Peer-to-Peer Networks
09 Self-Organization

Christian Schindelhauer

Technical Faculty

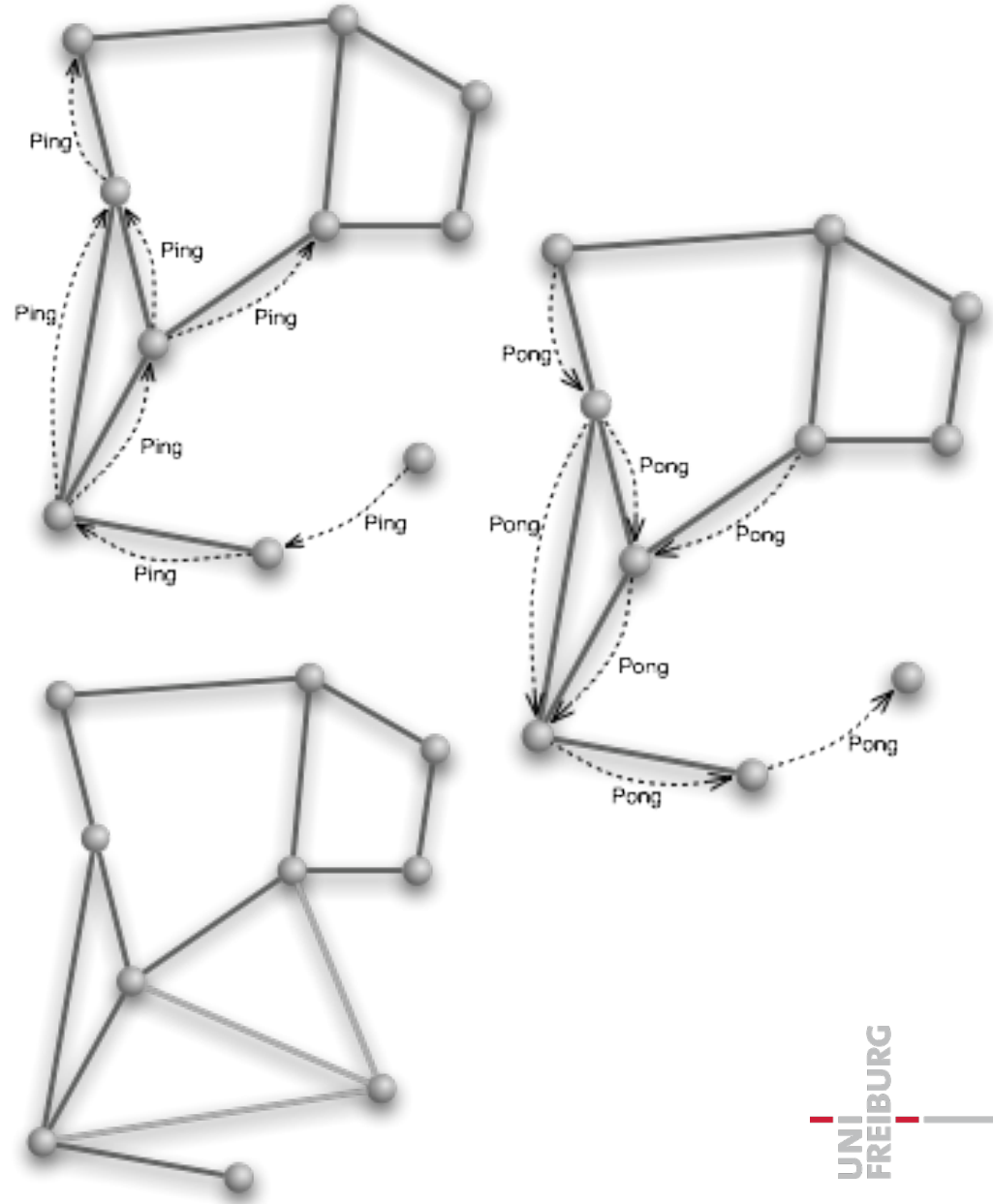Computer-Networks and Telematics

University of Freiburg

Protokoll

- Ping
  - participants query for neighbors
  - are forwarded according for TTL steps (time to live)
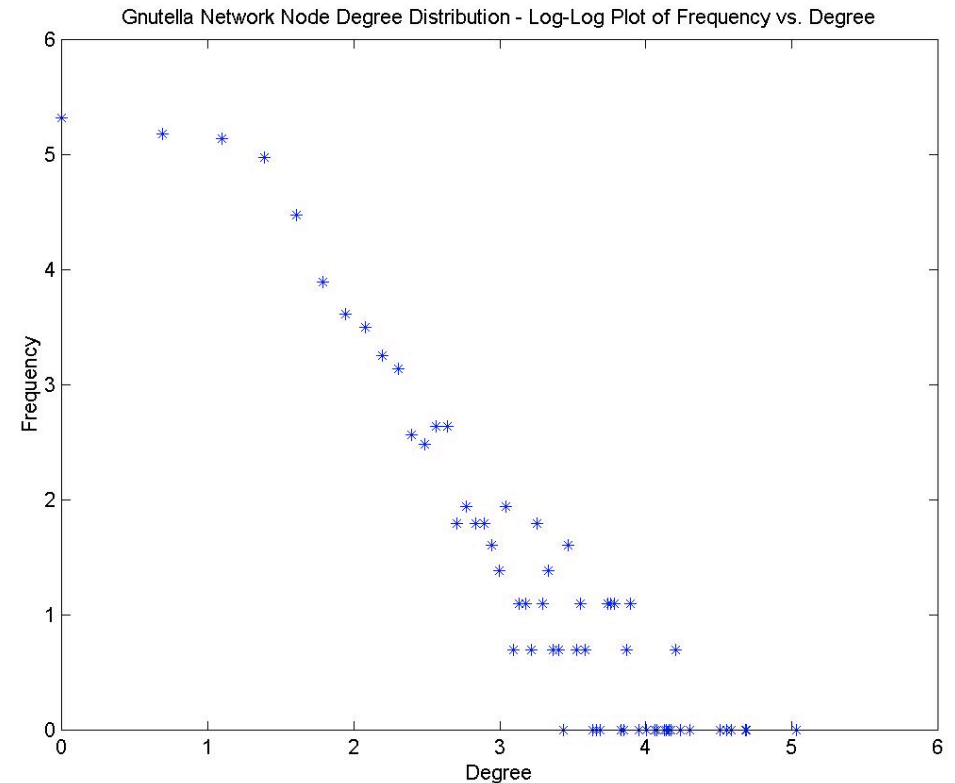- Pong
  - answers Ping
  - is forwarded backward on the query path
  - reports IP and port adress (socket pair)
  - number and size of available files

Degree Distribution in Gnutella

- Modeling Large-scale Peer-to-Peer Networks and a Case Study of Gnutella
    - Mihajlo A. Jovanovic, Master Thesis, 2001
- The number of neighbors is distributed according a power law (Pareto) distribution
    - log(#peers with degree d) = c - k log d
    - #peers with degree d = C/d^k

Gnutella Network Node Degree Distribution - Log-Log Plot of Frequency vs. Degree

Pareto-Distribution Examples

- Pareto 1897: Distribution of wealth in the population

- Yule 1944: frequency of words in texts

- Zipf 1949: size of towns

- length of molecule chains

- file length of Unix-system files

- ….

Pareto Verteilung

- Discreet Pareto-Distribution for $x \in \{1,2,3,\dots\}$

$$\mathbf{P}[X = x] = \frac{1}{\zeta(\alpha) \cdot x^{\alpha}}$$

  - with constant factor

$$\zeta(\alpha) = \sum_{i=1}^{\infty} \frac{1}{i^{\alpha}}$$

  - (also known as Riemann´s Zeta-function)
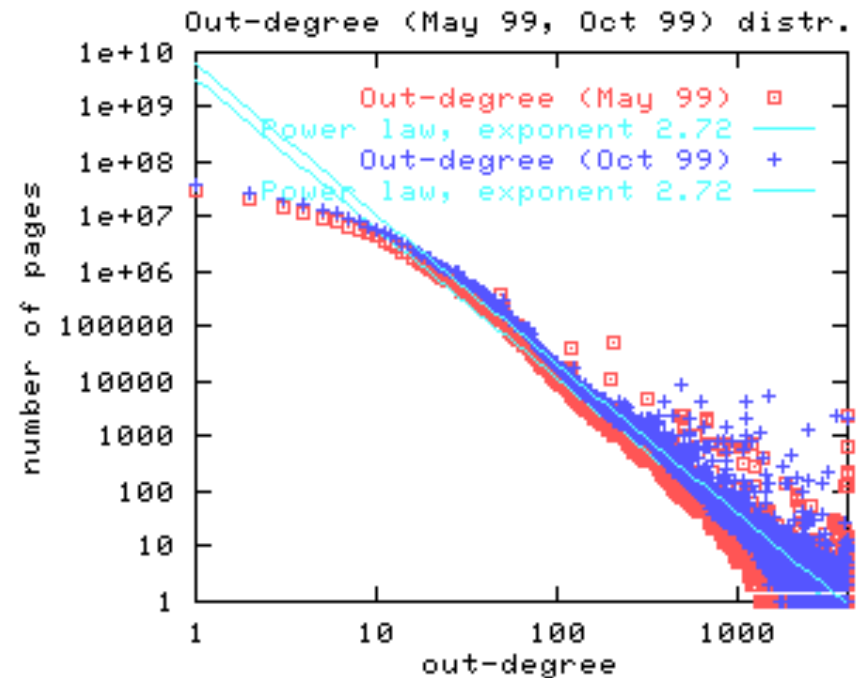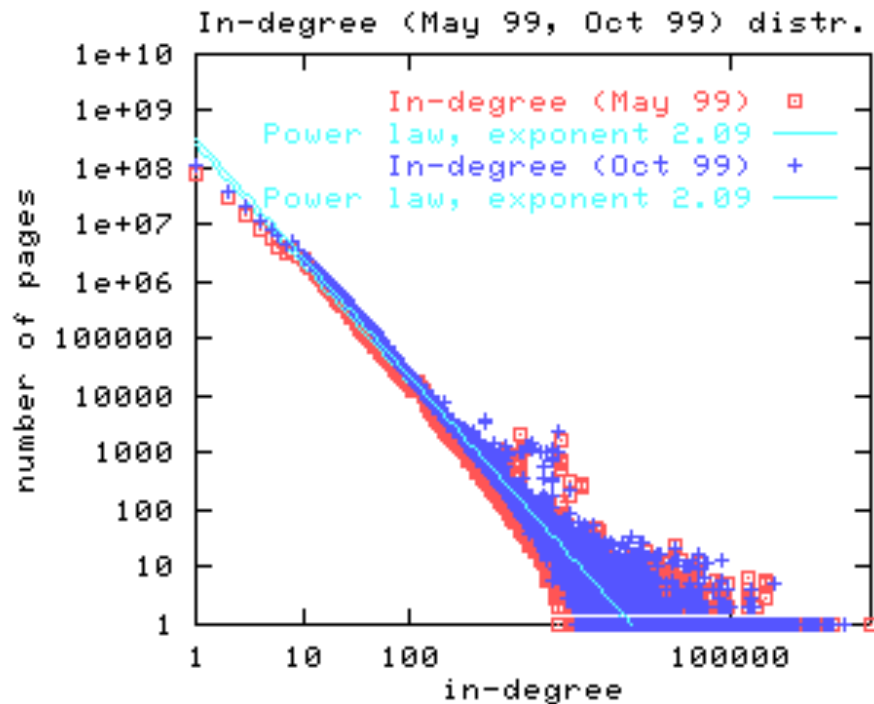
- Heavy tail property

  - not all moments $E[X^k]$ exist

  - the expectation exists if and only if (iff) $\alpha > 2$

  - variance and $E[X^2]$ exist iff $\alpha > 3$

  - $E[X^k]$ exists iff $\alpha > k+1$

- Density function of the continuous function for $x > x_0$

$$f(x) = \frac{\alpha - 1}{x_0} \left(\frac{x_0}{x}\right)^{\alpha}$$

Indegree and Outdegree of Web-Pages

- are described by a power law (Pareto) distribution

- Experiments of
  - Kumar et al 97: 40 millions Webpages
  - Barabasi et al 99: Domain *.nd.edu + Web-pages in distance 3
  - Broder et al 00: 204 millions web pages (Scan Mai und Okt. 1999)

Connectivity of Pareto Graphs

- William Aiello, Fan Chung, Linyuan Lu, A Random Graph Model for Massive Graphs, STOC 2000

- Undirected graph with n nodes where

  - the probability of k neighbors for a node is p_k

  - where $p_k = c\,k^{-\tau}$ for some normalizing factor c

- Theorem

  - For sufficient large n such Pareto-Graphs with exponent τ we observe

    - for τ < 1 the graph is connected with probability 1-o(1)

    - for τ > 1 the graph is nont connected with probability 1-o(1)

    - for 1< τ <2 there is a connected component of size Θ(n)

    - for 2< τ < 3.4785 there is only one connected component of size Θ(n) and all others have size O(log n)

    - for τ >3.4785: there is no large connected component of size Θ(n) with probability 1-o(1)

    - For τ >4: no large connected components which size can be described by a power law (Pareto) distribution

Zipf Distribution as a Variant of Power Laws

- George Kinsley Zipf claimed
    - that the frequency of the n most frequent word f(n)
    - satisfies the equation n f(n) = c.
- Zipf probability distribution for x ∈ {1,2,3,…}

$$\mathbf{P}[X = x] = \frac{c}{x}$$

   - with constant factor c only defined for connstant sized sets, since

$$\ln n \leq \sum_{i=1}^{n} \frac{1}{i} \leq 1 + \ln n$$

   - is unbounded
- Zipf distribution relate to the rank
    - The Zipf exponent α may be larger than 1, i.e. $f(n) = c/n^{\alpha}$
- Pareto distribution realte the absolute size, e.g. the number of inhabitants

Figure 1 The hierarchical differentiation in urban systems:
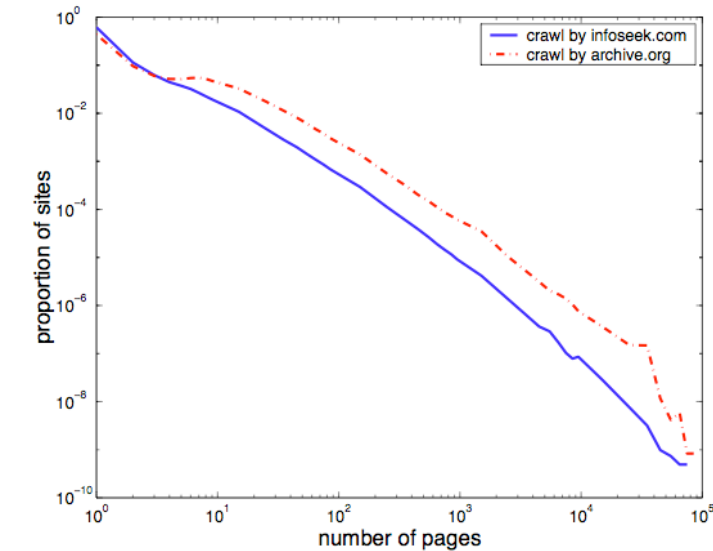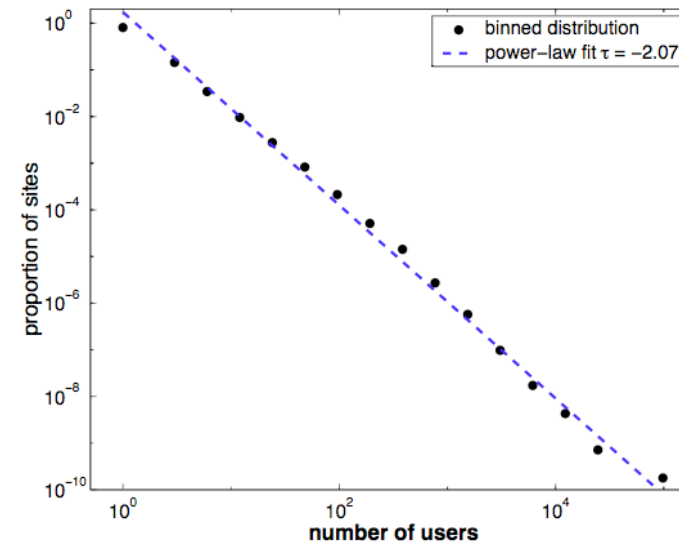Rank-size distribution of French agglomerations (1831-1999)
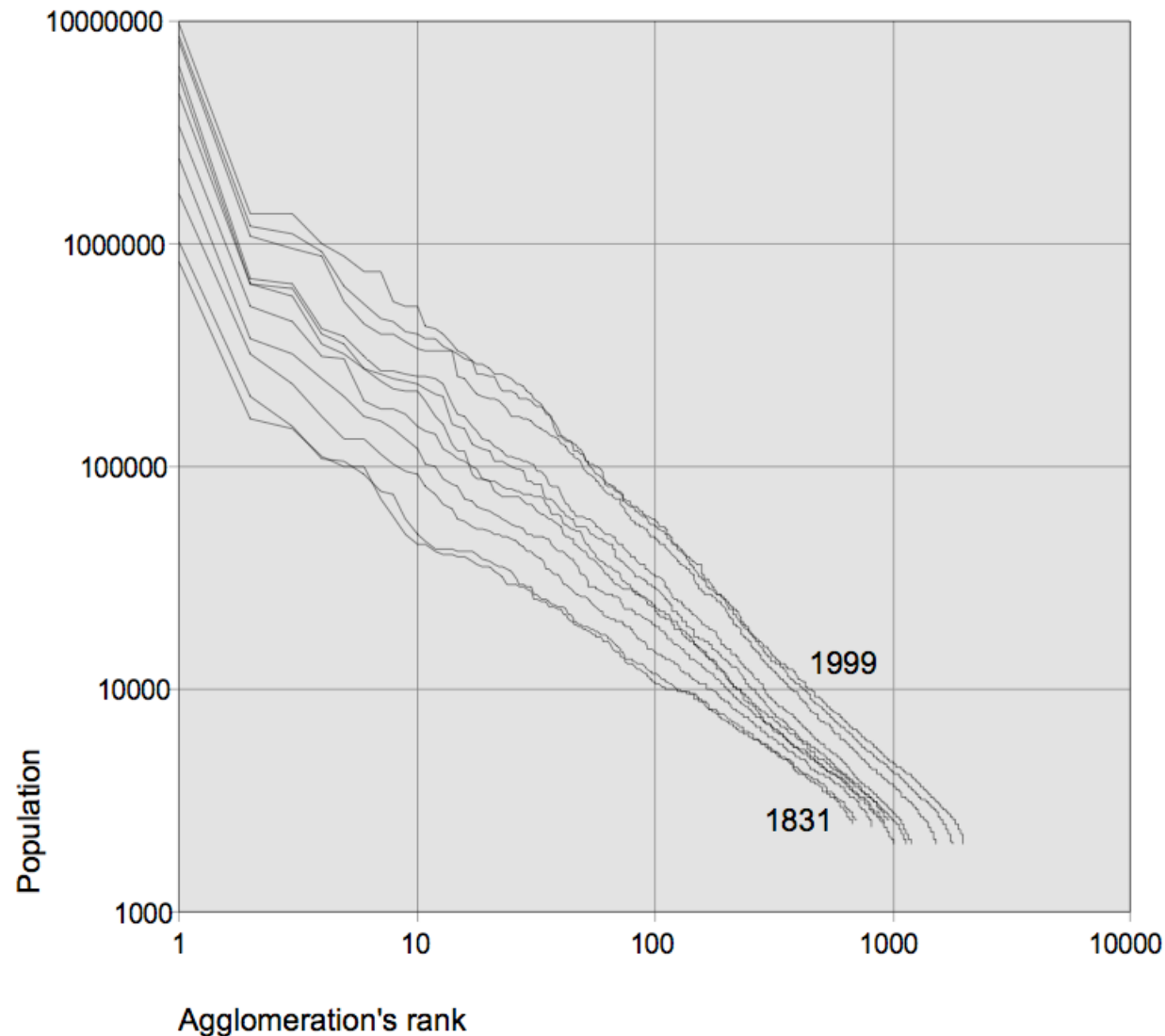
Zipf distribution

Zipf's Law and the Internet

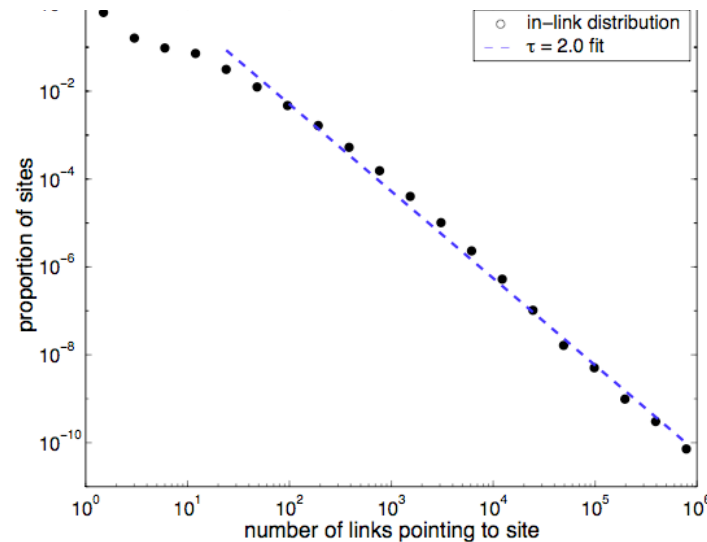## Lada A. Adamic, Bernardo A. Huberman, 2002

Pareto
Distribution!!

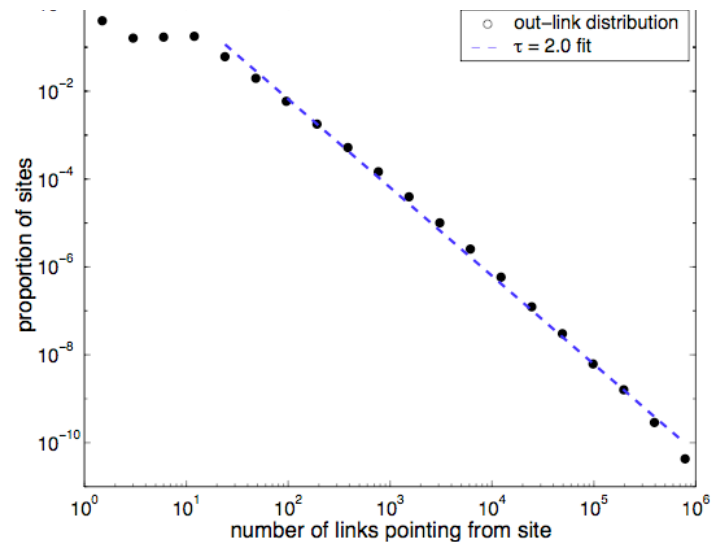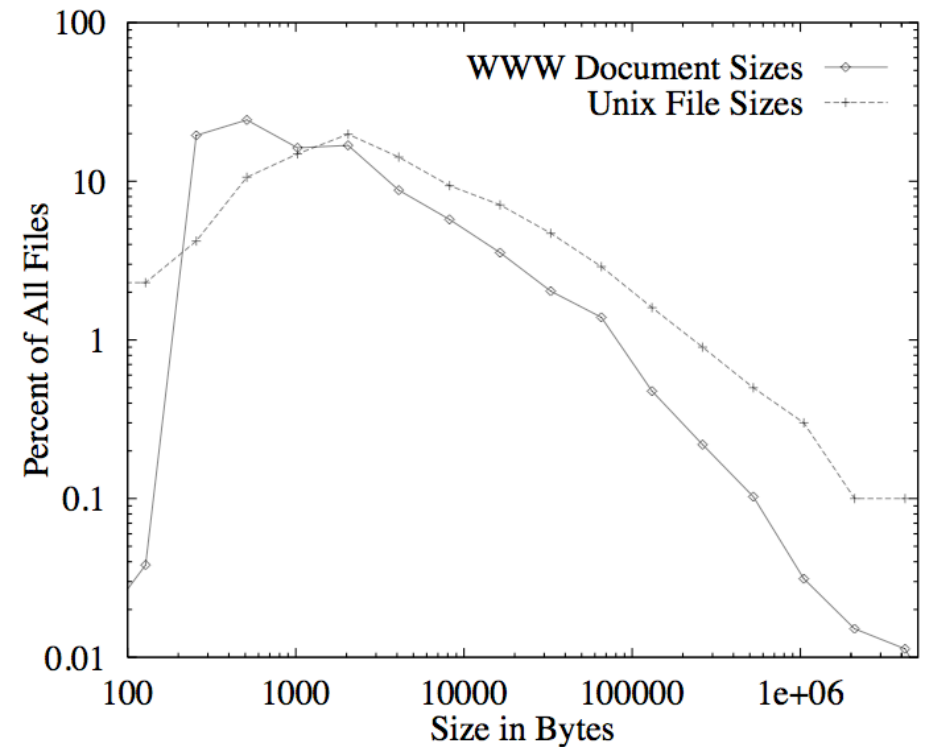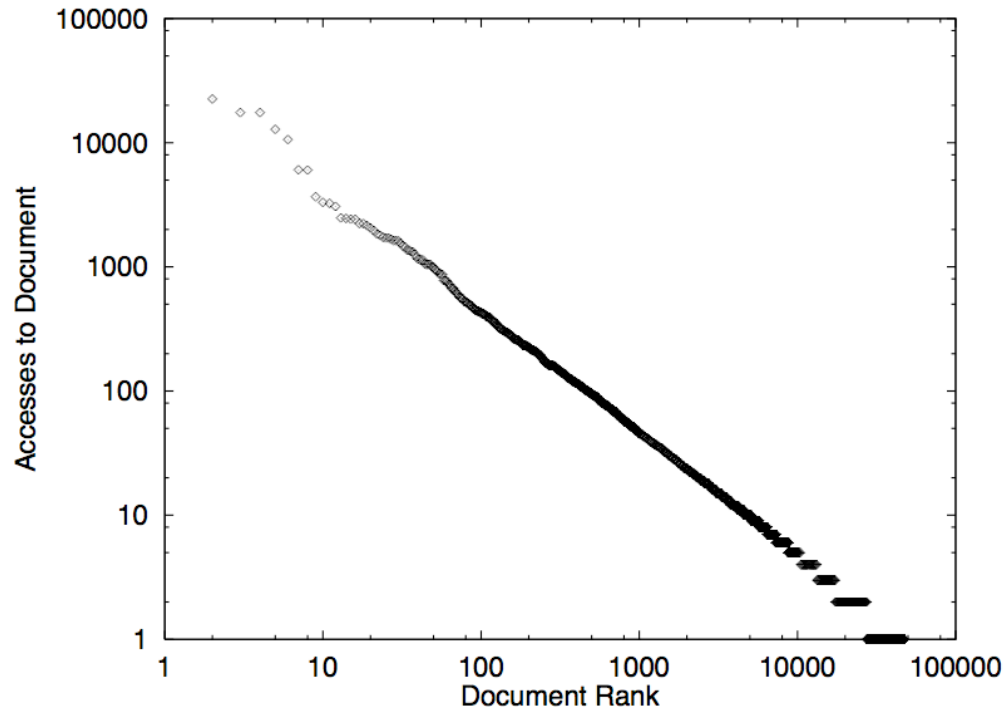**Figure 1.** Fitted power law distributions of the number of site a) pages, b) visitors, c) out links, and d) in links, measured in 1997.

Heavy-Tailed Probability Distributions in the World Wide Web
Mark Crovella, Murad, Taqqu, Azer Bestavros, 1996

Small World Phenomenon

- **Milgram's experiment 1967**
  - 60 random chosen participants in Wichita, Kansas had to send a packet to an unknown address
  - They were only allowed to send the packet to friends
    - likewise the friends

- **The majority of packtes arrived within six hops**

- **Small-World-Networks**
  - are networks with Pareto distributed node degree
  - with small diameter (i.e. $O(\log^c n)$)
  - and relatively many cliques

- **Small-World-Networks**
  - Internet, World-Wide-Web, nervous systems, social networks

How do Small World Networks Come into Existence?

- Emergence of scaling in random networks, Albert-Laszlo Barabasi, Reka Albert, 1999

- Preferential Attachment-Modell (Barabasi-Albert):
  - Starting from a small starting graph successively nodes are inserted with m edges each (m is a parameter)
  - The probability to choose an existing node as a neighbor is proportional to the current degree of a node

- This leads to a Pareto network with exponent 2,9 ± 0,1
  - however cliques are very seldom

- Watts-Strogatz (1998)
  - Start with a ring and connections to the m nearest neighbors
  - With probability p every edge is replaced with a random edge
  - Allows continuous transition from an ordered graph to chaos

- Extended by Kleinberg (1999) for the theoretical verification of Milgram's experiment

Analyzing Gnutella

- **Modeling Large-scale Peer-to-Peer Networks and a Case Study of Gnutella**
  - Mihajlo A. Jovanovic, 2001

Snapshot date	Nodes	Edges	Diameter
11/13/2000	992	2465	9
11/16/2000	1008	1782	12
12/20/2000	1077	4094	10
12/27/2000	1026	3752	8
12/28/2000	1125	4080	8

Analyzing Gnutella

- **Modeling Large-scale Peer-to-Peer Networks and a Case Study of Gnutella**
  - Mihajlo A. Jovanovic, 2001
- **Comparison of the characteristic path length**
  - mean distance between two nodes

	Gnutella	BA	WS	G(n,p)	2D mesh
11/13/2000	3.72299	3.47491	4.59706	4.48727	20.6667
11/16/2000	4.42593	4.07535	4.61155	5.5372	21.3333
12/20/2000	3.3065	3.19022	4.22492	3.6649	22
12/27/2000	3.30361	3.18046	4.19174	3.70995	21.3333
12/28/2000	3.32817	3.20749	4.25202	3.7688	22.6667

Topology-Management

- T-Man: Fast Gossip-based Construction of Large-Scale Overlay Topologies Mark Jelasity Ozalp Babaoglu, 1994

do at a random time once in each
consecutive interval of T time units
    $p \leftarrow$ selectPeer()
    myDescriptor $\leftarrow$ (myAddress,myProfile)
    buffer $\leftarrow$ merge(view,{myDescriptor})
    buffer $\leftarrow$ merge(buffer,rnd.view)
    send buffer to p
    receive $buffer_p$ from p
    buffer $\leftarrow$ merge($buffer_p$,view)
    view $\leftarrow$ selectView(buffer)

        (a) active thread

do forever
    receive $buffer_q$ from q
    myDescriptor $\leftarrow$ (myAddress,myprofile)
    buffer $\leftarrow$ merge(view,{myDescriptor})
    buffer $\leftarrow$ merge(buffer,rnd.view)
    send buffer to q
    buffer $\leftarrow$ merge($buffer_q$,view)
    view $\leftarrow$ selectView(buffer)

        (b) passive thread

Fig. 1. The T-MAN protocol.

Finding a Torus

after 3 cycles   after 5 cycles   after 8 cycles   after 15 cycles
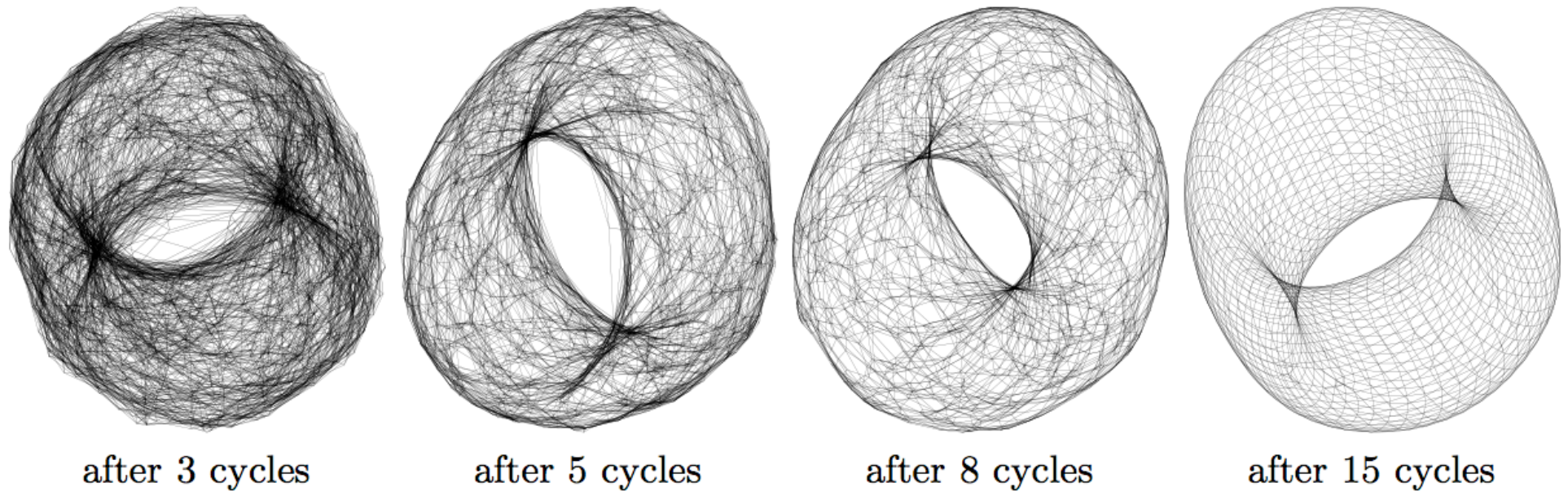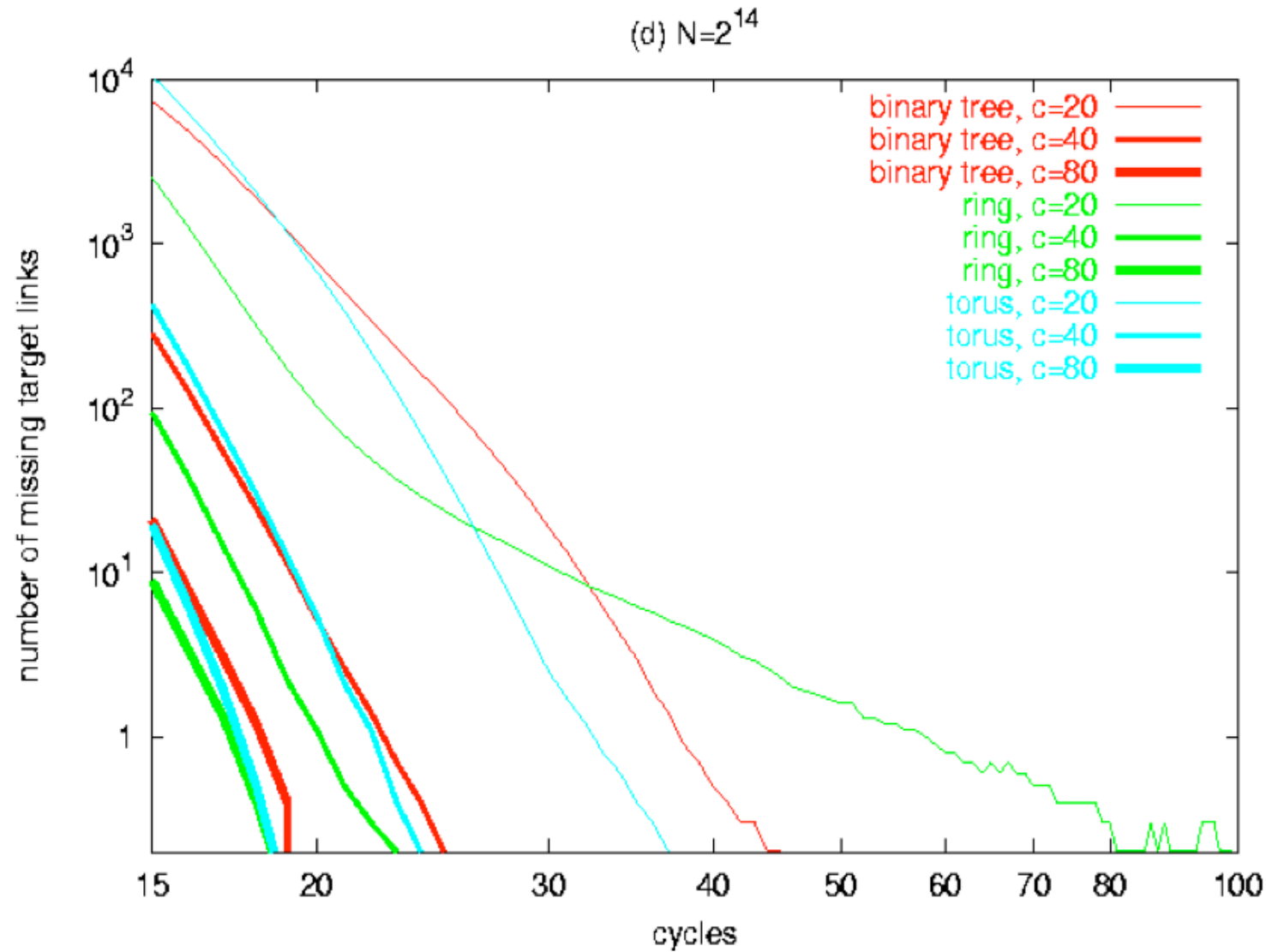
Fig. 2. Illustrative example of constructing a torus over $50 \times 50 = 2500$ nodes, starting from a uniform random topology with $c = 20$. For clarity, only the nearest 4 neighbors (out of 20) of each node are displayed.

Convergence of T-MAN

(d) N=2^{14}

Legend:
- binary tree, c=20
- binary tree, c=40
- binary tree, c=80
- ring, c=20
- ring, c=40
- ring, c=80
- torus, c=20
- torus, c=40
- torus, c=80

y-axis: number of missing target links
x-axis: cycles

T-Chord

- Chord on demand, A Montresor, M Jelasity, O Babaoglu - Peer-to-Peer Computing, 2005. P2P 2005.

Main Technique T-Man

The T-Man algorithm

// *view* is a collection of neighbors

Init: *view* = *rnd.view* $\cup$ { (*myaddress*, *mydescriptor*) }

// active thread
// executed by p
do once every
    δ time units
  q = selectNeighbor(*view*)
  msg_p = extract(*view*, *q*)
  send msg_p **to** *q*
  receive msg_q **from** *q*
  view = merge(*view*, msg_q)

A "round" of length δ

// passive thread
// executed by p
do forever
  receive msg_q from *
 msg_p = extract(*view*, *q*)
  send msg_p **to** *q*
  view = merge(*view*, msg_q)

9

Adaption for Chord
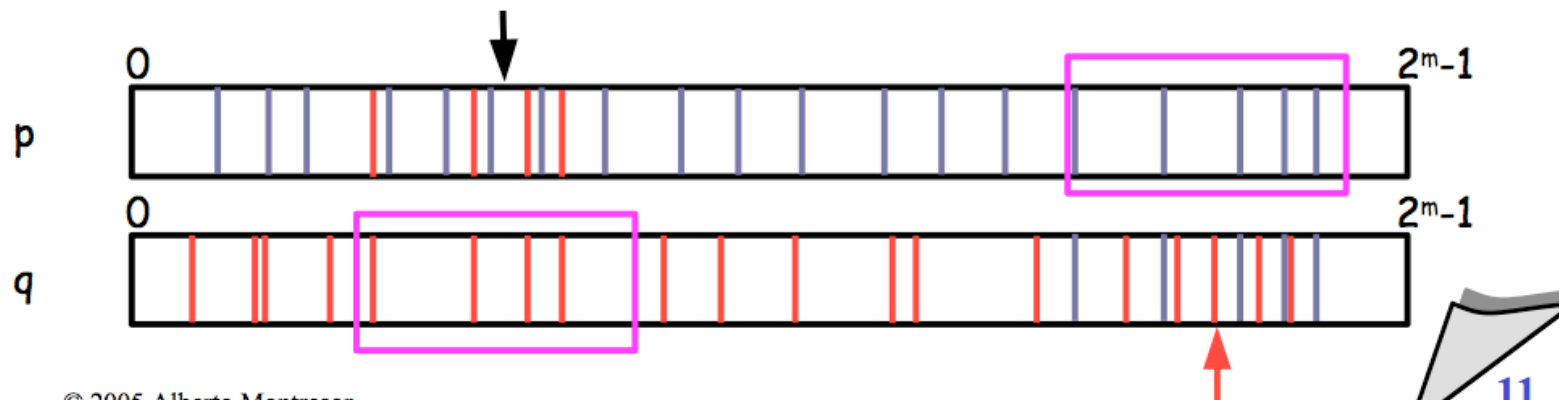
T-Man for T-Chord

- selectPeer():
    - randomly select a peer *q* from the *r* nodes in my view that are *nearest to* **p** *in terms of ID distance*

- extract():
    - send to *q* the *r* nodes in local view that are *nearest to* **q**
    - *q* responds with the *r* nodes in its view that are *nearest to* **p**

- merge():
    - both *p* and *q* *merge* the received nodes to their view

10
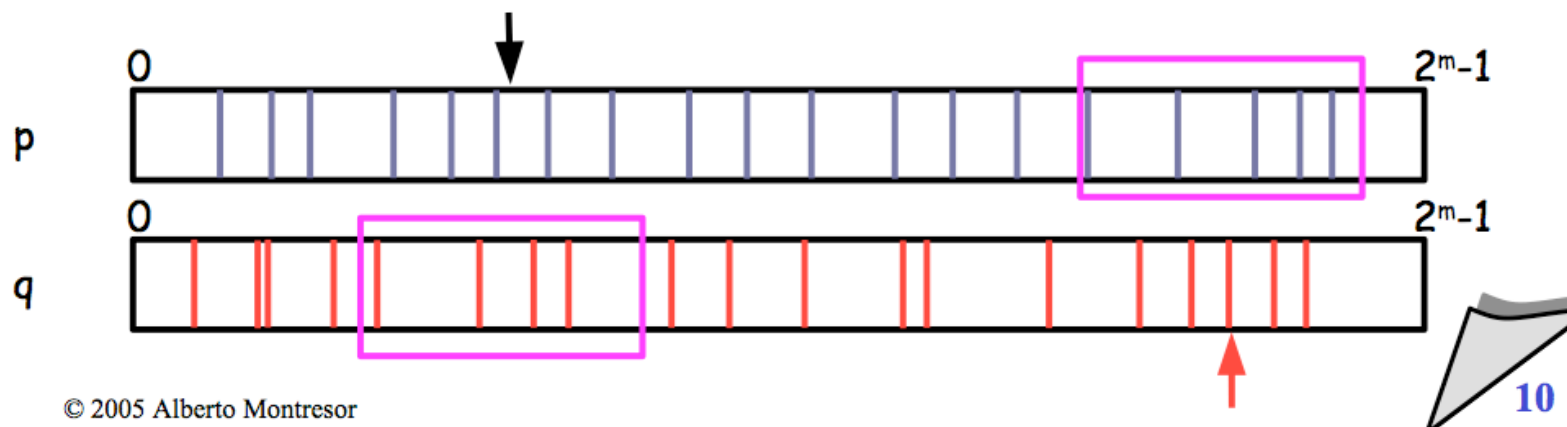
After Exchange of Links

T-Man for T-Chord

- selectPeer():
  - randomly select a peer **q** from the **r** nodes in my view that are *nearest to* **p** *in terms of ID distance*

- extract():
  - send to **q** the **r** nodes in local view that are *nearest to* **q**
  - **q** responds with the **r** nodes in its view that are *nearest to* **p**

- merge():
  - both **p** and **q** *merge* the received nodes to their view

11

23

Peer-to-Peer Networks
09 Self-Organization

Christian Schindelhauer

Technical Faculty

Computer-Networks and Telematics

University of Freiburg