# Discrete Event Simulation

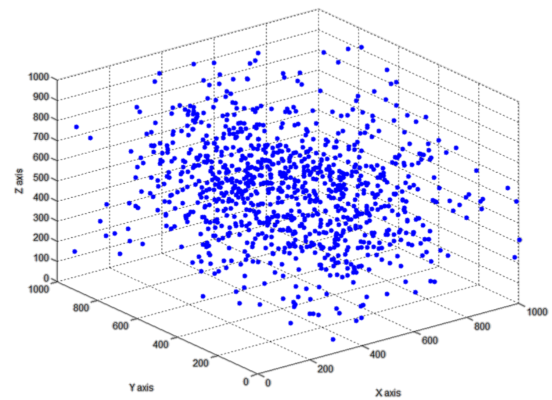
# IN2045

Dr. Alexander Klein
Stephan Günther
Prof. Dr.-Ing. Georg Carle

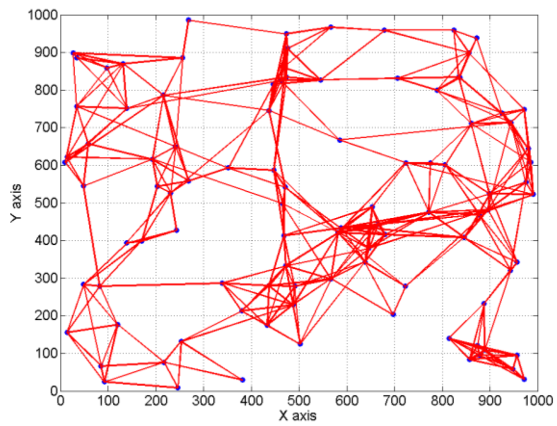
**Chair for Network Architectures and Services**
**Department of Computer Science**
**Technische Universität München**
**http://www.net.in.tum.de**

# Topics

- Point Fields
  - Generation of Point Fields
    - Constant / Variable Number of Points
    - Rectangle / Arbitrary Area
  - Homogeneous Point Fields
  - Inhomogeneous Point Fields
  - Poisson Field
  - Matern Cluster Field

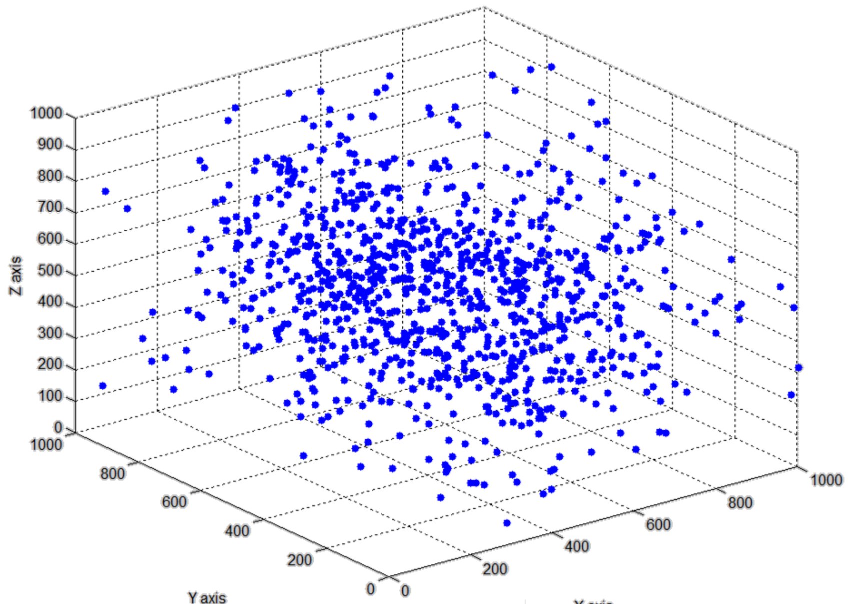


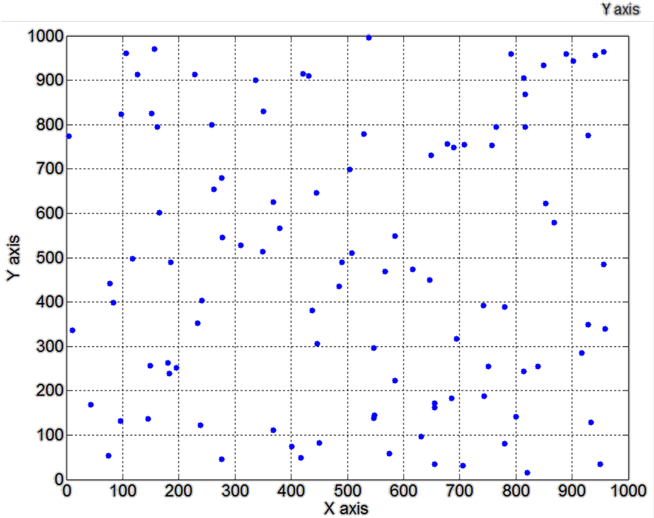

- Random Graphs
  - Generation of Random Graphs
    - Probabilistic Model
    - Waxman Model
  - Implementation Issues:
    - Adjacency Matrix/List
    - Incidence Matrix
  - Scale-free Graphs

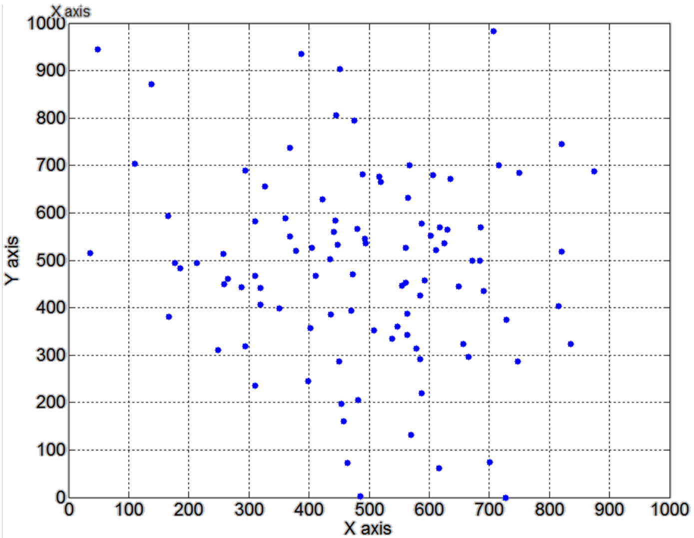


# Point Fields

- **Point fields:**



# Point Fields

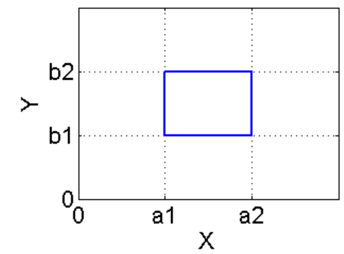
- **Point field:**
  - Two dimensional random process
  - Spatial distribution of objects in two dimensional space
    - Seismology (epicenters of earthquakes)
    - Plant ecology (position of trees or other plants)
    - Epidemiology (home locations of infected people)
    - Zoology (burrows or nests of animals)
    - Astronomy (location of stars)
    - **Telecommunication (spatial distribution of mobile users)**

  - **The development of many system parameters is influenced by the spatial distribution of the simulated objects**

# Point Fields

- **Point fields with a constant number of points (rectangle):**
  - Task:

    Generate a homogeneous point field with n points in a rectangle which is given by $(a_1,b_1),(a_1,b_2),(a_2,b_1),(a_2,b_2)$




  - Algorithm:
    - for i = 1:n
      - Generate random variable z1 ~ U(0,1)
      - Generate random variable z2 ~ U(0,1)
      - Point $(x,y) = (a_1 + z_1 \cdot (a_2 - a_1),\ b_1 + z_2 \cdot (b_2 - b_1))$

    end

# Point Fields

- **Point fields with a variable number of points (rectangle):**
  - F - size of the rectangle
  - $E[X]$ - average number of points in F
  - $\lambda = \frac{E[X]}{F}$ - intensity of the point field
  - X – discrete random variable which describes the number of points in the rectangle
  - Generate a homogeneous point field with X points

# Point Fields

## Binomial – Point Field

- Binomial distributed number of points

$$P(X = i) = \binom{n}{i} p^i (1 - p)^{n-1} \ , \quad p = \frac{E[X]}{n}$$

- Upper bound: n

## Poisson – Point Field

- Poisson distributed number of points

$$P(X = i) = \frac{(\lambda F)^i}{i!} e^{-\lambda F} \ , \quad \lambda = \frac{E[X]}{F}$$

- Upper bound: no upper bound !!!
- Generation: c.f. point fields with variable number of points

# Point Fields

- **Poisson – Point Field**
  - Optimized Generation:

1. Generate x-coordinates by using a one dimensional Poisson process

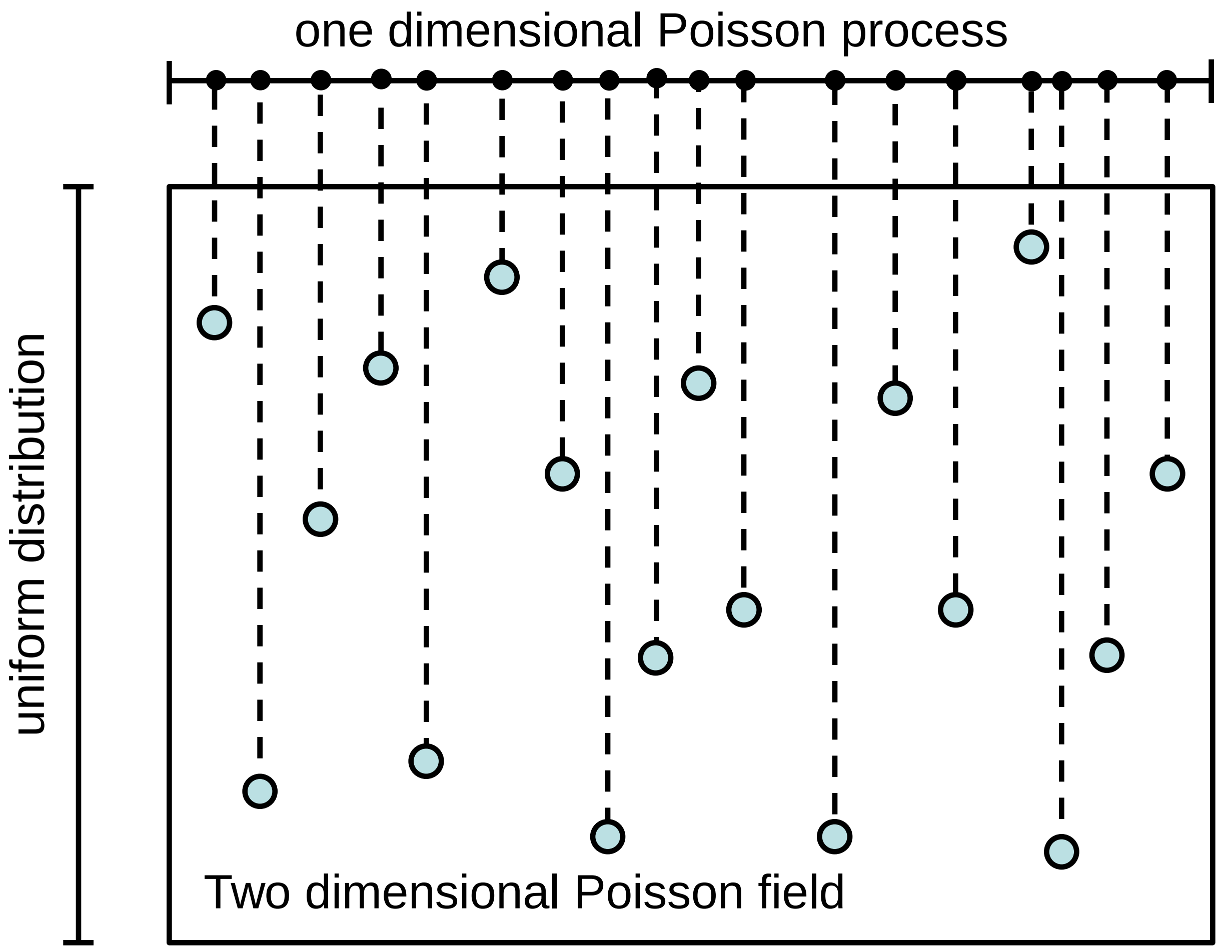
   in ($a_1$, $a_2$) with rate $\lambda^* = \dfrac{E[X]}{a_2 - a_1}$

   Note that the one dimensional process specifies the number of points in the point field.

2. Generate the y-coordinates according to a uniform distribution in the interval ($b_1$, $b_2$)

# Point Fields

- **Poisson – Point Field**
  - Optimized Generation:

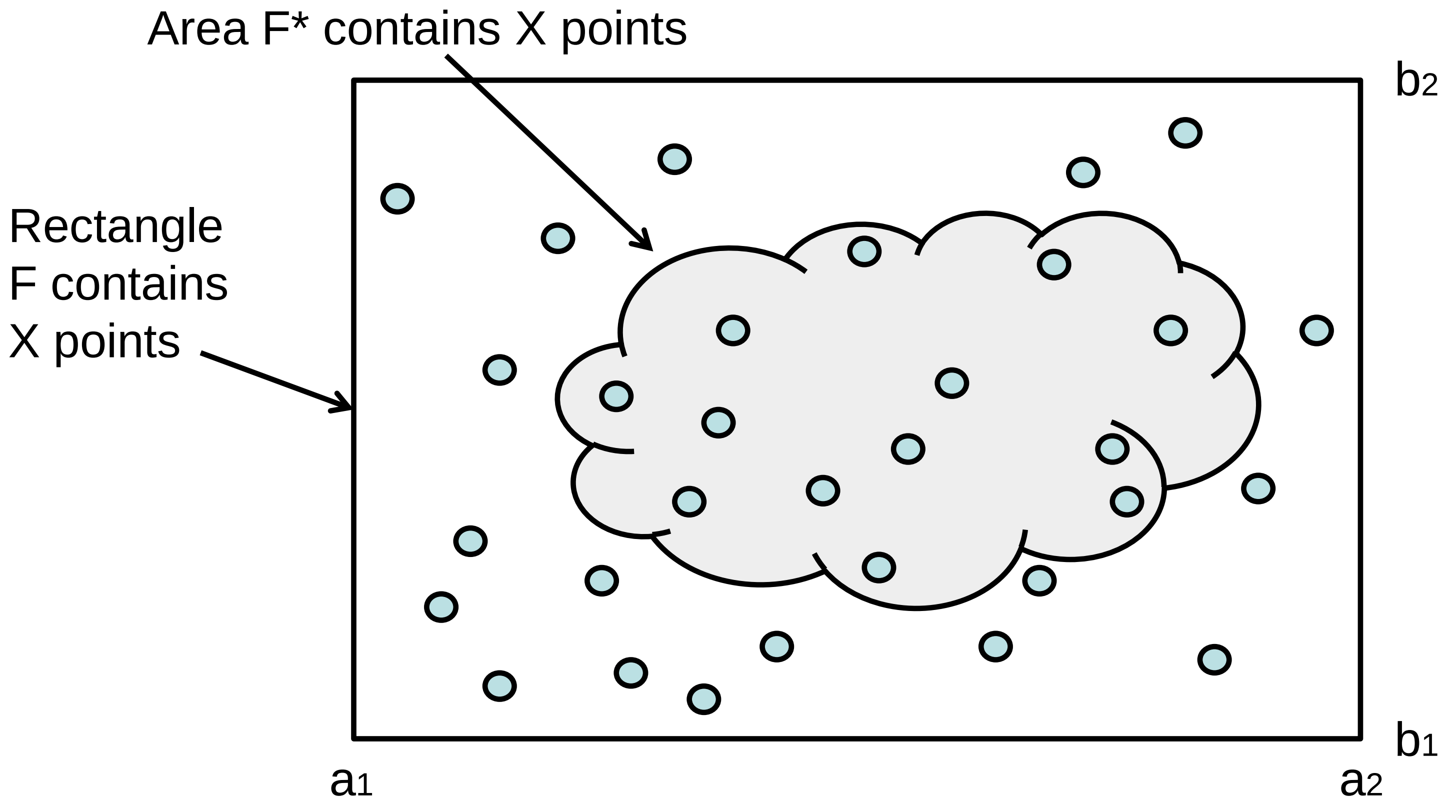


# Point Fields

- **Point fields with a variable number of points in an arbitrary area:**
  - Problem:
    - Area has arbitrary shape and size F*
    - Average number of points in F* = E[X*]
    - Point intensity in F*: $\lambda^* = \dfrac{E[X^*]}{F^*}$
    - Previously introduced algorithms only work for rectangles
  - Solution:
    - Generate a rectangle such that the arbitrary area F* fits in the rectangle
    - Generate points in the rectangle which includes the area F* until the desired number of points are in the area F*

# Point Fields

- **Point fields with a variable number of points in an arbitrary area:**
  - Optimized Generation:




- Generation similar to Accept-Reject method:
- No additional random number required
- Efficiency of the algorithm is given by F*/F

# Point Fields

## Inhomogeneous point fields

- Characteristics
  - Point intensity $\lambda(x, y)$ depends on the position
  - Maximum $\lambda_{\max} = \max_{(x,y)}(\lambda(x, y))$
  - Point intensity in F*: $\lambda^* = \dfrac{E[X^*]}{F^*}$
- Generation:
  - Calculate E[x] by integrating $\lambda(x, y)$ over F
  - Choose RV X according to E[x]
  - Repeat the following three steps until X points are generated
    1. Generate a point (x, y) in F
    2. Choose a random number z ⟹ RV Z ~ U(0,1)
    3. Accept if $z \le \dfrac{\lambda(x, y)}{\lambda_{\max}}$, otherwise reject

## Inhomogeneous / homogeneous point fields

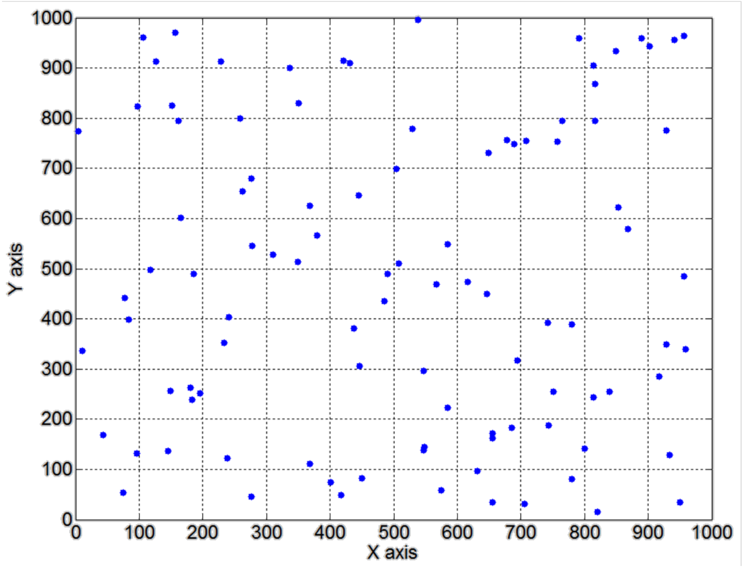
- Impact of the chosen distribution on the point field

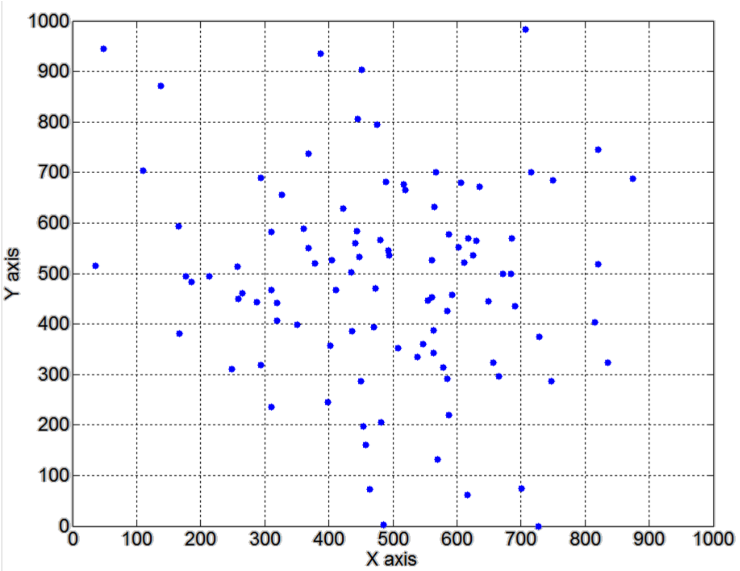
Example 1:

- x – axis – uniform distributed
- y – axis – uniform distributed

Example 2:

- x – axis – normal distributed
- y – axis – normal distributed



# Point Fields

- **Cluster point fields**
  - Idea:
    - Generate a point field with low density where each point represents a parent point
    - Generate a homogeneous Poisson field with intensity $\lambda_e$ around parent points which represent the centre of the clusters
  - Matern cluster field:
    - Create a homogeneous Poisson field around each parent point with radius R
    - Average number of points in each circle is given by $E[X]$
    - Intensity in each field around a parent point $\lambda = \frac{E[X]}{F} = \frac{E[X]}{\pi R^2}$

# Point Fields

## Cluster point fields

- Matern cluster field:

  A Matern cluster field can be generated in different ways

  - 1. Generation: Accept-Reject method

  ⇨ inefficient due to the high number of circle shaped Poisson field

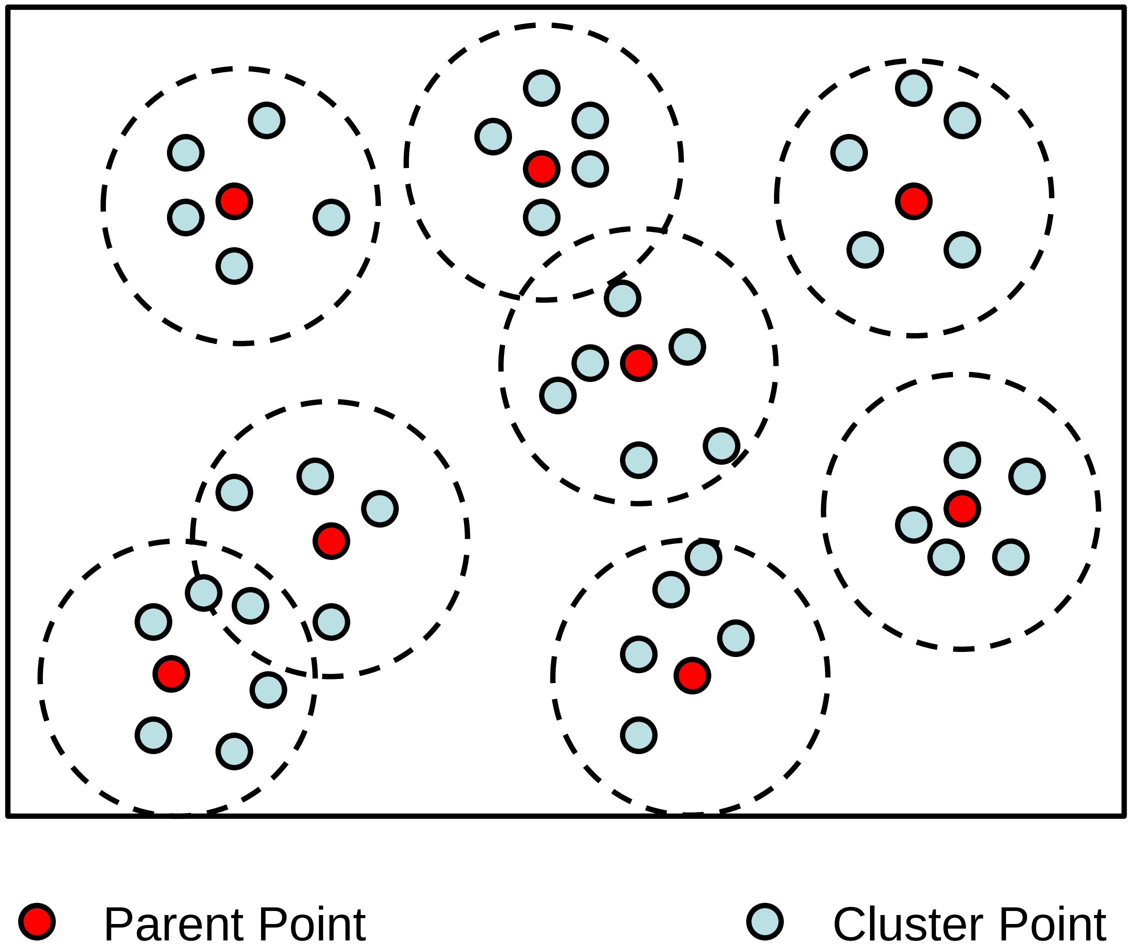
  - 2. Generation: Usage of polar coordinates $(\varphi, r)$
    - Generate uniform distributed coordinate $\varphi \in [0, 2\pi($
    - Generate distance $r \in [0, R($ according to the following density function

$$f(r) = \begin{cases} \frac{2r}{R^2} & r \leq R \\ 0 & \text{sonst} \end{cases}$$

- Uniform distribution of r results in a decrease of the intensity towards the border of the circle
- The rectangle for parent point generation has to be smaller than the Matern cluster field in order to mitigate border effects

# Point Fields

- Matern cluster field




Example: Each parent has 5 points in his cluster

# Random Graphs

# Random Graphs

## Random Graphs

A graph is an abstract representation of a set of objects where pairs of objects can be connected by links.

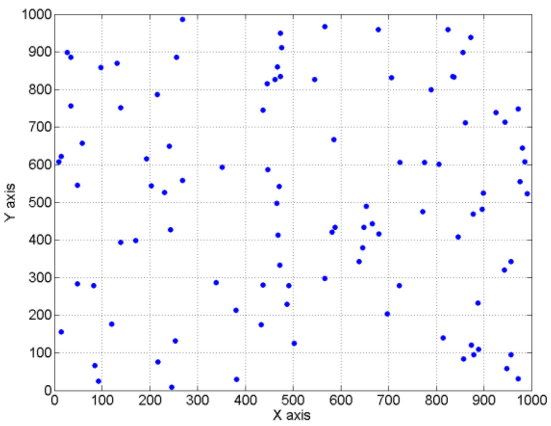
- Graph $G = (V, E)$
- V: Vertices/Nodes = Router
- E: Edges = Links
  - $e = \{u, v\} \in V \times V$
  - Undirected ⇒ bidirectional
  - Directed ⇒ unidirectional
- Node degree $\delta(v)$, $v \in V$ Number of edges that are connected with v
- Average node degree: $\delta^* = 2 \cdot |E| / |V|$
- In-degree $\delta^-(v)$: number of edges that point to node v
- Out-degree $\delta^+(v)$: number of edges that point away from node v
- Distance $d_G(u, v)$: shortest path between two vertices in the graph
- Network diameter: longest path between two vertices in the graph
- K-(edge/vertex)-connected: A graph is called k-connected if at least k edges have to be removed in order to partition the graph

# Random Graphs

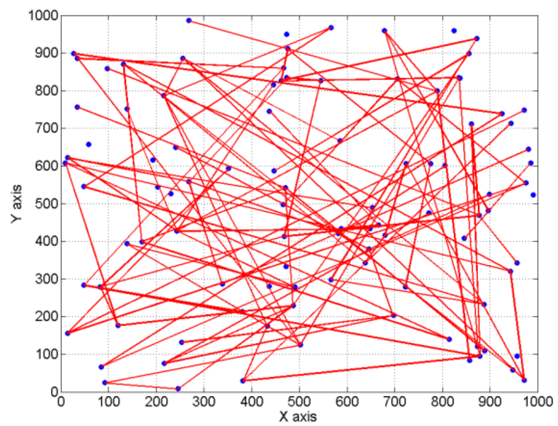
- **Random Graphs with predefined characteristics**
  - Generate a predefined number of nodes in a plane (point field)
  - Connect the nodes in the network by applying one of the following models

    1. Basic model(1/2):
       – Generation:
         Generate an edge between two nodes with probability p
       – Advantage:
         » Fast and simple
       – Disadvantage:
         » Number of links per node varies
         » Average node degree only depends on the number of nodes
         » Connectivity between two nodes does not depend on the distance between them
         » Does not guarantee full connectivity of the network
         » Does not fit for large networks

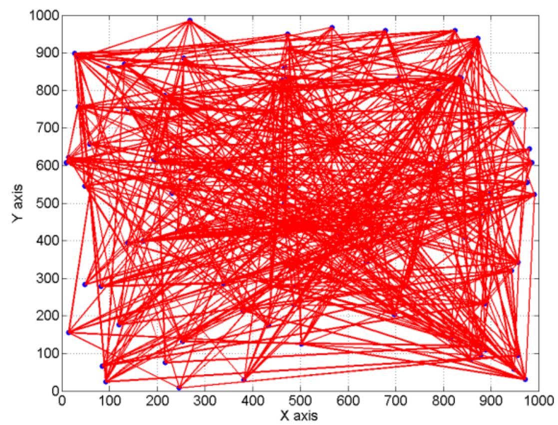
- **Random Graphs with predefined characteristics**
  1. Basic model(2/2):




2D Plane – 100 nodes




Basic Model: p=0.01




Basic Model: p=0.05

# Random Graphs

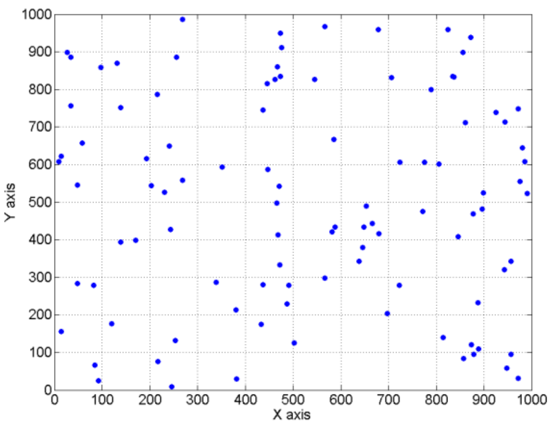
## ❑ Random Graphs with predefined characteristics

2. Waxman model:
   - Connectivity between two nodes becomes more likely the shorter the distance between them
   - Probability that two nodes are connected is given by

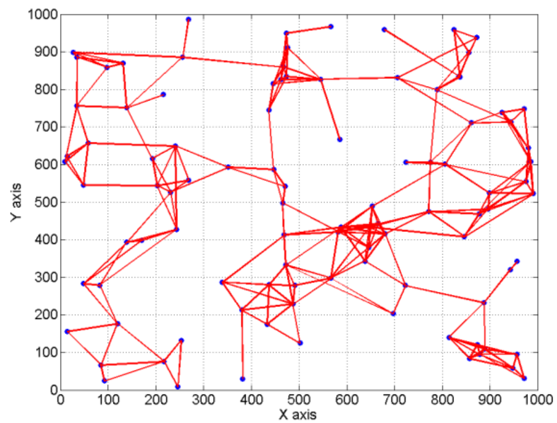
$$P(u,v) = \alpha \cdot e^{\frac{-d}{\beta \cdot L}} \quad \text{with} \quad \alpha > 0, \beta \le 1$$

     » D: Euclidean distance between the two nodes

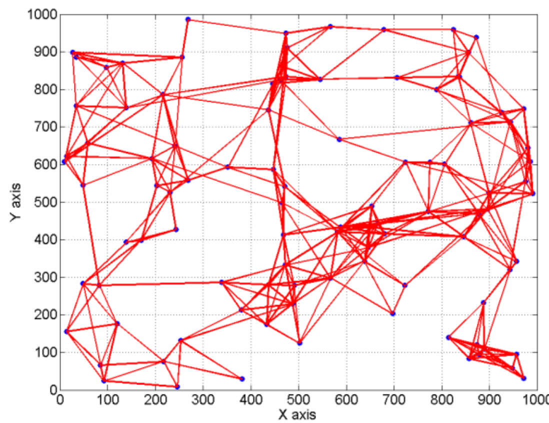
     » L: The maximum distance between two nodes




2D Plane – 100 nodes




Waxman:
α=10,β=0.025,L=1400




Waxman:
α=10,β=0.030,L=1400

## Random Graphs with predefined characteristics

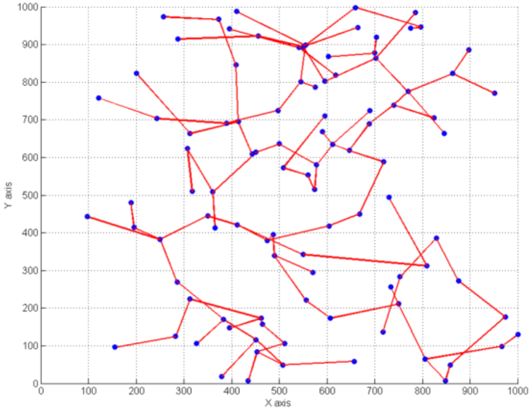
3. Node degree model:

   Problem: Generate a random graph where nodes have at least a minimum degree but less than a maximum degree

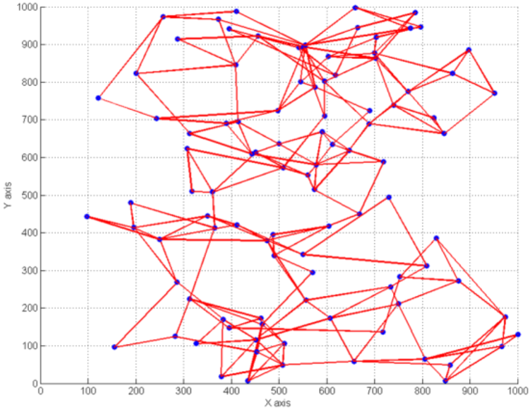
   – These graphs are usually generated in an iterative way by adding

   $$|E| = \frac{\delta^* \cdot |V|}{2}$$ edges

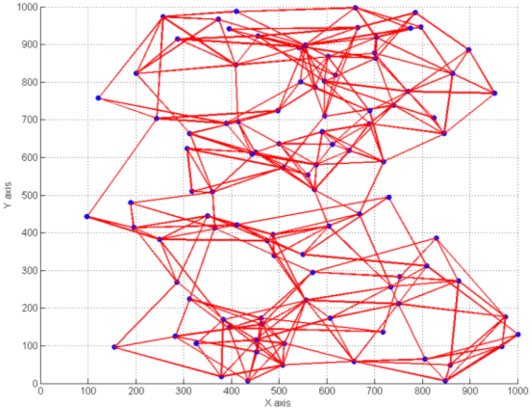
   – k-connected topologies are often used to make the network more resilient against node failures



1-connected graph



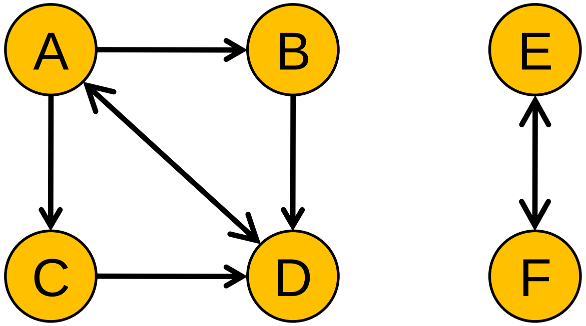
2-connected graph



3-connected graph

# Random Graphs

## Implementation of a graph:




- **Basic operations:**
  - add / remove (Edge e / Vertex v)
  - find (Edge e / Vertex v)
  - getVertices (Graph g)
  - getEdges (Graph g)
- **Complex operations:**
  - getDegree (Vertex v) - Degree of node v
  - isReachable (Vertex src, Vertex dst) - True if a path exists from src to dst, false otherwise.
  - shortestPath (Vertex src, Vertex dst)
  - isComplete (Graph g) - True if all nodes have max degree.
  - isConnected (Graph g) - True if isReachable returns true for all nodes.
  - totalWeight (Graph g) - Sum of all weights.
  - getOneHopNeighbors (Graph g, Vertex v) - List of all direct neighbors.

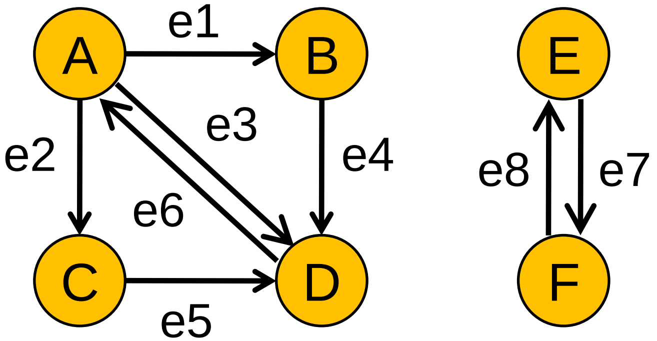
# Random Graphs

- **Implementation of a graph:**
  - **Matrix structures:**
    - **Adjacency matrix:**

      Is an n by n matrix A where n is the number of vertices |V| in the graph. Two vertices i and j are connected with an edge pointing from vertex I to vertex j if the element $a_{i,j}$ is 1, otherwise 0.

Example:

| Src Vertex / Dst Vertex | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | 0 | 1 | 1 | 1 | 0 | 0 |
| B | 0 | 0 | 0 | 1 | 0 | 0 |
| C | 0 | 0 | 0 | 1 | 0 | 0 |
| D | 1 | 0 | 0 | 0 | 0 | 0 |
| E | 0 | 0 | 0 | 0 | 0 | 1 |
| F | 0 | 0 | 0 | 0 | 1 | 0 |



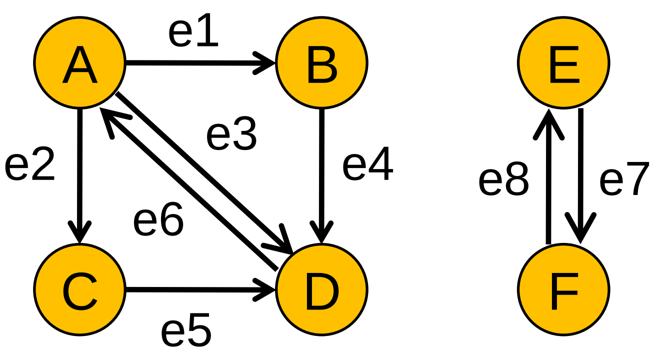
# Random Graphs

- **Implementation of a graph:**
  - **Matrix structures:**
    - **Adjacency matrix:**

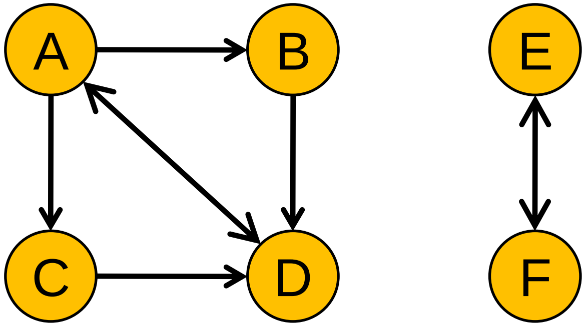
      Is an n by n matrix A where n is the number of vertices |V| in the graph. Two vertices i and j are connected with an edge pointing from vertex I to vertex j if the element $a_{i,j}$ is 1, otherwise 0.

Example:

| Src Vertex / Dst Vertex | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| **A** | 0 | 1 | 1 | 1 | 0 | 0 |
| **B** | 0 | 0 | 0 | **Outgoing** | | |
| **C** | 0 | 0 | 0 | 1 | 0 | 0 |
| **D** | 1 | 0 | 0 | 0 | 0 | 0 |
| **E** | **Incoming** | | | 0 | 0 | 1 |
| **F** | 0 | 0 | 0 | 0 | 1 | 0 |



- **Implementation of a graph:**
  - **Matrix structures:**
    - **Adjacency matrix:**




**Out-degree / in-degree**

| Src Vertex / Dst Vertex | A | B | C | D | E | F | Out-degree |
|---|---|---|---|---|---|---|---|
| **A** | 0 | 1 | 1 | 1 | 0 | 0 | **3** |
| **B** | 0 | 0 | 0 | 1 | 0 | 0 | **1** |
| **C** | 0 | 0 | 0 | 1 | 0 | 0 | **1** |
| **D** | 1 | 0 | 0 | 0 | 0 | 0 | **1** |
| **E** | 0 | 0 | 0 | 0 | 0 | 1 | **1** |
| **F** | 0 | 0 | 0 | 0 | 1 | 0 | **1** |
| **In-degree** | **1** | **1** | **1** | **3** | **1** | **1** | 0 |

Out-degree and in-degree of each vertex is represented by the sum of the corresponding row or column of the adjacency matrix

# Random Graphs

- **Implementation of a graph:**
  - **Matrix structures:**
    - **Adjacency matrix:**
      - – Characteristics:
        - » Complexity:
          - – Insert / Delete O(1)
          - – Find O(1)
          - – Find neighbors O(|V|)
        - » Memory consumption:
          - – Directed graph $O(|V|^2)$
          - – Undirected graph $O(|V|^2 / 2)$
        - » Use case:
          - – Small graphs due to simplicity and memory consumption
          - – Dense graphs due to low complexity

# Random Graphs

- **Implementation of a graph:**
  - **Matrix structures:**
    - **Incidence matrix:**

      Is a matrix B of size |V| (number of vertices) by |E| (number of edges) with entries $b_{i,j}$ which indicate whether the vertex i incidence edge j.

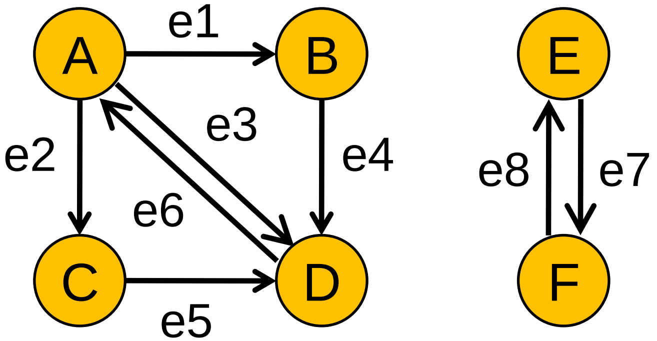
      » Edge j enters vertex i: $b_{i,j} = 1$

      » Edge j leaves vertex i: $b_{i,j} = -1$

      » No incident: 0

Example:

| Vertex / Edge | e1 | e2 | e3 | e4 | e5 | e6 | e7 | e8 | e9 |
|---|---|---|---|---|---|---|---|---|---|
| **A** | -1 | -1 | -1 | 0 | 0 | 1 | 0 | 0 | 0 |
| **B** | 1 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 |
| **C** | 0 | 1 | 0 | 0 | -1 | 0 | 0 | 0 | 0 |
| **D** | 0 | 0 | 1 | 1 | 1 | -1 | 0 | 0 | 0 |
| **E** | 0 | 0 | 0 | 0 | 0 | 0 | -1 | 1 | 0 |
| **F** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -1 | 0 |

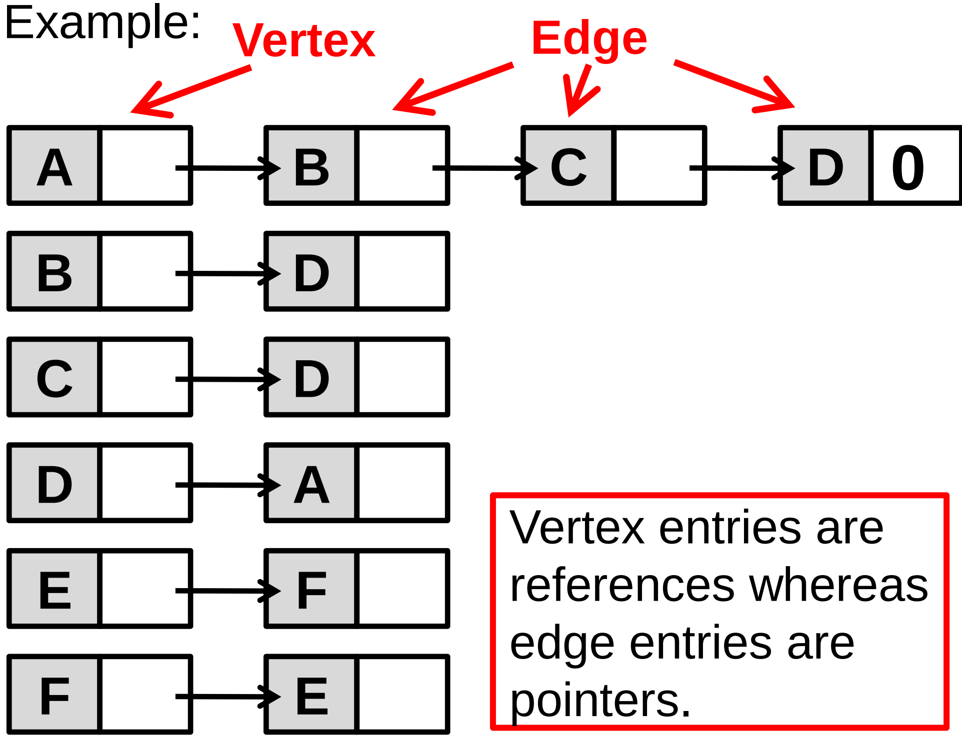


# Random Graphs

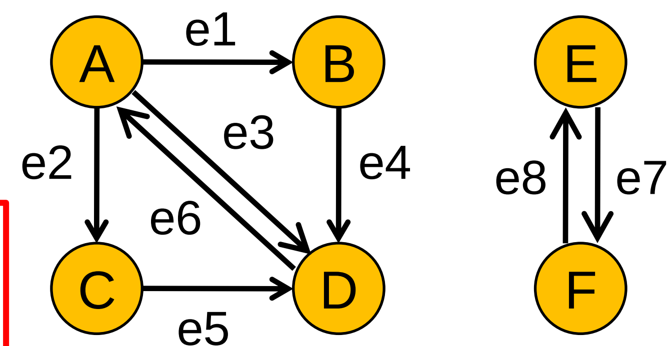
- **Implementation of a graph:**
  - **Linked structures:**
    - **Adjacency list:**

      Is an array/list of length |V| which holds for each node a list of its neighbor nodes.

Example:




Vertex entries are references whereas edge entries are pointers.



# Random Graphs

- **Implementation of a graph:**
  - **Linked structures:**
    - **Adjacency list:**
      - Characteristics:
        - » Complexity:
          - Insert O(1)
          - Delete O(|V|)
          - Find O(|V|)
          - Find neighbors O(1)
        - » Memory consumption:
          - Directed graph O(|V|+|E|)
          - Undirected graph O(|V|+2|E|)
        - » Use case:
          - Sparse graphs
          - Efficient if neighbor nodes have to be found frequently

# Random Graphs

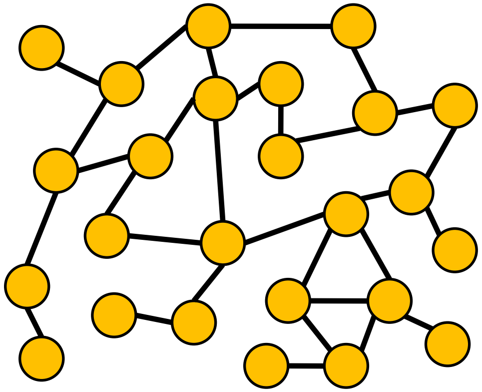
- **Special Case:**
  - **Scale-free graph:**

    A graph is called scale-free if its node degree k follows the power law.

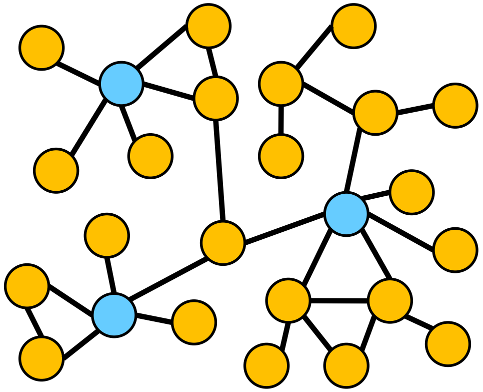
    $$P(k) = ck^{-\gamma}$$

    c and γ are constants. Typical range $0 < c < 1$, $2 < \gamma < 3$.

    - **Examples:**
      - Social networks
      - Collaboration networks
      - Computer networks
      - Disease transmission



Random Graph

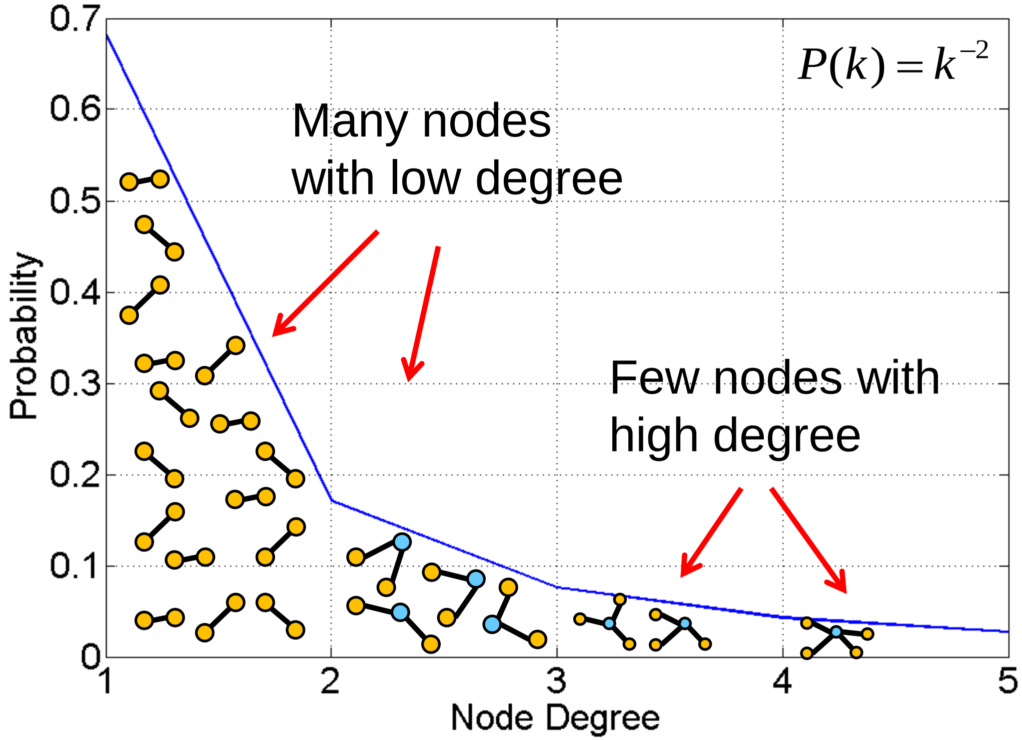


Scale-free Graph

# Random Graphs

- **Special Case:**
  - **Scale-free graph:**

    Characteristics:
    - High number of nodes with a small node degree.
    - Small number of nodes (hubs) with a high node degree.



# Random Graphs

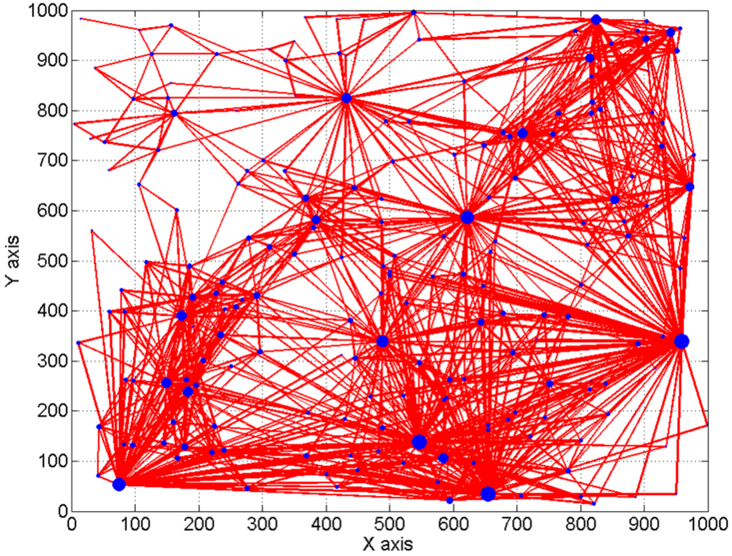
- **Special Case:**
  - **Scale-free graph:**

    A graph is called scale-free if its node degree k follows the power law.

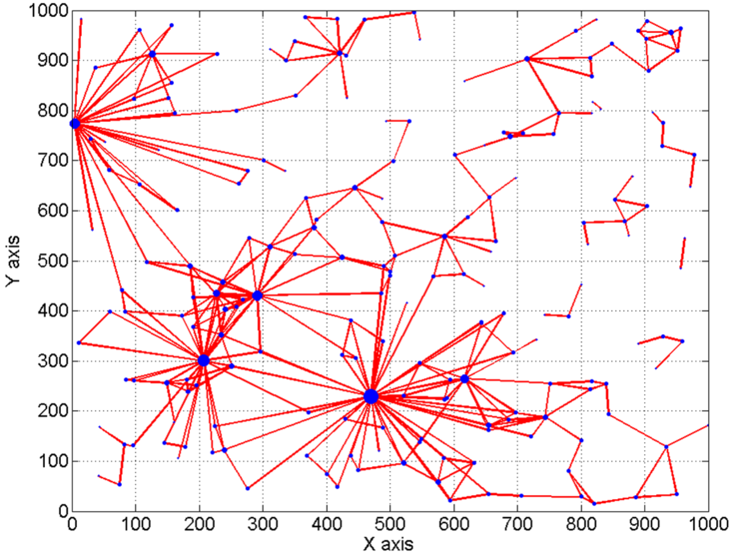
$$P(k) = ck^{-\gamma}$$

c and γ are constants. Typical range 0 < c < 1, 2 < γ < 3.

- **Examples:**




Scale-free Graph: n=200, γ=1.5




Scale-free Graph: n=200, γ=2.0

- Scale-free networks – real-world examples**:**
  - **Six degrees of separation:**
    - Small-world phenomenon:

      **Experiment by Stanley Milgram (1967)**

      - » Give letters to approx. 100 participants which should forward their letter to a specific person they do not know personally. Also the address of the person is not known.
      - » The participants where only allowed to forward the letter by hand to a person, which they think, could forward it more closer to the destination.
      - » The letters reached the destination via a maximum of 6 people.
      - » Experiment was repeated and confirmed several times with sender and receiver even being part of different ethnological groups.

⇨ Everybody on this planet is separated only by six other people.

# Random Graphs

- Scale-free networks – real-world examples:
  - **Kevin Bacon Game and the movie actor network :**
    - The Kevin Bacon Number defines the separation of movie actors away from Kevin Bacon.
    - One actor has distance 0 (Kevin Bacon himself).1902 actors have distance 1 since they played in a movie starring Kevin Beacon. 160463 actors have distance 2 since they played in movie in which someone played who played in a movie starring Kevin Bacon.

| Kevin Bacon Number | Number of Actors |
|---|---|
| 0 | 1 |
| 1 | 1902 |
| 2 | 160463 |
| 3 | 457231 |
| 4 | 111310 |
| 5 | 8168 |
| 6 | 810 |
| 7 | 81 |
| 8 | 14 |

⇨ Kevin Bacon seems to be the reasonable center of the network.

# Random Graphs

- Scale-free networks – real-world examples:
  - **Kevin Bacon Game and the movie actor network :**
    - Kevin Bacon is only the 1049th best center out of nearly 800.000 movie actors. This makes make Kevin Bacon a better center than 99% of the actors.
    - However, there are still some better centers, like Sean Connery due to his higher first and second degree.

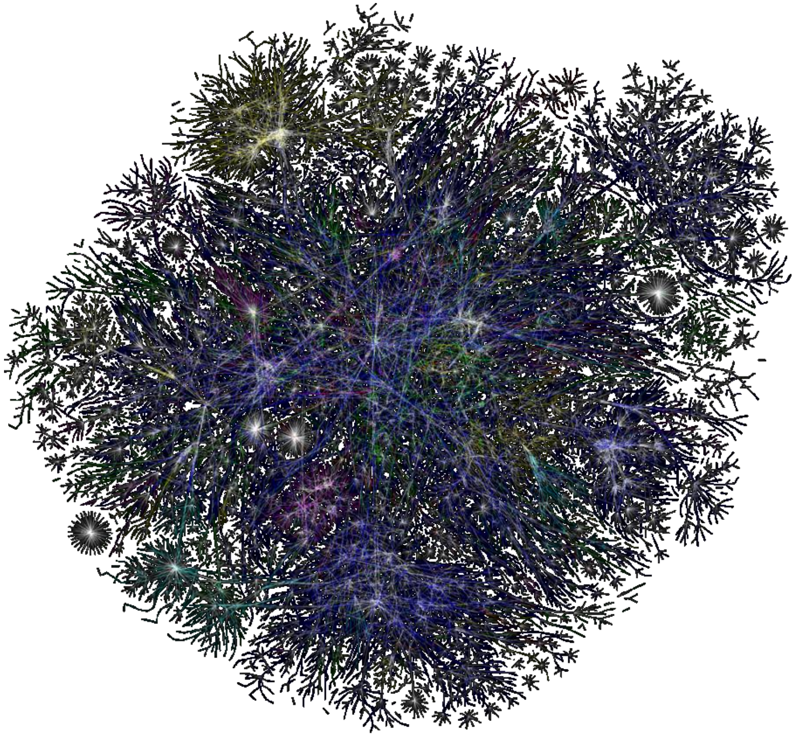
| Kevin Bacon Number | Number of Actors |
|---|---|
| 0 | 1 |
| 1 | 1902 |
| 2 | 160463 |
| 3 | 457231 |
| 4 | 111310 |
| 5 | 8168 |
| 6 | 810 |
| 7 | 81 |
| 8 | 14 |

**vs.**

| Sean Connery Number | Number of Actors |
|---|---|
| 0 | 1 |
| 1 | 2272 |
| 2 | 218560 |
| 3 | 380721 |
| 4 | 40263 |
| 5 | 3537 |
| 6 | 535 |
| 7 | 66 |
| 8 | 2 |

# Random Graphs

- Scale-free networks – real-world examples**:**
  - **The Internet:**
    - The network diameter of the Internet is shorter than expected.
    - The maximum number of hops of a loop free path is approximately 30 hops.



**Internet 2005**

Picture taken from http://www.opte.org

## The Internet

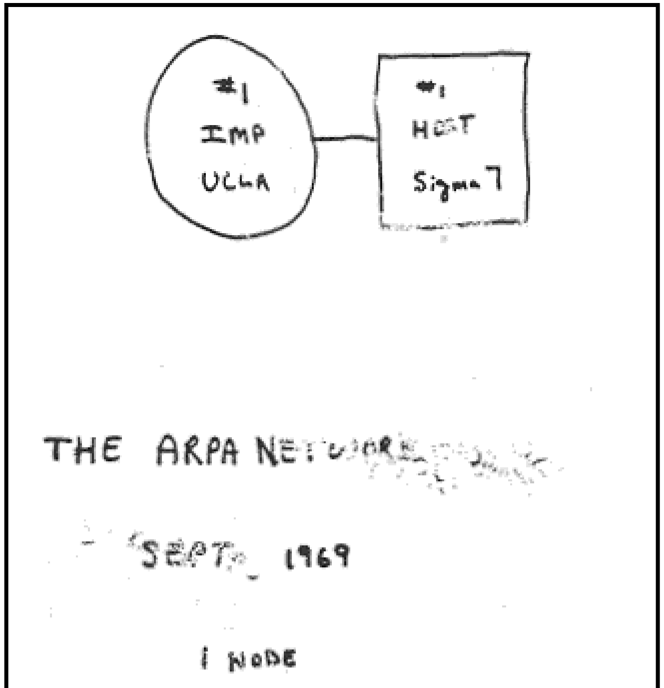



FIGURE 6.1 Drawing of September 1969 (Courtesy of Alex McKenzie)

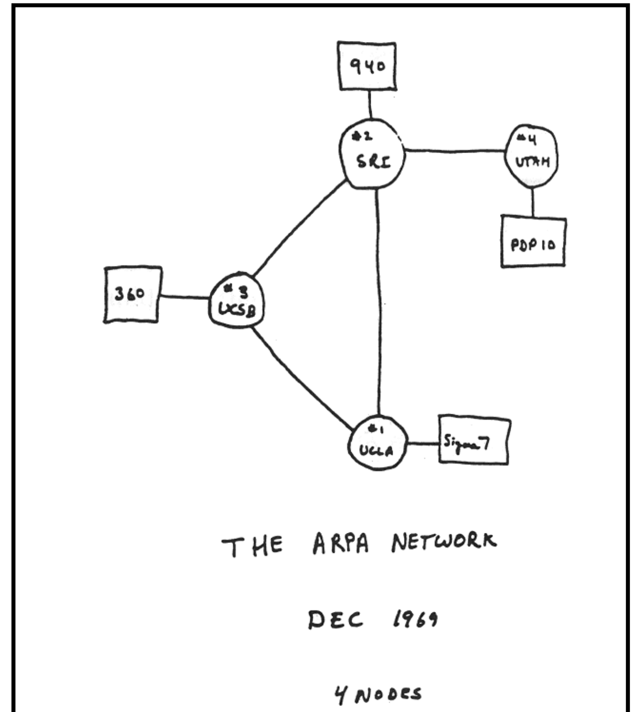



FIGURE 6.2 Drawing of 4 Node Network (Courtesy of Alex McKenzie)

**ARPANET September 1969**

**ARPANET December 1969**

**The Internet was a success story already from the beginning where it increased its size within 3 months by a factor of four!**

Pictures taken from http://www.cybergeography.org/atlas/historical.html

# Random Graphs

- Scale-free networks – real-world examples**:**
  - **Social networks - Facebook:**


**Facebook Friendships**

Picture taken from http://www.opte.org