Data Flow, Random Placement

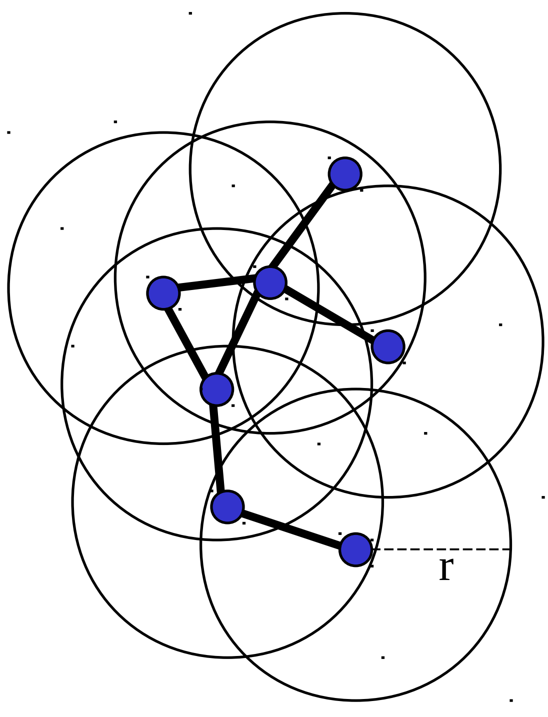
Unit Disk Graph

- Motivation:
  - The strength of the received signal decreases proportionally to $d^{-\gamma}$, where
    - d is the distance d from the sender
    - γ is the a *path loss exponent*
  - Connections only exist if the signal/noise ratio is beyond a threshold
- Definition
  - Given a finite point set V in $\mathbf{R}^2$ or $\mathbf{R}^3$,
  - a **Unit Disk Graph (UDG)** G=(V,E) with radius r of the point set is defined by the undirected edge set:

$$E = \{\{u, v\} \;\mid\; ||u, v||_2 \leq r\}$$

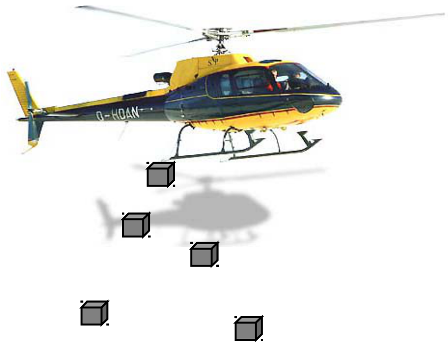
  - where $||u,v||_2$ is the Euclidean distance:

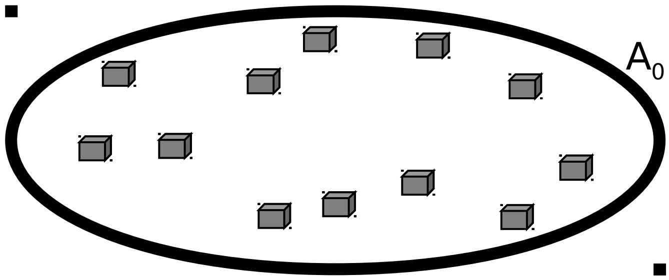
$$||u, v||_2 = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2}$$



Random Placement Model

- Motivation
  - Throwing nodes from a plane
  - Natural processes lead to a random placement
- Definition
  - A set of points is placed randomly in an area A_0 if every position occurs with equal probability, i.e.
  - the probability density function (pdf) f(x) is a constant

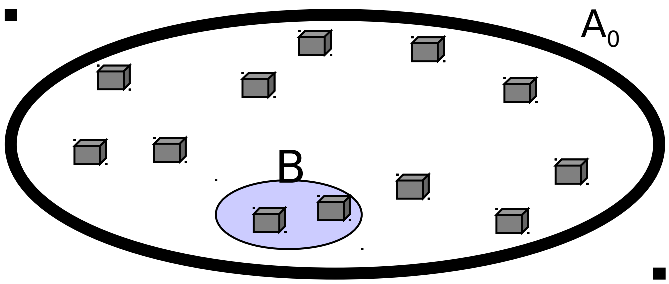




Properties of Random Placement

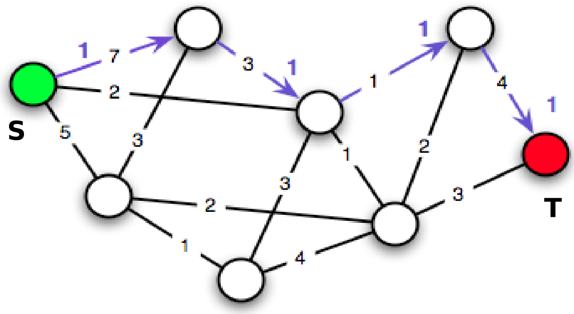
- The probability that a node falls in a specific area B of the overall area A_0 is:
  - Pr[a node falls into B] = $|B| / |A_0|$,
  - where $|B|$ is the area of B
- Lemma
  - Let $p = |B|/|A_0|$.
    Then be the probability that k of n nodes fall in an area B is:

    Pr[k of n nodes fall into B] =
    $$\binom{n}{k} p^k (1-p)^{n-k}$$



Data Flow in Networks

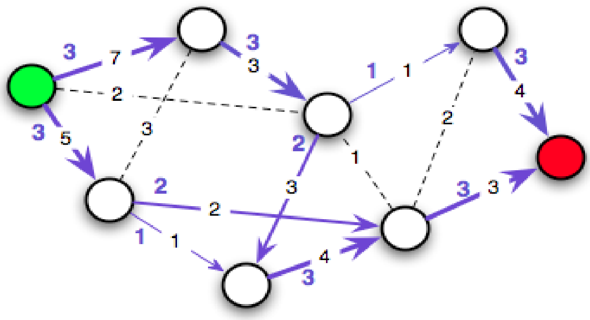
- Motivation:
  - Optimize data flow from source to target
  - Avoid bottlenecks
- Definition:
  - **(Single-commodity) Max flow problem**
  - Given
    - a graph G=(V,E)
    - a capacity function w: E $\longrightarrow$ $\mathbf{R}^{+}_{0}$,
    - source set S and target set T
  - Find a maximum flow from S to T



- A flow is a function f : E $\longrightarrow$ $\mathbf{R}^{+}_{0}$ with
  - for all e $\in$ E: f(e) $\leq$ w(e)
  - all u,v $\in$ V: f(u,v)≥0
  - $\forall u \in V \setminus (S \cup T) \quad \sum_{v \in V} f(v,u) = \sum_{v \in V} f(u,v)$
- The size of the flow is: $\sum_{u \in S} \sum_{v \in V} f(u,v)$

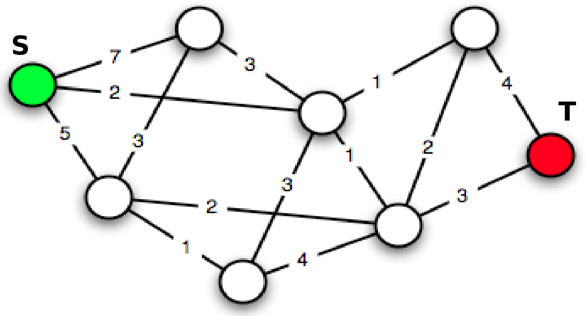
Finding the Max Flow

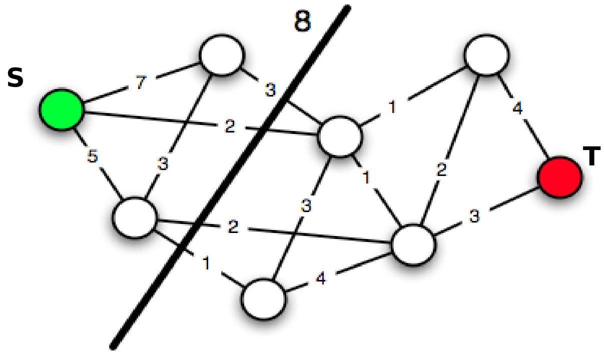
- In every natural pipe system the maximum flow is computed by nature
- Computer Algorithms for finding the max flow:
  - Linear Programming
    - The flow equalities are the constraints of a linear optimization problem
    - Use Simplex (or ellipsoid or interior point method) for solving this linear equation system
  - Ford-Fulkerson
    - As long there is an open path (a path which improves the flow) increase the flow on this path
  - Edmonds-Karp
    - Special case Ford-Fulkerson
    - Use Breadth-First-Search to find the paths



Min Cut in Networks

- Motivation:
  - Find the bottleneck in a network
- Definition:
  - **Min cut problem**
  - Given
    - a graph G=(V,E)
    - a capacity function w: E $\rightarrow \mathbf{R}^{+}_{0}$,
    - source set S and target set T
  - Find a minimum cut between S and T
- A cut C is a set of edges such that
  - there is no path from any node in S to any node in T
- The size of a cut C is: $\sum_{e \in C} w(e)$



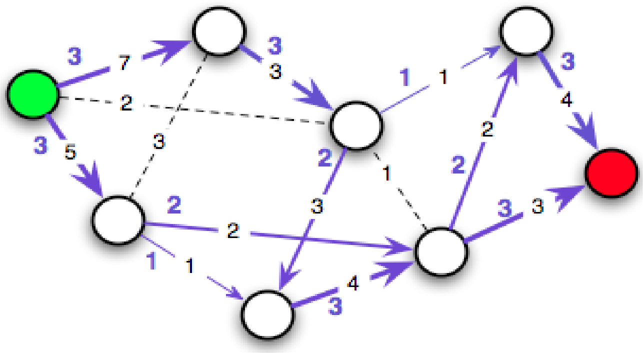


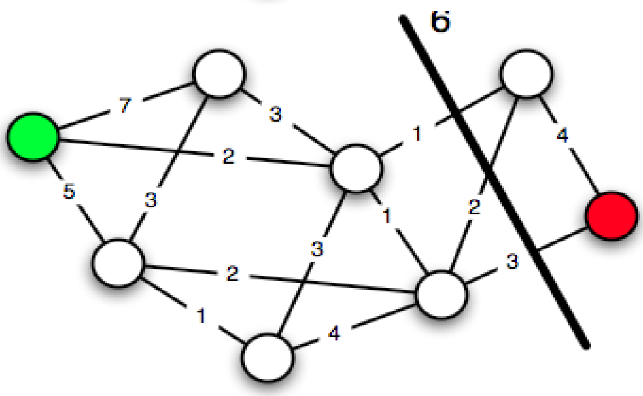
Min-Cut-Max-Flow Theorem

- **Theorem**

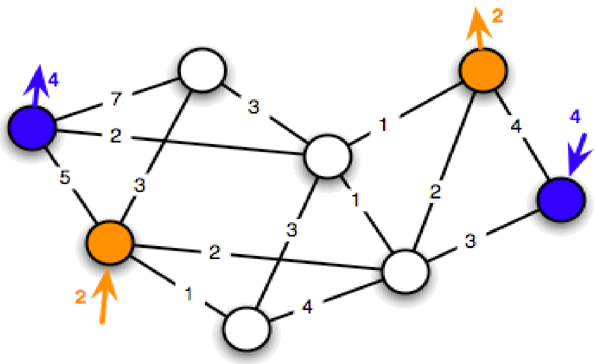
  For all graphs, all capacity functions, all sets of sources and sets of targets

  the minimum cut equals the maximum flow.



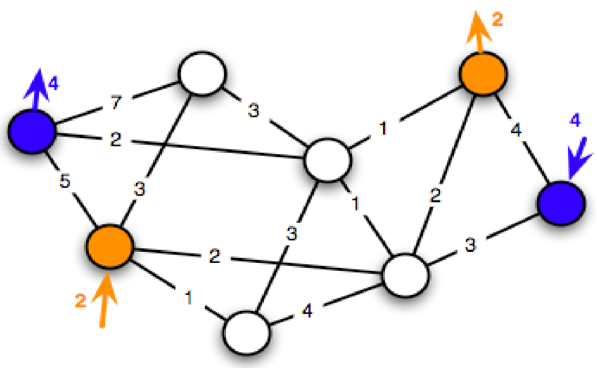


Multi-Commodity Flow Problem

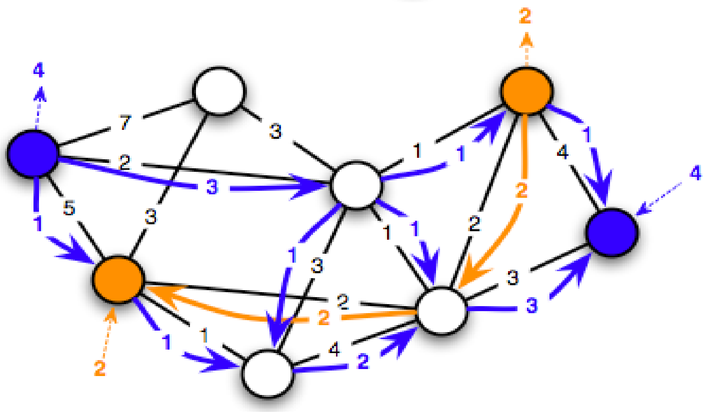


- Motivation:
  - Theoretical model of all communication optimization for point-to-point communication with capacities
- Definition
  - **Multi-commodity flow problem**
  - Given
    - a graph G=(V,E)
    - a capacity function w: E → $\mathbf{R}^{+}_{0}$,
    - commodities K_1, .., K_k:
      - $K_i=(s_i,t_i,d_i)$ with
      - s_i is the source node
      - t_i is the target node
      - d_i is the demand
- Find flows $f_1,f_2,\ldots,f_k$ for all commodities obeying
  - Capacity: $$\sum_{i=1}^{k} f_i(u,v) \le w(u,v)$$
  - Flow property: $$\forall v \notin \{s_i,t_i\} \ : \ \sum_{u\in V} f_i(u,v) = \sum_{u \in V} f_i(v,u)$$
  - Demand $$\sum_{v\in V} f_i(s_i,v) = \sum_{u\in V} f_i(u,t_i) = d_i$$

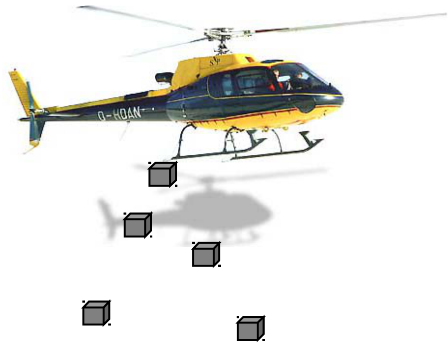
Solving Multi-Commodity Flow Problems



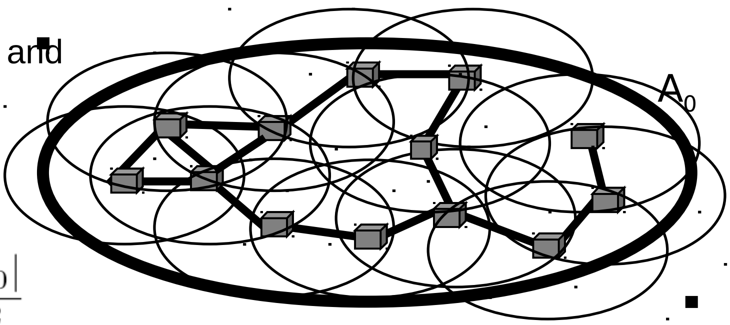
- The Multi-Commodity Flow Problem can be solved by linear programming
  - Use equality as constraints
  - Use Simplex or Ellipsoid Algorithm
- There exist weakened versions of min-cut-max-flow theorems



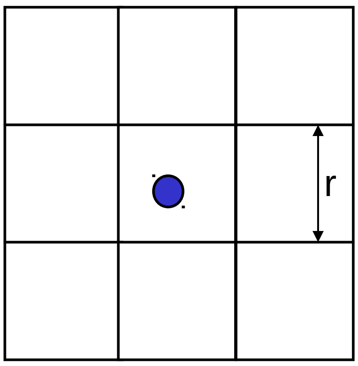
Minimum Density for Connectivity



- Gupta, Kumar: **The Capacity of Wireless Networks,** *IEEE Transactions on Information Theory*, 46(2), 388-404, 2000.
  - Critical Power for Asymptotic Connectivity in Wireless Networks
- Motivation:
  - How many nodes need to be placed to achieve a connected UDG (unit-disk graph)
- **Theorem**
  - In the square area A_0 it is necessary and sufficient to place n nodes uniformly at random a connected UDG, where
    $$c \cdot \pi r^2 \cdot n = |A_0| \log n$$
  - for some constant c.
- Equivalently: $$\Theta\left(\frac{n}{\log n}\right) = \frac{|A_0|}{r^2}$$

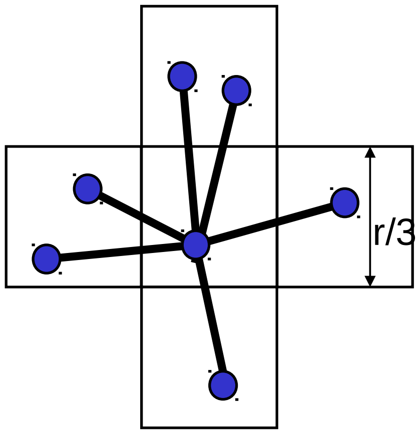


Why so Many Nodes are Necessary?



- Sufficient condition for unconnectedness:
  - at least one node in a square of side length r
  - 8 neighbored squares are empty
- Probability, that none of the n nodes fall in surrounding squares:
  $$\left(1 - \frac{8r^2}{|A_0|}\right)^n$$
- for x ∈ [0,0.75]:
  $$e^{-2x} \le (1-x) \le e^{-x}$$
- Thus (for sufficiently large A_0)
  $$\left(1 - \frac{8r^2}{|A_0|}\right)^n \ge e^{-\frac{16 r^2 n}{|A_0|}}$$
- The expected value of such isolated nodes is at least
  $$n \cdot e^{-\frac{16 r^2 n}{|A_0|}}$$
- If $r^2 = o\left(\frac{|A_0| \ln n}{n}\right)$, then the expected number of isolated nodes is at least 1

Are so Many Nodes Enough?



- Sufficient condition of connectivity
  - In the adjacent squares of side length r/3 is at least one node
- Probability that at least one node is in such a square:

$$1 - \left(1 - \frac{r^2}{9|A_0|}\right)^n$$

- Choose

$$r^2 = c \cdot \frac{|A_0| \ln n}{n}$$

- Then this probability is:

$$1 - \left(1 - \frac{c \ln n}{9n}\right)^n \geq 1 - e^{-\frac{c}{9} \ln n} = 1 - n^{-\frac{c}{9}}$$

- Choose c>9
  - then the probability of such an occupied neighbored square is $o(n^{-1})$
  - Multiplying this probability with 4n (for all neighbored squares) gives an upper bound on the probability that each node does not have neighbors to the four sides
  - Then this probability is o(1).

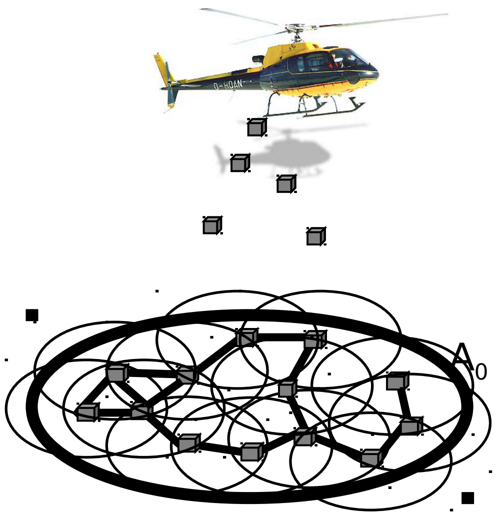
Network Flow in Random Unit Disk Graphs

- Motivation:
  - What is the communication capacity of the network
- **Theorem**
  - Assume that if n nodes are uniformly random placed in the square area A_0, where

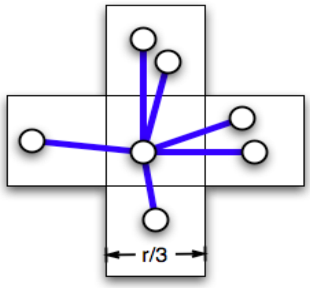
$$\Theta\left(\frac{n}{\log n}\right) = \frac{|A_0|}{r^2}$$

  - Assume that each node is able to transmit data to a neighbor in the UDG. Assume that each node chooses a target uniformly at random and send data to the target. The data rate, which can be achieved at all nodes is:

$$\Theta\left(\frac{W}{\sqrt{n \log n}}\right)$$



Proof Sketch



- 1. observation:
  - if $$\Theta\left(\frac{n}{\log n}\right) = \frac{|A_0|}{r^2}$$
  - the random placement leads to a grid-like structure, in which the side length of the grid cells is r/3.
- 2. observation:
  - The network is mainly a grid of m x m cells, where $$m = \Theta\left(\sqrt{\frac{n}{\log n}}\right)$$
  - On the avarage, each cell contains log n nodes and each node has log n edges to nodes in a neighboring cell
- In a grid such a demand can be routed with capacity n^2/m (horizontal or vertical cut is bottleneck)
- In this network the minimum cut is m log n = $(n \log n)^{1/2}$
- The multicommodity flow is therefore $W/(n \log n)^{1/2}$

Discussion

- For randomly placed nodes in a square A $\Omega(n \log n)$ nodes are necessary
  - to obtain a connected UDG,
  - where $n = |A|/r^2$.
- Then the network behaves like a grid
  - up to some polylogarithmic factor.
- The bottleneck of grids is the width
  - in the optimal case of square-like formations this is $n^{1/2}$.
- If the overhead of a factor $\Omega(\log n)$ is not achieved,
  - then the UDG of randomly placed nodes is not connected.

Literature

- Piyush Gupta, P. R. Kumar: The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2), 388-404, 2000.