Peer-to-Peer Networks

Chapter 4: Graphs and Methods

Thorsten Strufe

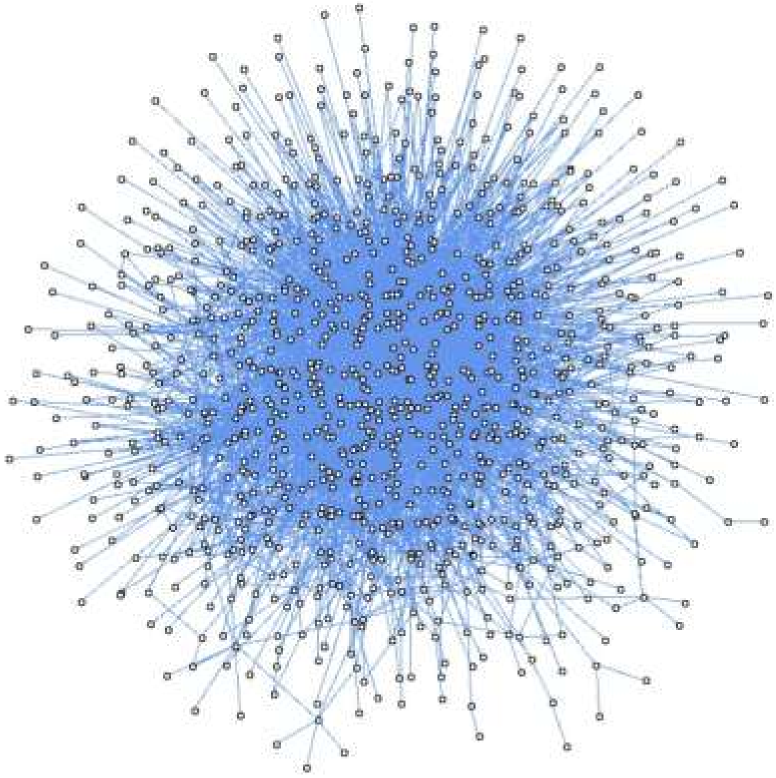
Note: these slides have been prepared with influence by material of Prof. Michael Welzl, Prof. Pietro Michiardi, and Dr. Stefan Schmid

Chapter Outline

- P2P Overlays as Graphs (This chapter is a reminder)
- Graphs
- Metrics in and Properties of Graphs
- Algorithms on Graphs
- A tiny introduction to game theory

Some questions…

- How scalable is Gnutella?
- How robust is Kazaa?
- Why does FreeNet work?
- What would an ideal (unstructured) P2P system look like?
- What do the overlay networks of existing P2P systems look like?



Gnutella snapshot, 2000

Scalability of Gnutella: quick answer

- Bandwidth Generated in Bytes (Message 83 bytes) *[SIC]
  - Searching for a 18 byte string

	T=2	T=3	T=4	T=5	T=6	T=7	T=8
N=2	332	498	664	830	996	1,162	1,328
N=3	747	1,743	3,735	7,719	15,687	31,623	63,495
N=4	1,328	4,316	13,280	40,172	120,848	362,876	1,088,960
N=5	2,075	8,715	35,275	141,515	566,475	2,266,315	9,065,675
N=6	2,988	15,438	77,688	388,938	1,945,188	9,726,438	48,632,688
N=7	4,067	24,983	150,479	903,455	5,421,311	35,528,447	192,171,263
N=8	5,312	37,848	262,600	1,859,864	13,019,712	91,138,648	637,971,200

  - N = number of connections
  - T = number of hops

** [SIC]: Error already in source, orders of magnitude are important, here ;)*
Source: Jordan Ritter: Why Gnutella Can't Scale. No, Really.

Graphs

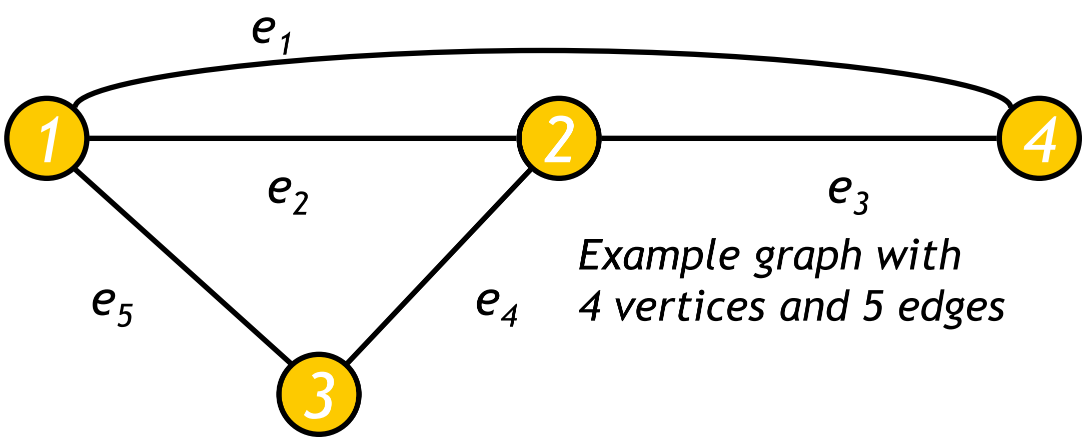
- Rigorous analysis of P2P systems: based on graph theory
  - Refresher of graph theory needed
- First: graph families and models
  - Random graphs
  - Small world graphs
  - Scale-free graphs
- Then: graph theory and P2P
  - How are the graph properties reflected in real systems?
    - Users (peers) are represented by vertices in the graph
    - Edges represent connections in the overlay (routing table entries)
- Concept of self-organization
  - Network structures emerge from simple rules
  - E.g. also in social networks, www, actors playing together in movies

What Is a Graph?

- Definition of a graph:

  Graph $G = (V, E)$ consists of two finite sets, set V of vertices (nodes) and set E of edges (arcs) for which the following applies:

  1. If $e \in E$, then exists $(v, u) \in V \times V$, such that $v \in e$ and $u \in e$
  2. If $e \in E$ and above (v, u) exists, and further for $(x, y) \in V \times V$ applies $x \in e$ and $y \in e$, then $\{v, u\} = \{x, y\}$



Example graph with 4 vertices and 5 edges

Side note:

Edges can have (multiple) “weights” $w : E \rightarrow \mathbb{R}$

How are Graphs Implemented?

- Adjacency/Incidence Matrix

	1	**2**	**3**
1	0	1	0
2	1	0	1
3	0	1	0

- Adjacency/Incidence List

(1,2)
(2,1),(2,3)
(3,2)

1:2
2:1,3
3:2

- (Plus specialized others..)

VERY good book is: Sedgewick: Algorithms in C, part 3 (Graph Algorithms)

Properties of Graphs

- An edge $e \in E$ is directed if the start and end vertices in condition 2 above are identical: $v = x$ and $y = u$
- An edge $e \in E$ is undirected if $v = x$ and $y = u$ as well as $v = y$ and $u = x$ are possible
- A graph G is directed (undirected) if the above property holds for all edges
- A *loop* is an edge with identical endpoints
- Graph $G_1 = (V_1, E_1)$ is a subgraph of $G = (V, E)$, if $V_1 \subseteq V$ and $E_1 \subseteq E$ (such that conditions 1 and 2 are met)

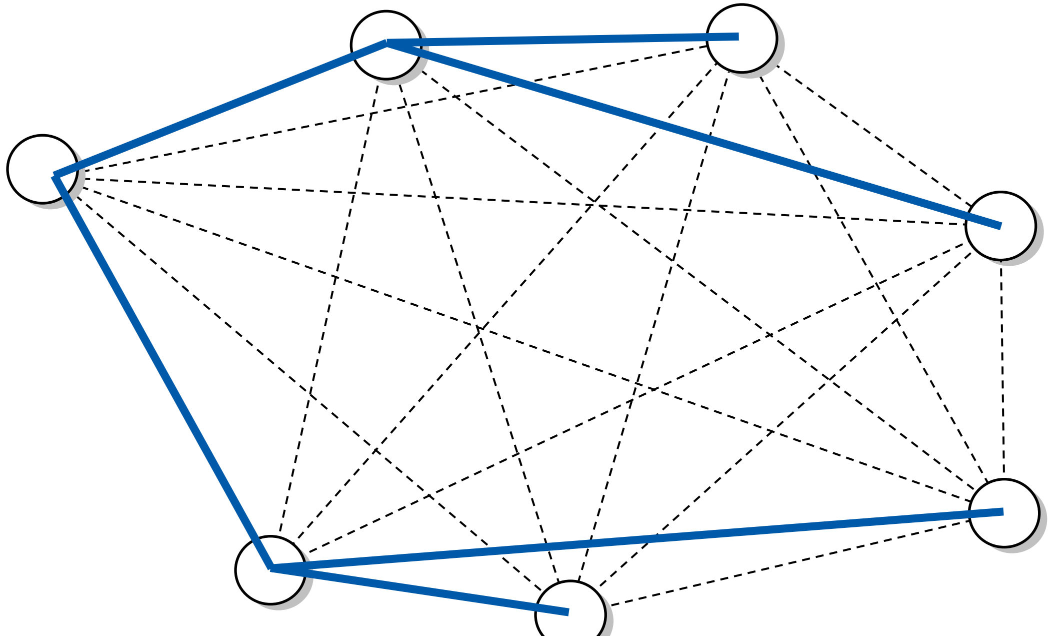
Important Types of Graphs

- Vertices $v, u \in V$ are connected if there is a path from v to u: $(v, v_2), (v_2, v_3), \ldots, (v_{k-1}, u) \in E$
- Graph G is connected if all $v, u \in V$ are connected
  - Strong connectivity of directed graphs means, that paths between each node pair exist
  - Weak connectivity: edges between all node pairs exist, but not paths…
- An undirected, connected, acyclic graph is called a tree
  - Side note: Undirected, acyclic graphs which are not connected are called forest
- Directed, connected, acyclic graph is called DAG
  - DAG = Directed Acyclic Graph (connectivity is assumed)
- An induced graph $G(V_C) = (V_C, E_C)$ is a graph $V_C \subseteq V$ and with edges $E_C = \{e = (i, j) \mid i, j \in V_C\}$ (all edges from G connecting the nodes in G_C)
- An induced graph that is connected is called a component

Overlays?

- A CLIQUE is a graph that is fully connected

  $(u,v) \in E \mid$ for all $u \in V$ and $v \in V$, $u \neq v$
- A P2P Overlay (V_o,E_o) (in general) is a subgraph such that $V_o=V$ and $E_o \subseteq E$ (edges are selected edges from a CLIQUE graph)



- ***Why?*** Considering the nodes to be on the internet, they all can create connections between each other…

Important Graph Metrics

- Order: the number of vertices in a graph
- Size of the graph is the number of edges $|E|$

- Distance: $d(v, u)$ between vertices v and u is the length of the shortest path between v and u

- Diameter: $d(G)$ of graph G is the maximum of $d(v, u)$ for all $v, u \in V$

- The density of a graph is the ratio of the number of edges and the number of possible edges.

Graph Metrics: Vertex Degree

- In graph $G = (V, E)$, the degree of vertex $v \in V$ is the total number of edges $(v, u) \in E$ and $(u, v) \in E$
  - Degree is the number of edges which touch a vertex

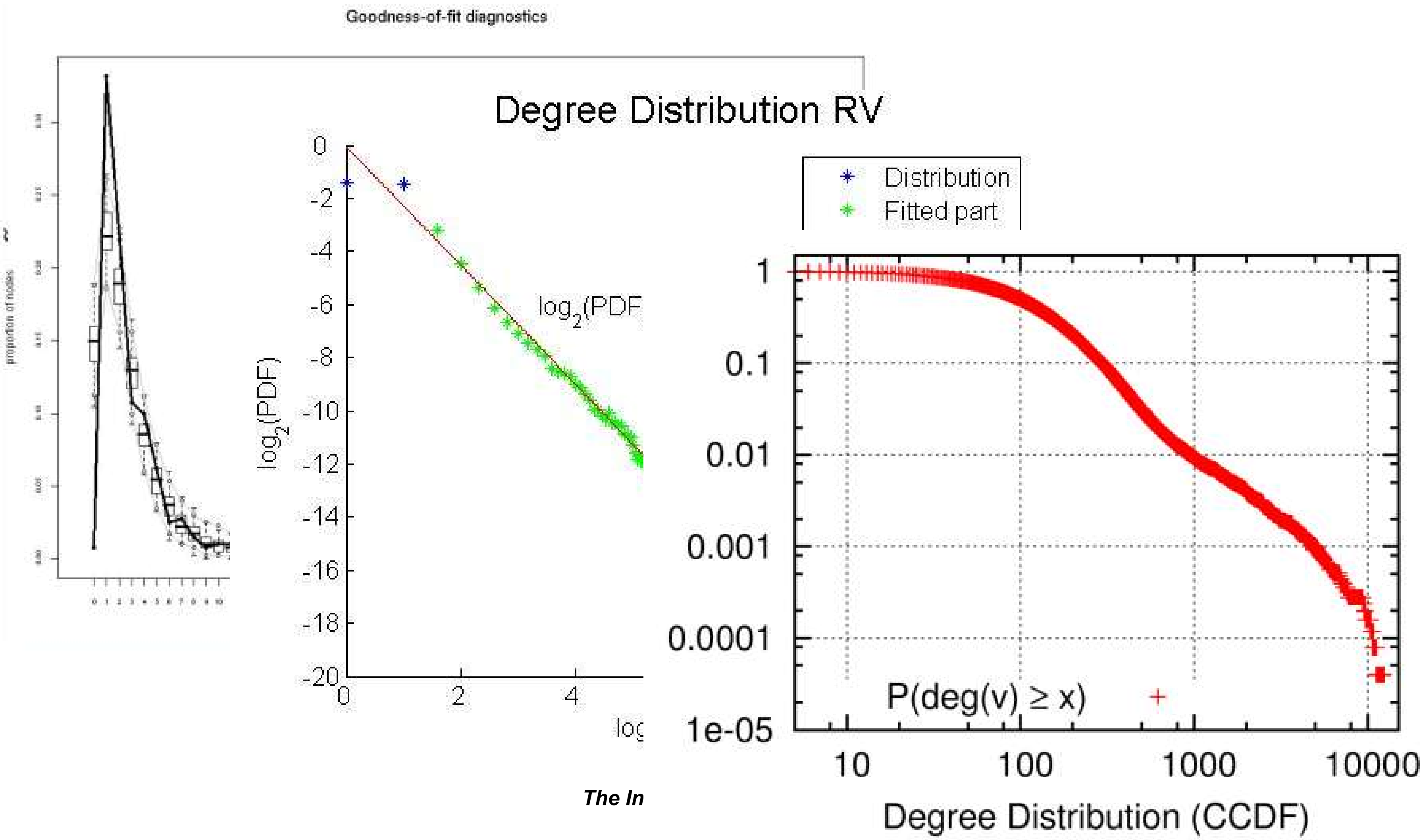
- For directed graph, we distinguish between in-degree and out-degree
  - In-degree is number of edges coming to a vertex
  - Out-degree is number of edges going away from a vertex

- The degree of a vertex can be obtained as:
  - Sum of the elements in its row in the incidence matrix
  - Length of its vertex incidence list

- The ***degree distribution*** is the distribution over all node degrees

  (given as a frequency distribution or (often) complementary cumulative distribution function CCDF (Komplement der Verteilungsfunktion))

Graph Metrics: Degree Distribution (Examples)



Online Social Network (xing crawl) [strufe10popularity]

Further Graph Metrics

- All pairs shortest paths (APSP): *d(v, u)* | all v,u ∈ V
- Hop Plot: Distance distribution over all distances *Hist(APSP(G))*
- Average/characteristic path length: Sum of the distances over all pairs of nodes divided by the number of pairs

For defined routings on graphs:

Characterisic Routing Length: average length of paths *found* (potentially stochastic…)

Sanity Check:

- APSP:

1,1,1,2,2

1,1,1,2,2

1,1,1,1,1

1,1,1,1,2

1,1,1,2,2

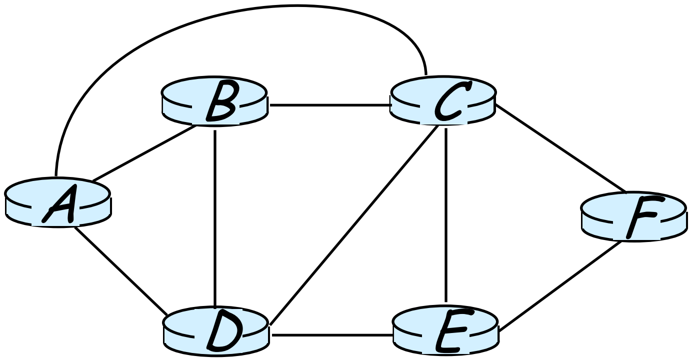
1,1,2,2,2

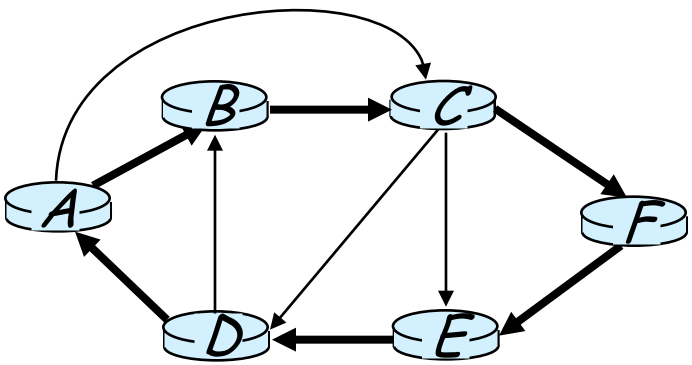
- HopPlot:

1: 20

2: 10 *40/30 = 1.333*

- CPL: *2.033*
- CRL (CHORD):





Important Graph Metrics (3)

- Edge connectivity: is the minimum number of edges that have to be removed to separate the graph into at least two components

- Vertex connectivity: the minimum number of nodes..

- How to calculate them?
- Which of both is higher?
- In which cases are they the same?

- maxflow, Menger‘s Theorem…

- *Does each network have **ONE** maximum flow?*

- *What is the edge/vertex connectivity if we have a node with degree=1?*
  - *Is this a sensible metric? How to „heal“ this?*

==> balanced cut, size of the remaining giant connected component, fraction of disconnected nodes

Important Graph Metrics (2)

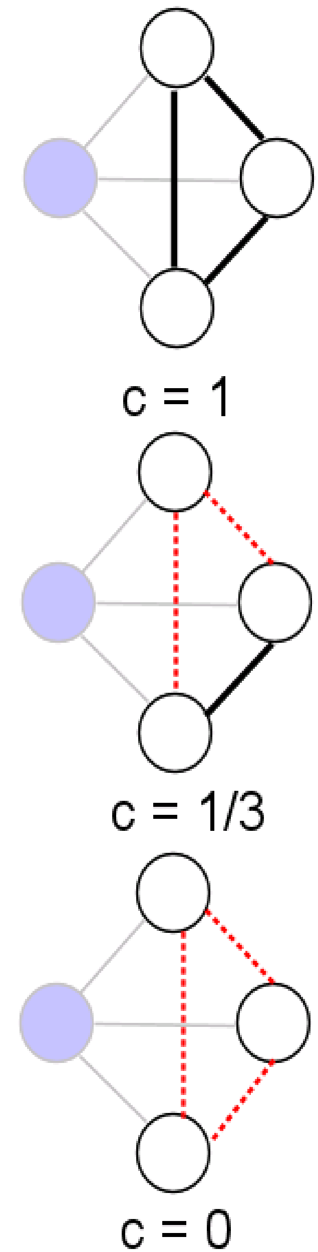
- Clustering coefficient: number of edges between neighbors divided by maximum number of edges between them
  - k neighbors: k(k-1)/2 possible edges between them

$$C(i) = \frac{2E(N(i))}{d(i)(d(i)-1)}$$

E(N(i)) = number of edges between neighbors of i

d(i) = degree of i

- ***What if***: a node has only one neighbor? ☺



Source: Wikipedia

P2P Self Organization and Routing

- So what can a peer actually do?
  - *(Initially: bootstrapping, ID selection)*
- Select neighbors
  - Randomly
  - According to some rule
- Select next hop (when delegation is needed..)
  - Randomly
    - Single next hop
    - Multiple next hops (request replication… flooding)
  - According to some rule
- *(Change ID, but that‘s already advanced.. ☺)*

So what!?

Location Overlay $\mathcal{L} = (V_L, E_L)$ *(of Vertices **V** and Edges **E**)*

- Reliability
  - High success probability

→ ***hit ratio*** = $\frac{|Hit|}{|Requests|}$

- Low response times

→ ***response time*** = $\frac{\sum_{i=1}^{|Requests|} (t_{hit_i} - t_{req_i})}{|Requests|}$

- Resource usage
  - Low message complexity

→ ***message complexity*** = $\frac{\sum_{m_e \in \mathcal{M}} m_e \cdot d(e)}{|Requests|}$

- ***Great! We can do maths, now!*** ☺ (Plus: we can define metrics..)

Classes of Graphs

- Regular graphs
- Random graphs
- Graphs with Small-World characteristic
- Scale-free graphs
- …Graphs with plenty more characteristics
  - (dis-) assortativity
  - Rich-club connectivity
  - …