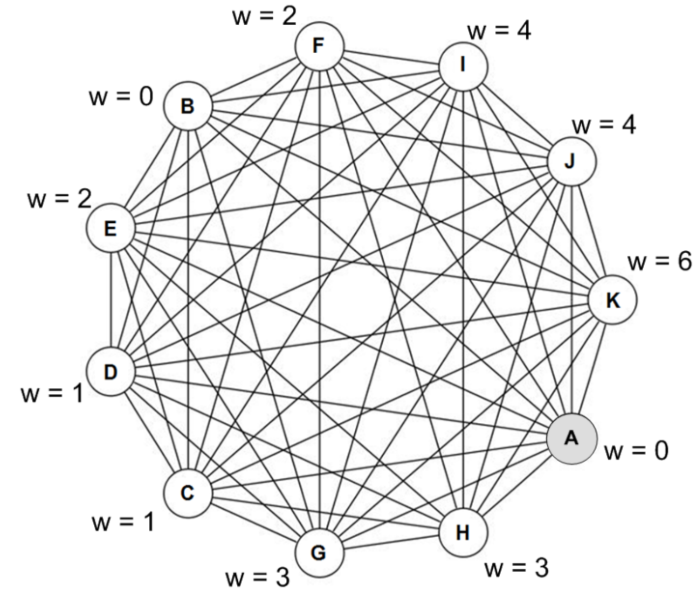
- Let α be the soonest time to inform an uninformed node
- If $\alpha + \omega(min) + 1 + \omega(max) \leq br$
  - Stick *min* to the tree to receive at time α
- Otherwise
  - Find those indices where $t(index) + \omega(max) \leq br$
  - If none, return BadGuess
  - Stick *max* to inform it as late as possible



BroadGuess

br = 7





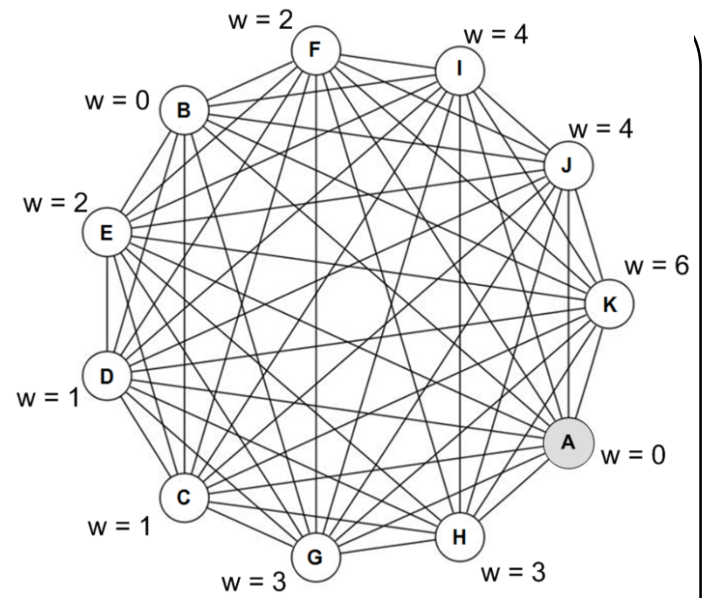
- α: 3
- min: C (1)
- max: J (4)

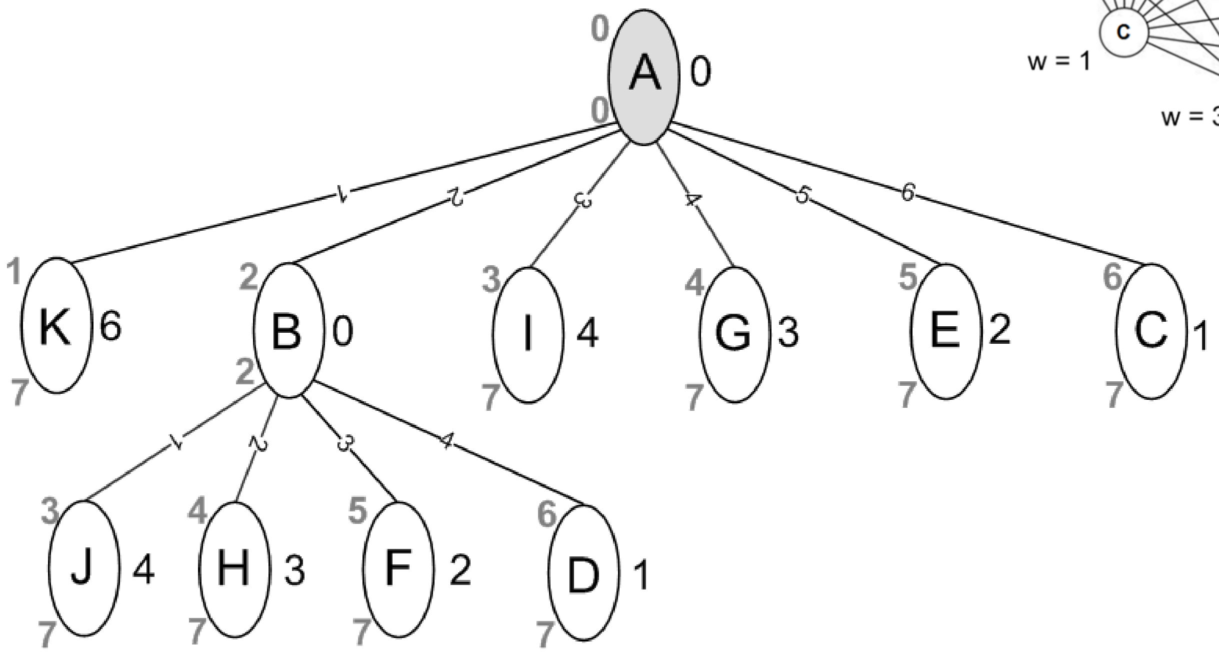
$\alpha + \omega(min) + 1 + \omega(max) =$

$3 + 1 + 1 + 4 = 9 > br$

BroadGuess

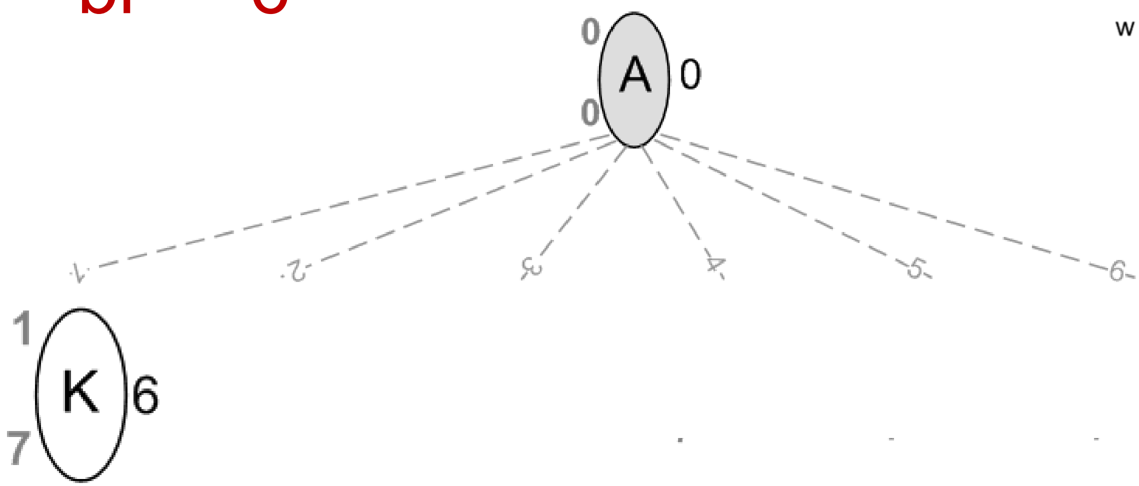
br = 7

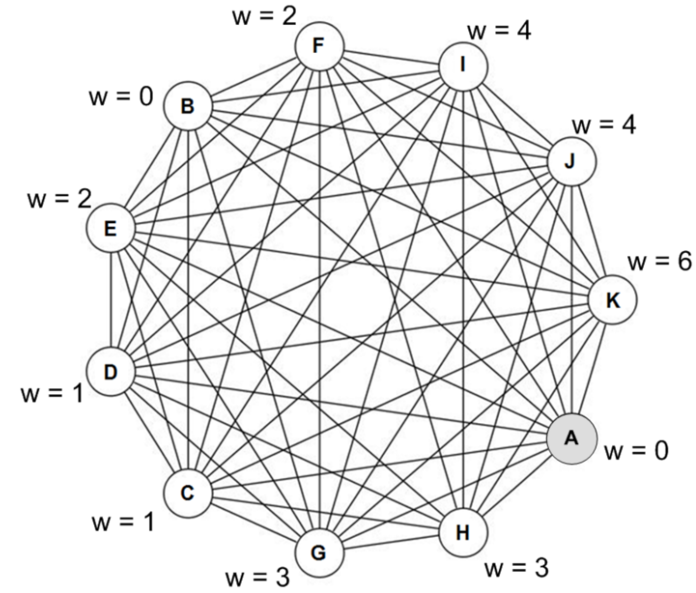




BroadGuess

br = 6





- α: 1
- min: B (0)
- max: K (6)

BadGuess

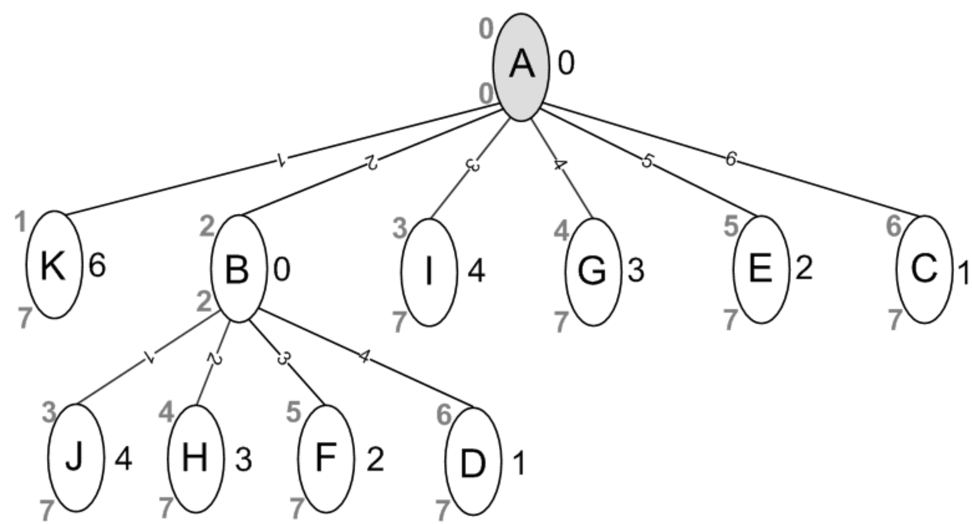
α + ω(min)+ 1+ω(max) =
1 + 0 + 1 + 6 = 8 > br

Proof of correctness

- The output trees complete within candidate broadcast time
- BadGuess → No tree with broadcast time *br* (hard)
  - Express any optimum broadcast tree in a *normal* form which *respects* the candidate solution *br*
  - Show all trees in this form are 'the same'
    - Inform internal nodes in the same rounds
  - Show any subtree generated by the algorithm is 'the same' as normal trees
    - Inform internal nodes in the same rounds as normal forms

Proof of correctness

- A normal tree:
  - Non-lazy
  - Two internal nodes u, v: $\omega(u) \leq \omega(v) \Leftrightarrow t(u) \leq t(v)$
  - Internal node u, leaf x: $\omega(u) \leq \omega(x)$



Proof of correctness

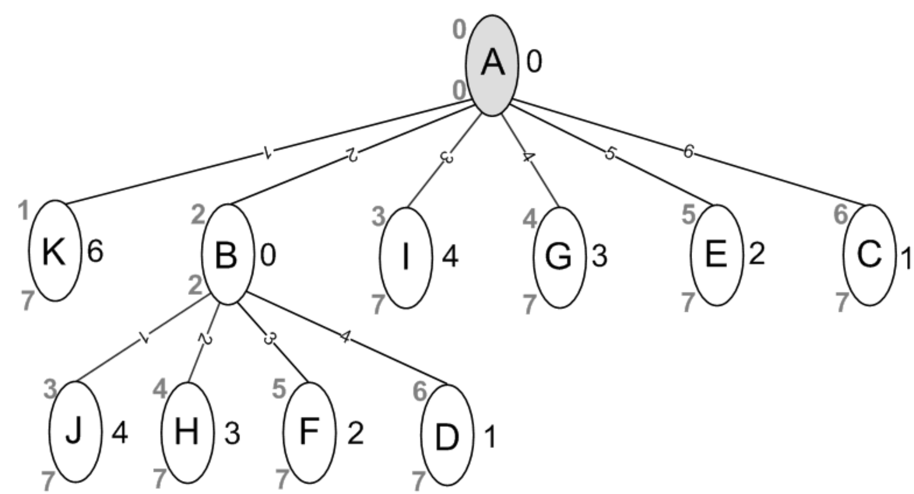
- A broadcast tree respects a candidate time *br* if for internal node *u* and leaf *x*, $t(x) < t(u)$:

$$t(u) + \omega(x) > br \qquad \text{AND}$$

$$t(u) + \omega(u) + 1 + \omega(x) > br$$

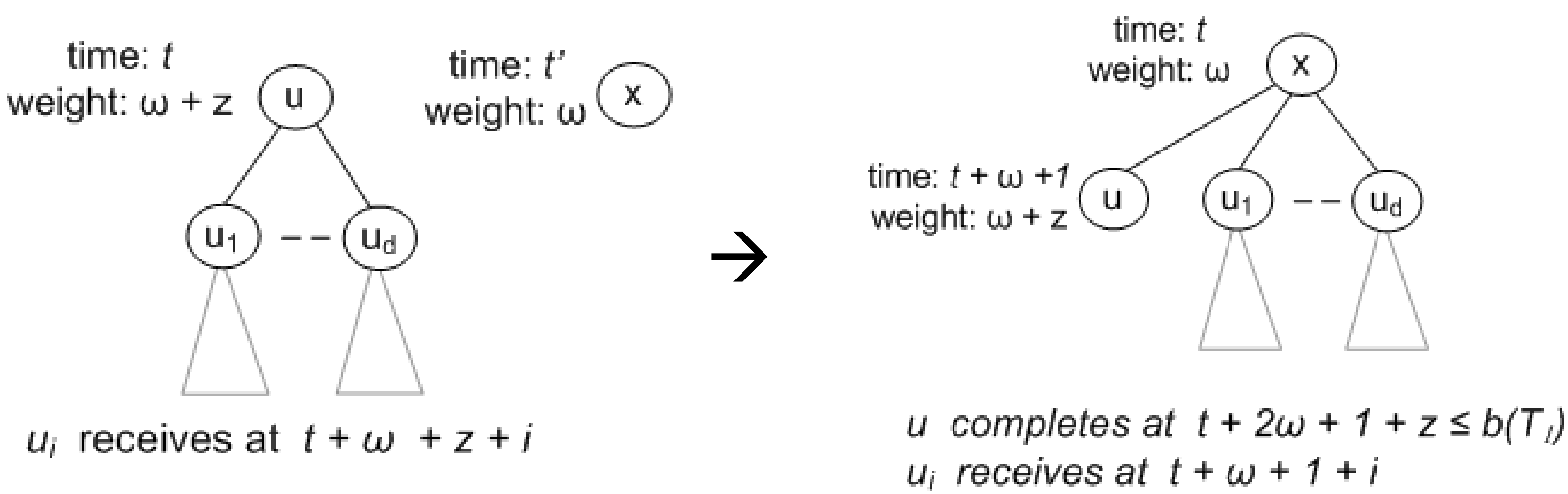
- Informing *x* can not be postponed with respect to *br*

Respects br=7, but not br = 8 →



Proof of correctness

- A broadcast tree with time *br* → A normal broadcast tree which respects *br*
- Proof: play with the links
  - Ex: Internal node *u*, leaf *x*: $\omega(u) \leq \omega(x)$



Proof of correctness

- Claim: BadGuess → No tree with broadcast time *br*
- A broadcast tree with time *br* → A normal broadcast tree which respects *br*
- In all normal trees which respect a candidate time *br*, internal nodes receive at the same time (proof omitted)
- In the normal sub-tree generated by the algorithm, internal nodes receive at the same time as these trees (details omitted)

Conclusion

- There is a $O(n^2 \log(n+W))$ algorithm for optimum broadcasting in complete weighted-vertex graphs
  - Polynomial even if $W=O(2^n)$

Thank you