Application and analysis of random walks in peer-to-peer and ad hoc networks

G. Haßlinger, T-Systems & S. Kempken, Univ. Duisburg-Essen
E-Mail: gerhard.hasslinger@telekom.de & kempken@inf.uni-due.de

- Flooding, random walks and combined search methods
- Transient analysis of basic random walks and variants
- Evaluation for different ("un-")structured networks
- Conclusions

Relevance of „unstructured“ networks

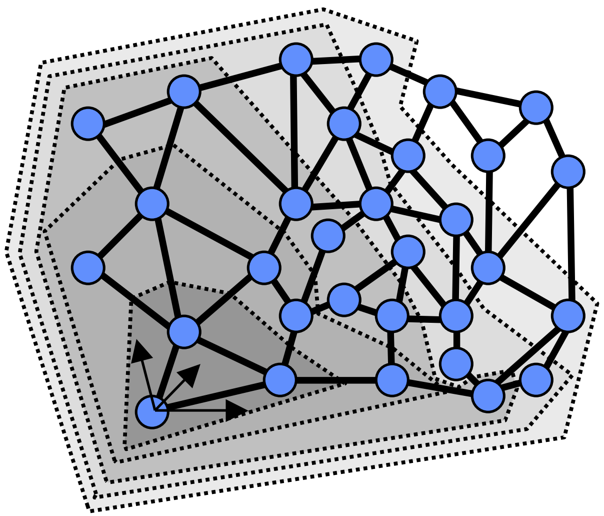
Some communication networks and overlays don`t offer direct support for search or routing, e.g.

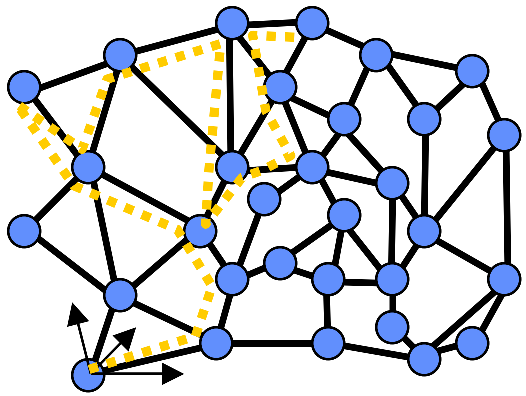
- mobile ad hoc networks
- sensor network
- some peer-to-peer networks (Gnutella)

Advantage: More flexibility to set up and expand networks,
less overhead in managing networks with high churn
Disadvantage: More expensive search by exploration of the network

The Internet itself exhibits unstructured growth and churn, but today Google provides content exploration and a search index and the IETF (Cisco, Juniper ...) has established a routing scheme

Network exploration by flooding & random walks





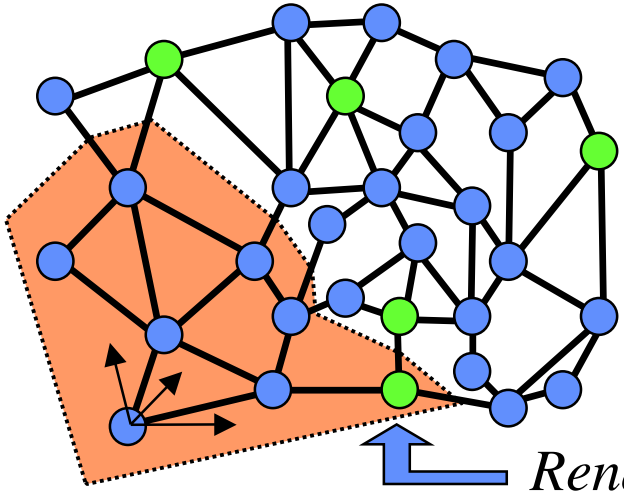
- Flooding is exhaustive for all neighbors up to a distance *d* or time to live (TTL);
  Parallel search; large amount of messages spread in all directions
- Random walks follow some probabilistic winding route

Network exploration by flooding & random walks

- **Control** of message overhead **for flooding is difficult:**
  Unknown network coverage as a function of the distance d
  Coverage may rise e.g. from 3% to 30% in one step $d \to d + 1$

- A random walk of predefined length L has fixed expense
  Randoms walks can proceed **in parallel, with forking or**
  can be **enhanced by flooding** with small d from some points
  $\Rightarrow$ **Many ways to combine random walks & flooding schemes**
  (see Gkantsidis et al., IEEE Infocom 2004 & '05)
  Network coverage is partial, but random walk searches are
  efficient e.g. for replicated data in P2P networks
  Random network growth is also efficiently supported by r. walks

P2P systems with randomized processes, e.g., BubbleStorm



The BubbleStorm approach:

Search or Query Bubble: Q

- Set of nodes traversed by a search

Data Bubble: D

- Set of nodes with replicated data enabling to serve the query

Rendezvous node set: $D \cap Q$

If D is a random subset of the overlay V, then $D \cap Q$ is empty with probability $< (1 - |D|/|V|)^{|Q|} < e^{-s}$ for $|D|\cdot|Q| > s|V|$
If e.g. $|D|=|Q| > 4\sqrt{|V|}$
then the query is served with prob. $> 1 - e^{-16} > 0.999\,999\,8...$

Source: www.dvs1.informatik.tu-darmstadt.de/research/bubblestorm, ACM SIGCOMM´07

Transient analysis of basic random walks

Performance studies on random walks in P2P prefer simulation, although transient analysis offers a simple and scalable alternative

Bounds based on the second largest eigenvalue of the transition matrix prove convergence but are not very tight (Gkantsidis Infocom'04)

Performance criteria are

1. Convergence of a random walk to steady state
2. Network coverage of a random walk of length *L*

Transient analysis
- computes the probability distribution for the random walks' sojourn node step by step
- starting from a node or an arbitrary initial distribution

Transient analysis of a random walk: 1. Convergence

Complete implementation of the transient analysis for **convergence to steady state** is as simple as this:

```
for (k = first_step; k <= last_step; k++)
{ for (j = first_node; j <= last_node; j++) new_probability[j] = 0;
  for (j = first_edge; j <= last_edge; j++)
      new_probability[edge_destination[j]] +=
            probability[edge_start[j]] / node_degree[edge_start[j]];
  for (j =  first_node; j <= last_node; j++) probability[j] = new_probability[j];
};           //   Comment: Obviously self-explaining C++ code ...
```

The run time complexity of the transient analysis of the random walk convergence is proportional to

$$\left.\begin{array}{l}\text{- the number of steps of the walk}\\ \text{- the number of network edges}\end{array}\right\} \Rightarrow \text{ works for large scale networks}$$

Transient analysis of a random walk: 2. Network Coverage

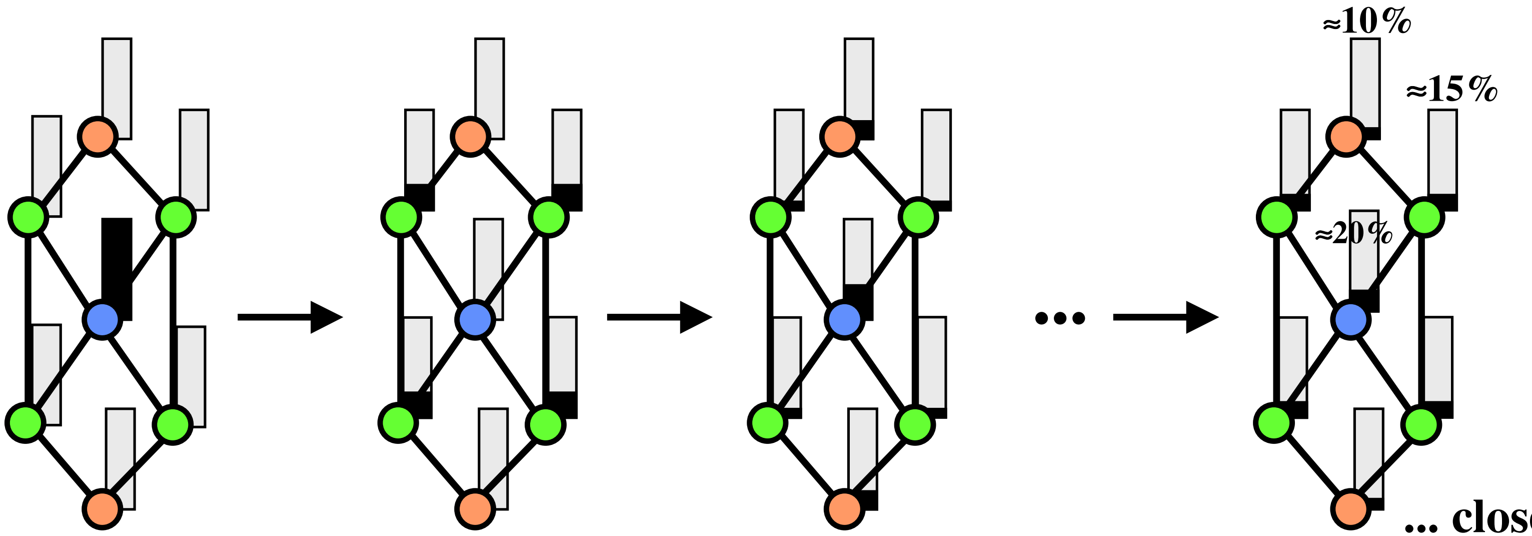
An absorbing state („black hole“) is introduced at a considered network node

⇒ The probability to enter the absorbing state from some starting conditions, e.g. from steady state, equals the probability to discover the network node during the random walk

For networks with heterogeneous nodes, coverage can be studied depending on different types or degrees of nodes

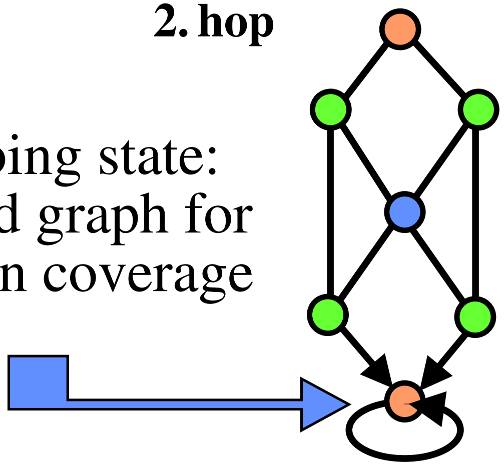
Implementation:
A few lines have to be added to the previous code without affecting the complexity

Demonstration of a transient random walk analysis



Start — **1. hop** — **2. hop** — **12. hop: ... close to steady state**

Absorbing state:
Modified graph for
results on coverage



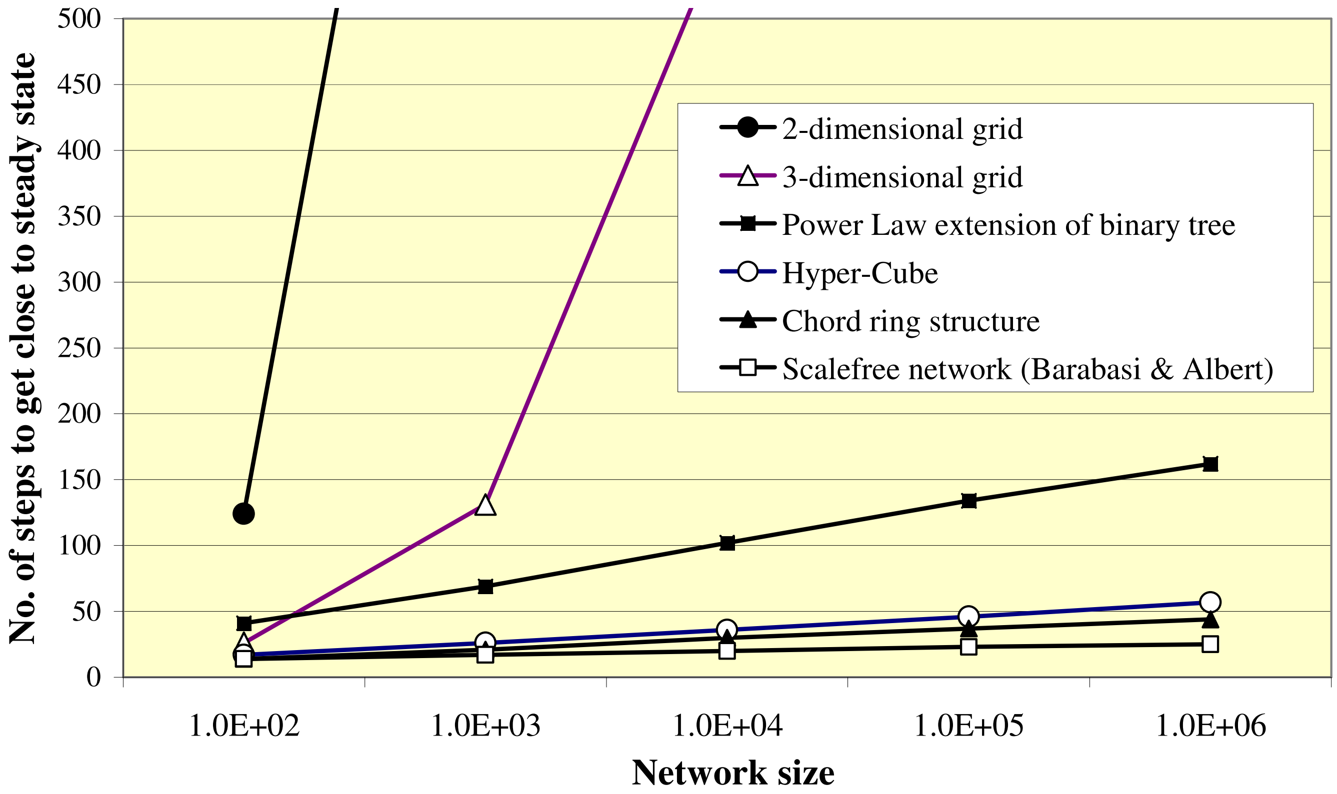
⇒ Prob. to enter a state within *k* hops

Application to different types of networks

Network type	Number of edges	Min. node degree	Max. node degree	Diameter
2-dim. grid, wrapped	2 $\|V\|$	4	4	$O(\sqrt{\|V\|})$
3-dim. grid, wrapped	3 $\|V\|$	6	6	$O(\sqrt[3]{\|V\|})$
Hyper cube	$\|V\| \log_2\|V\|/2$	$\log_2\|V\|$	$\log_2\|V\|$	$\log_2\|V\|$
Chord structure: ring & unidirectional pointers	$\|V\| \log_2\|V\|$	$\log_2\|V\|$	$\log_2\|V\|$	$\log_2\|V\|$
Binary Tree	$\|V\| - 1$	1	3	$O(\log_2\|V\|)$
Power law extension of a binary tree	$(\log_2\|V\| - 1) \cdot (\|V\| + 1) + 2$	$\log_2\|V\| - 1$	$\|V\| - 1$	2
Scale-free networks (Barabasi & Albert)	2 K $\|V\|$	K	Large	Small

Results: Convergence to steady state

Number of steps of the random walk until $\Delta = \sum_{n \in V} | p_L(n) - q(n) | \leq 0.01$

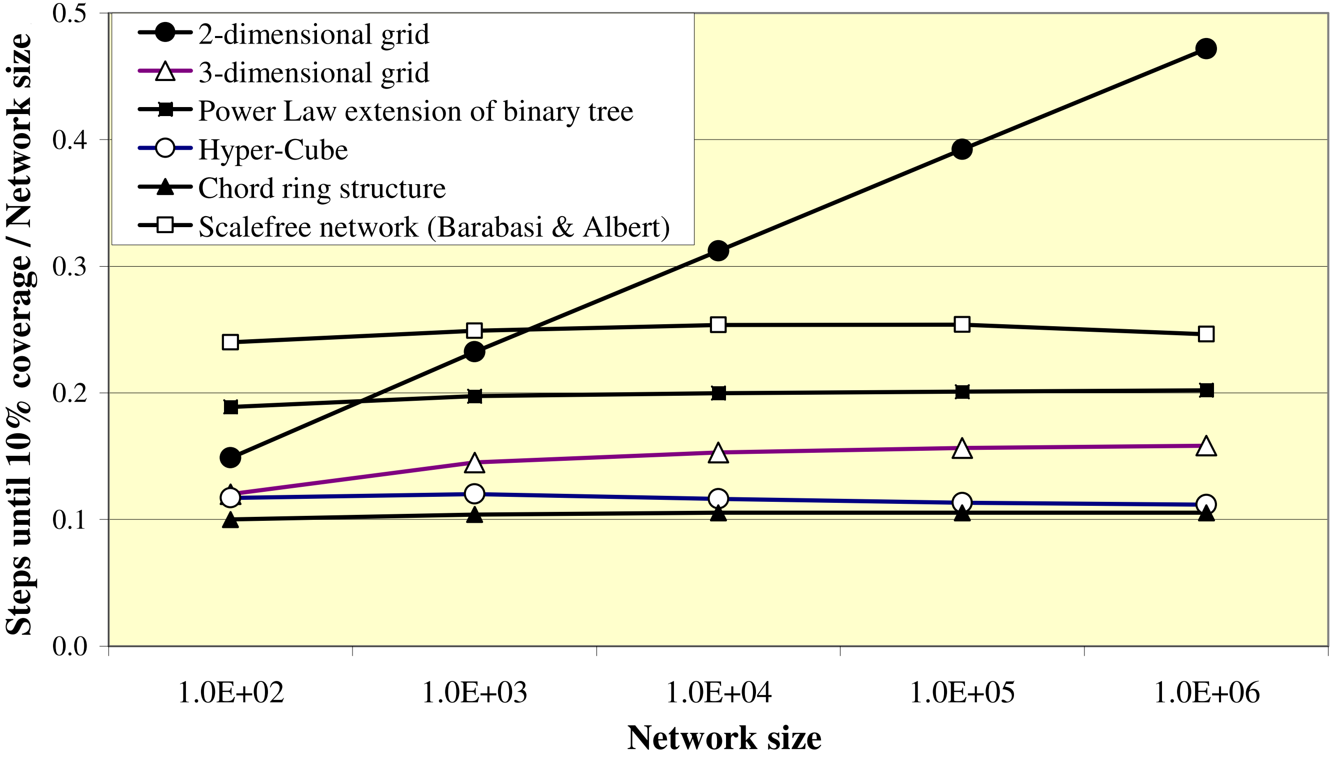


$q(n)$: steady state distribution

$q_L(n)$: distribution of the current sojourn node of the random walk after L steps

Start from a node with smallest degree in hetero-geneous networks (worst case)

Results: Network Coverage



Random walk starts in steady state

The absorbing state is a node with smallest degree (worst case)

Nodes of high degree are often reached in a few steps

Number of steps of the random walk until 10% of the network nodes are visited; this is usually sufficient to find replicated data in P2P networks

Transient analysis of a random walk: Extensions for several variations

The following cases are tractable by extended analysis:

- Random walk without step back (except for nodes of degree 1)
  $\rightarrow$ increased state space for analysis: network edges instead of nodes, but the run time complexity is unchanged
- Random walk followed by flooding on distance *d* after the last step
  $\rightarrow$ extend absorbing state to the set of all neighbors up to distance *d*
- Several random walks in parallel
  $\rightarrow$ product formula for the probability that independent trials miss a node
- Random walk search for replicated data on *n* nodes
  $\rightarrow$ use a set of *n* absorbing states or assume a binomial distributed hit count based on single node search

Conclusions

- Random walks are useful for network search and scalefree network expansion, in combination with flooding
- Transient analysis yields accurate evaluation of random walks
  - for the basic case and many variants
  - is scalable for networks of large size
- Efficiency of random walks depends on the network structure and differs for
  - convergence to steady state,
    $\rightarrow$ is fastest for low (e.g. logarithmic) diameter
  - and network coverage
    $\rightarrow$ is fastest for some homogeneous network types

Related work

C. Gkantsidis, M. Mihail and A. Saberi, Random walks in peer-to-peer networks, Proc. IEEE Infocom (2004)

C. Gkantsidis, M. Mihail and A. Saberi, Hybrid search schemes for unstructured P2P networks, Proc. IEEE Infocom (2005)

M. Zhong and K. Shen, Popularity-biased random walks for peer-to-peer search under the square root principle, Proc. 5th Internat. Workshop on P2P systems, IPTPS (2006)

A.-L. Barabási and R. Albert, Emergence of scaling in random networks, Science 286 (1999) 509-512

D. Coppersmith, U. Feige and J. Shearer, Random Walks on Regular and Irregular Graphs, SIAM Journal on Discrete Mathematics 9/2 (1996) 301-308

W. Terpstra, J. Kangasharju, C. Leng and A. Buchmann, BubbleStorm: Resilient, probabilistic and exhaustive P2P search, to appear ACM SIGCOMM 2007, Kyoto, Japan <www.dvs1.informatik.tu-darmstadt.de/research/bubblestorm>