Chair for Network Architectures and Services
Department of Informatics
TU München – Prof. Carle

Peer-to-Peer Systems and Security IN2194

Chapter 1 Peer-to-Peer Systems 1.2 Unstructured Systems

Dr. Heiko Niedermayer
Christian Grothoff, PhD
Prof. Dr.-Ing. Georg Carle

Overview

1.2a) Basics

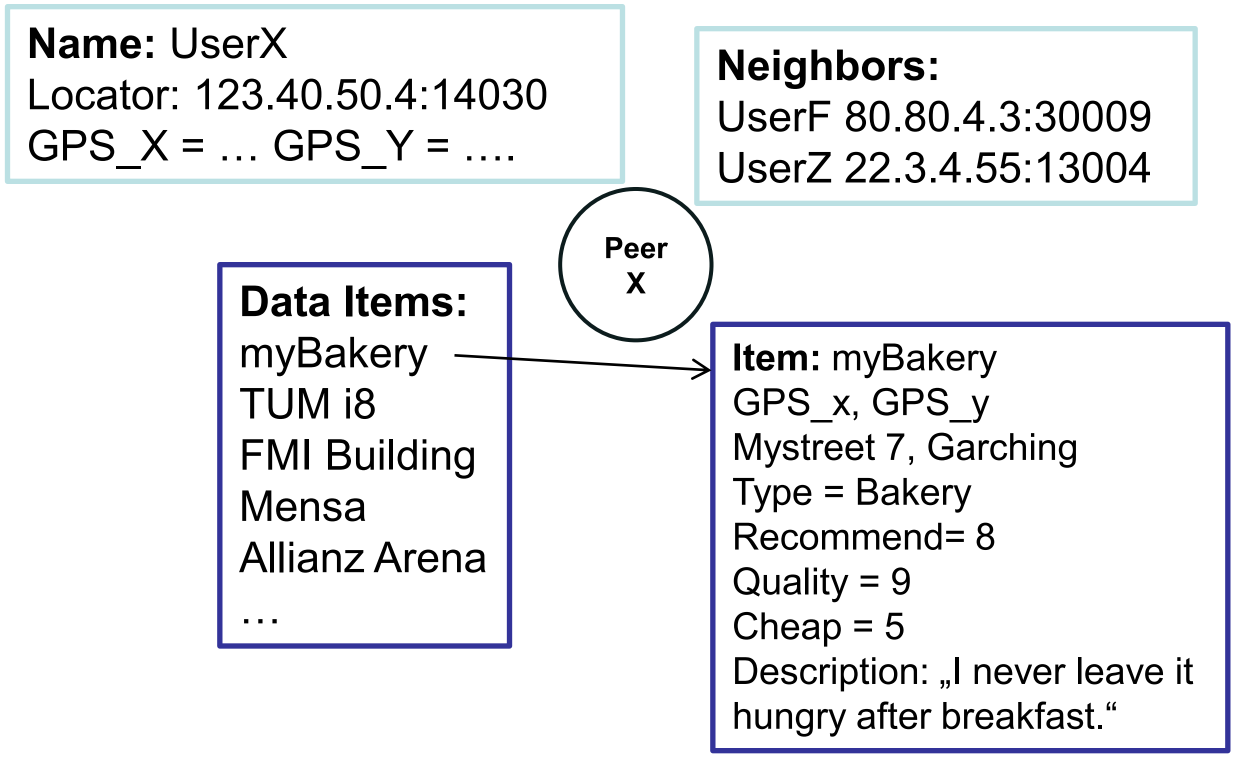
- „Unstructured“ / „Structured“
- Early unstructured Peer-to-Peer networks
  - Napster
  - Gnutella
- Theory
  - Random Graphs
  - Small World Theory
  - Scale-Free Graphs

1.2b) Systems

- Unstructured VoIP / IM Systems / Skype
- Swarming
  - BitTorrent
  - Mesh-Based Streaming

Example CoolSpotsMunich / one peer

- Our toy example CoolSpotsMunich as unstructured network
- A node in the CoolSpotsMunich network:



Name: UserX
Locator: 123.40.50.4:14030
GPS_X = … GPS_Y = ….

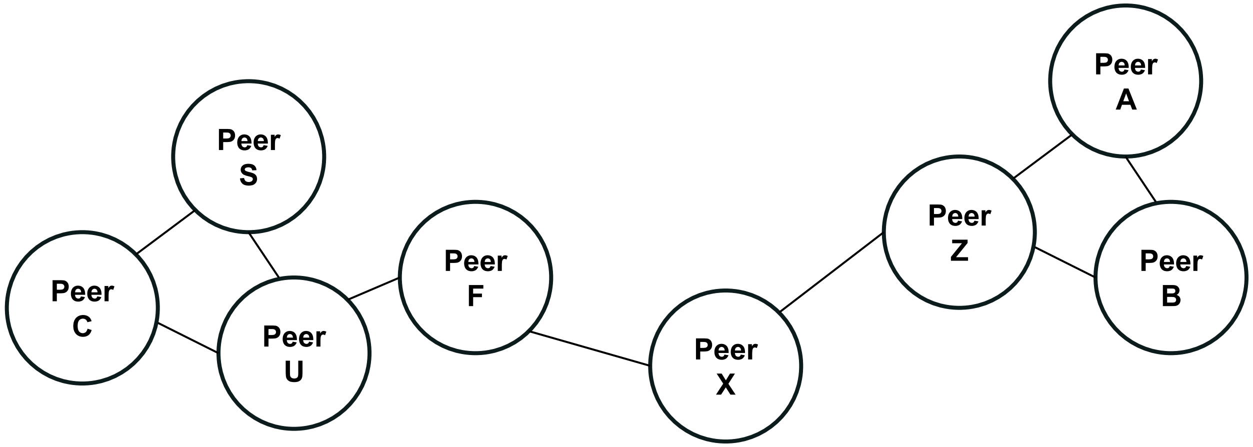
Neighbors:
UserF 80.80.4.3:30009
UserZ 22.3.4.55:13004

Peer X

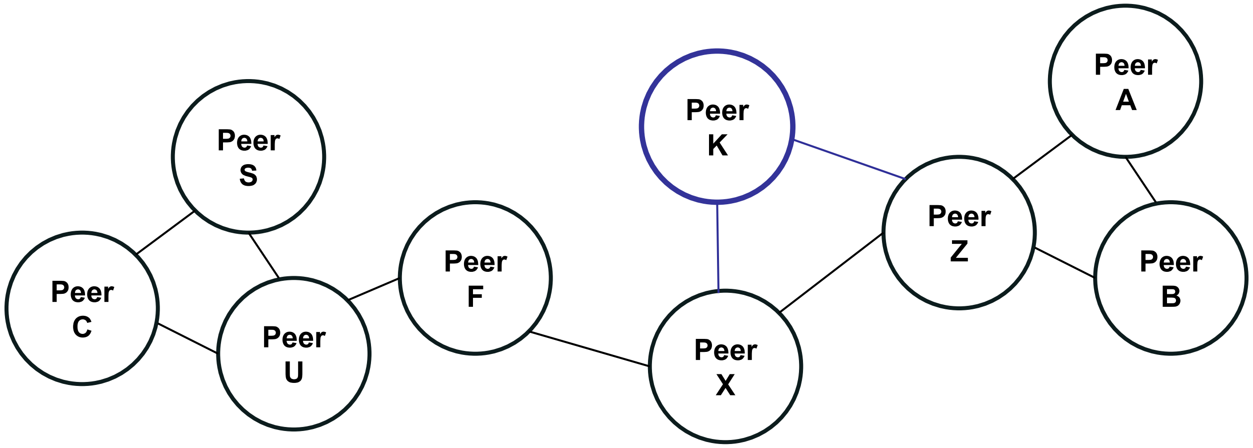
Data Items:
myBakery
TUM i8
FMI Building
Mensa
Allianz Arena
…

Item: myBakery
GPS_x, GPS_y
Mystreet 7, Garching
Type = Bakery
Recommend= 8
Quality = 9
Cheap = 5
Description: „I never leave it hungry after breakfast.“

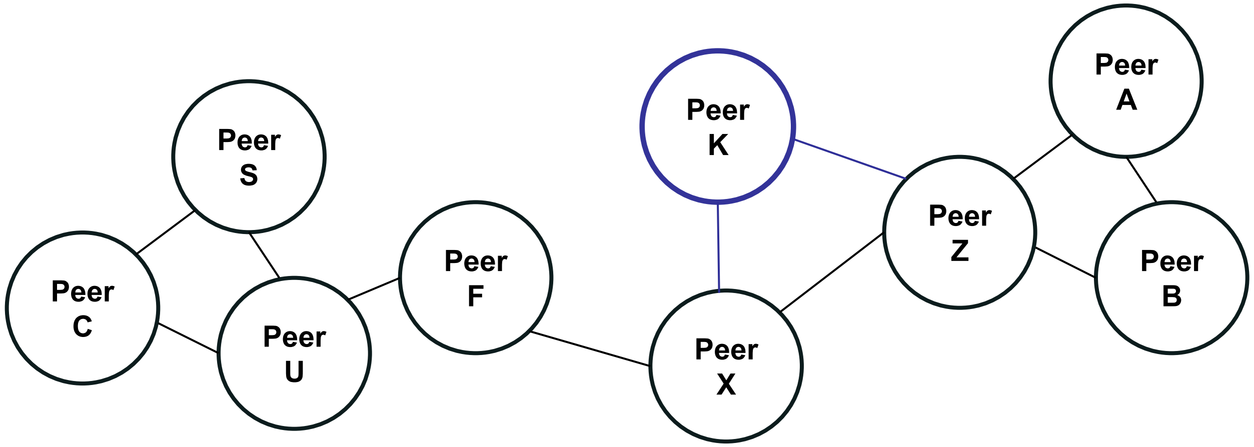
Example CoolSpotsMunich / network



- User K is a friend of User X and joins via Peer X because the friend told him his locator (IP:Port)
- Peer X tells him also some neighbors, Peer K contacts Peer Z



Example CoolSpotsMunich / request



- Now, User K is in Garching and he is hungry.

→ K asks X: Request for place to obtain food near GPS_X, GPS_Y not further than 2 km with recommend > 6

→ X tells him about myBakery and skips mensa (as recommend = 4)

→ K asks Z: Request for place to obtain food near GPS_X, GPS_Y not further than 2 km with recommend > 6

→ Z doesnt know anything in Garching, but proposes to ask A and B.

→ K also asks A, A tells him about the mensa

→ K also asks B, B tells him about food in FMI building

→ K terminates his requests (the user decides to go to myBakery)

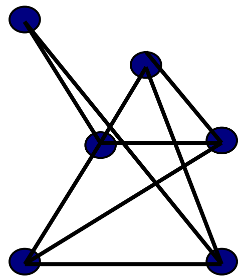
Unstructured / Structured

Unstructured / Structured

Unstructured / Structured

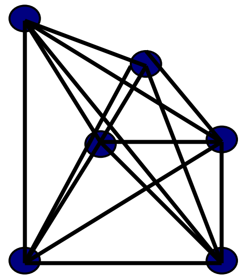
Unstructured Network

- Graph is created by random node interactions / user behaviour.
- Does not self-organize into a predefined structure.

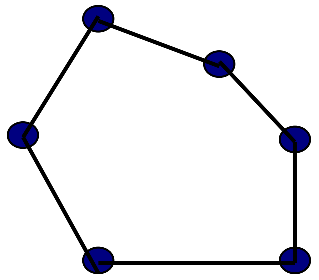


Examples for predefined structures

- Full Mesh / Clique
  - All nodes are connected with each other.
  - *n* nodes ➔ degree = *n-1*
  - Diameter = 1



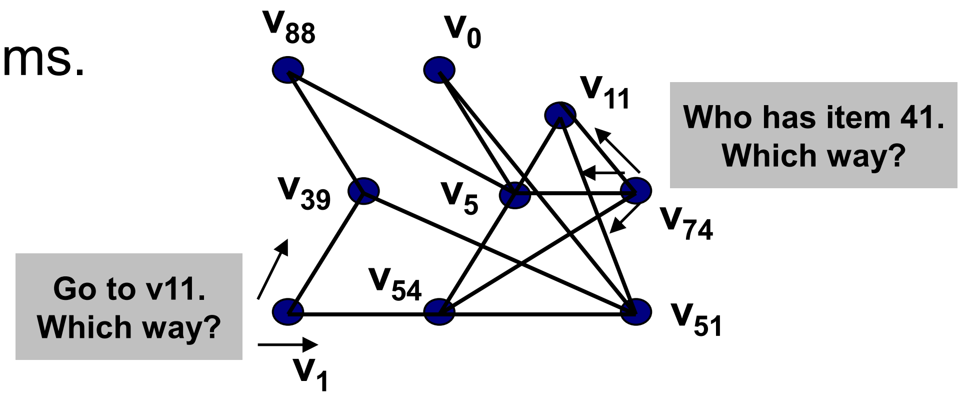
- Ring
  - Nodes organized in a ring
  - Degree = 2
  - *n* nodes ➔ diameter = n/2



Unstructured networks

Properties

- No structure has to be created and maintained whenever something changes in the network.
  - Join
    - Completed once the node is registered at one other node (except for the need of this node to get to know more nodes….)
  - Leave
    - No need to rework, but to locally remove the link
- Unless destination is known, there is no way to know where it is but to search all over the network.
- Nodes store their own items.



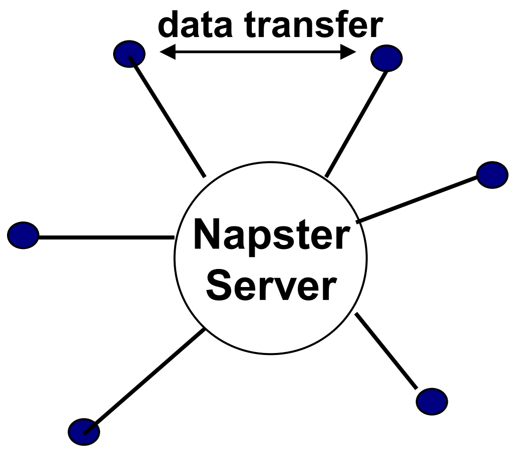
Early Unstructured P2P Systems

Early Unstructured P2P Systems

Napster

Napster

- A centralized Peer-to-Peer system
  - Centralized P2P = management and indexing done by central servers
- 1999 by Shawn Flemming (student at Northwestern University)
- Finally shut down in 2001 as result of law suits.
- Approach
  - Central Server
    - Manages index of files
  - Peers
    - Register to server with their shared files
    - Query server for files ➔list of Peers with their hits for the query
    - Download from Peer
  - Peer-to-Peer
    - Only the data exchange between the Peers



Filesharing

Filesharing

- ❑ Share and announce content
- ❑ Search for content
- ❑ Download content

Problems

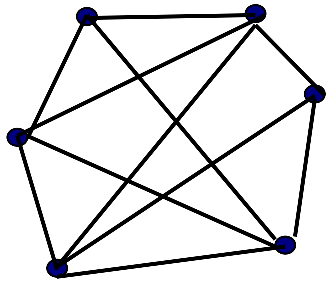
- ❑ Legal issues (see Napster) → Decentralization
- ❑ How to find content?
  - ▪ String queries
    - • Substring
  - ▪ Fuzzy queries
  - ▪ Usually no exact queries

  → Thus, the task for the unstructured decentralized network is to search the network for hits.

Gnutella

Gnutella 0.4

- Pure Peer-to-Peer approach
  - No central entities like in Napster.
  - Avoid single points of failure, any peer can be removed without loss of functionality.
- Join
  - Via any node in the network
    - Taken from downloaded host list, peer cache, …
    - Receives a list of recently active peers from this node.
  - Explore neighborhood with ping/pong messages.
  - Establish connections until a quota is reached.
- Limited flooding as routing principle
  - Flood message to neighbors unless TTL of message exceeded.
  - Store the source of these messages to be able to return the hit to the source (= previous node, not the original source of the request).



Gnutella

Basic primitives of Gnutella 0.4

- Ping / pong: discover neighborhood
- Query / query hit: discover content
- Push: download request sent to firewalled nodes
  - Firewalls may only allow connections to be established from inside to the Internet and not the other way around.
  - The firewall and NAT aspects of Peer-to-Peer are discussed in a later section.

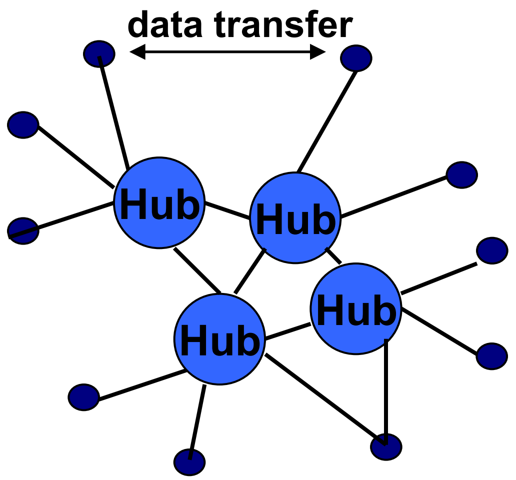
Properties

- Immense bandwidth consumption due to flooding for the signalling and unsuccessful search traffic!
  - Gnutella 0.4 does not scale (~ overhead dominates the network).
- Provides a weak form of anonymity as query is without source address and hits are returned hop-by-hop on the path.

Gnutella2

Gnutella2

- Hybrid Peer-to-Peer approach
  - Distinction between client peers and super peers
    - Super peers form unstructured network
    - Client peers connect to some super peers
- Hubs (super peers)
  - Accept hundreds of leaves (client peers)
  - Many connections to other hubs
  - Query Hit Table
    - List of files provided by its leaves.
- Leaves (client peers)
  - Each leaf connects to one or two hubs.
- Search
  - Gather a list of hubs and iteratively ask them.
- Properties
  - Less traffic overhead, scales better



Theory

Theory

Theory

Observation

- Graphs of unstructured networks are created by random and social interactions.
  - Randomness
  - Social aspects (social network, entry points, uptime, …)
  - Content (interesting files, …)

Questions

- What is their form?
- Are they good?

In the following we present some theoretic graph models that are used to approximate these graphs and their properties.

Theory: Random Graphs

Randomly-created Graphs

- Way to model the structure of these networks
- Necessary to understand the behaviour of these networks

Random Graphs / Uniform Random Graphs

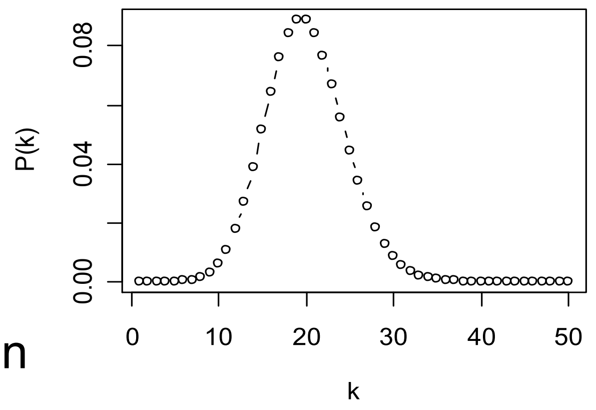
- Graph G = (V,E)
  - E is created randomly
  - n = |V|, m = |E|
- Assumption
  - Nodes randomly connect to each other.
- We will also call them uniform random graphs to distinguish them from other graphs that are also randomly-created, but where nodes are not all equal and strategies bias the link selection.
- Average distance in random graphs is most likely to be close to optimal for given n and m.

Erdös-Rényi model

Uniform random graphs according to Erdös-Rényi model (1960)

- Given:
  - n nodes und probability p
- Construction:
  - **For** any two nodes v1, v2 **do** with probability p: connect(v1,v2)
- Resulting graph:
  - E[|E|] = p * n^2 / 2
  - The node degree follows the binomial distribution (approx. by Poisson distribution for large n).
- Discussion:
  - Too simple and uniform for a model of real networks.

Degree distribution for n=50, p=0.4



The Small-World Phenomenon

- We meet someone we know at a place where we do not expect something like that to happen. ➔ What a small world ?!?

An experiment by Stanley Milgram (1960s)

- Milgram sent mail to people in Nebraska.
- The mail should only be sent to people they personally know who might know better how to reach to the targeted receiver.
- The targeted receivers of the mails were people from Boston.
- The result was that on average six hops were required and that the median was below six.
- Subsequently, this lead to the term ”‘Six degrees of separation”’ and the conclusion that we live in ”‘small world”’.

Discussion of the Milgram experiment

- First of all, ”‘six degrees of separation”’ sounds more like a maximum, but it is an average and the maximum, say the diameter of the graph, may be significantly larger.
- Judith Kleinfeld [Klei02] looked into the experiments of Milgram in more detail.
  - Most of Milgram’s messages did not find their receiver. In fact, the success rate (chain completion rate) was below 20 %.
  - The people that were selected were also biased in such a way that well-off higher-ranked people were preferred. Moreover, even six degrees may be a strong barrier in reality, say a big world, that cannot be bridged in particular among different races and classes.
  - ***A big world afterall….?***

Graph measure: Characteristic path length (L)

In the following, we introduce two scalar properties that can be used to characterize graphs.

Characteristic path length (L)

- L corresponds to the average length of a shortest path in an undirected graph

$$L = \operatorname*{avg}_{i,j \in V, i \neq j} d(i,j) = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n} \sum_{j=i+1}^{n} d(i,j)$$

- Recap of the definition of the diameter

$$D = \max_{i,j \in V, i \neq j} d(i,j)$$

- L and random graphs (e.g. constructed by Erdös-Rényi model)

$$L_{random} \sim \frac{\log n}{\log(m/n)}$$

Graph measure: Clustering coefficient (C)

Cluster

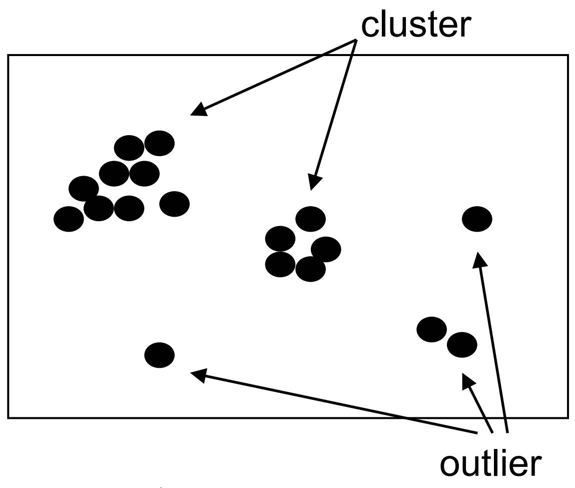
- engl. Traube, Bündel, Schwarm, Haufen
- In data analysis points with similar properties.

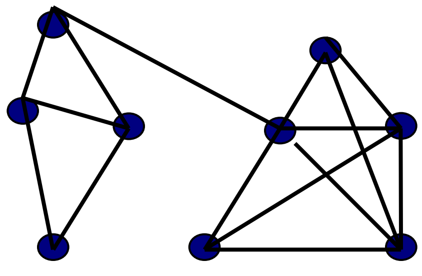
Clustering in networking

- Here, a group of nodes that are all closely connected.
- An informal notion of a cluster is that nodes in a cluster are close to each other. So, most neighbors of a node in a cluster are also close or even neighbors of each other.

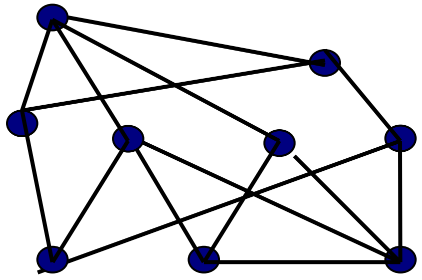
➔ *„When my friends are also friends, we are a cluster.“*

We will use this idea to define a measure called clustering coefficient.





Graph with 2 clusters



Rather unclustered

Graph measure: Clustering coefficient

Clustering coefficient C

- Given graph G = (V,E)
- We define the neighborhood of a vertex v

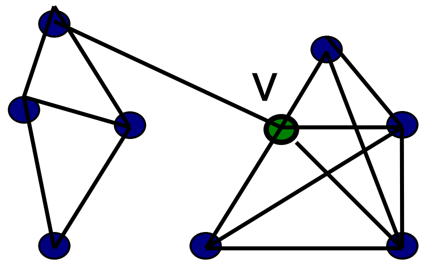
$$\Gamma_v = \{u \in V \mid u \quad adjacent \quad to \quad v\}$$

- Given U as subset of V, we define E(U) the edges of the subgraph of V spanned with the nodes U.
- Local clustering coefficient of node v

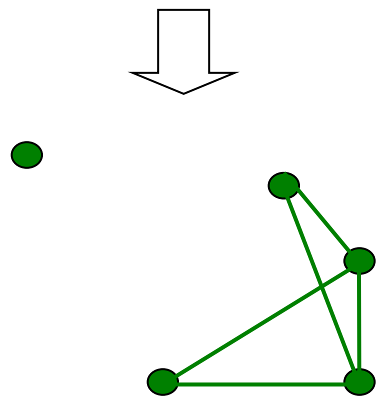
$$C_v = \frac{\#edges\_of\_subgraph\_G(\Gamma_v, E(\Gamma_v))}{\#all\_possible\_edges\_between\_nodes\_\Gamma_v} = \frac{|E(\Gamma_v)|}{\binom{\deg ree(v)}{2}}$$

- **Clustering coefficient C of G**

$$C = \frac{1}{n}\sum_{v \in V} C_v = \frac{1}{n}\sum_{v \in V} \frac{|E(\Gamma_v)|}{\binom{\deg ree(v)}{2}}$$



G(V,E)



G(U,E(U))

Examples – Clustering coefficient

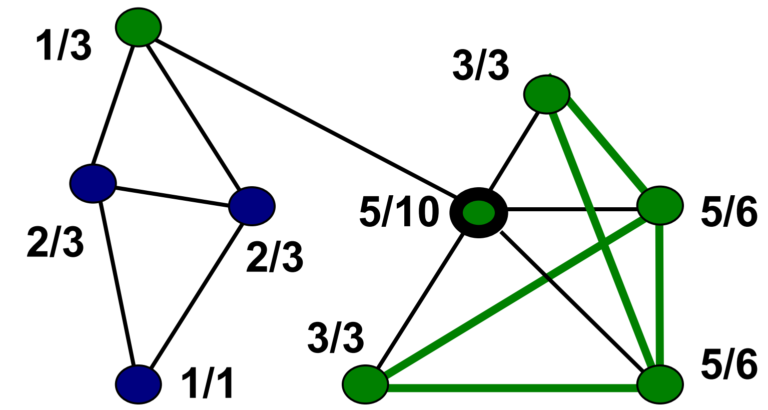
Calculation

$$\binom{2}{2} = \frac{2*1}{2*1} = 1 \qquad \binom{4}{2} = \frac{4*3}{2*1} = 6$$
$$\binom{3}{2} = \frac{3*2}{2*1} = 3 \qquad \binom{5}{2} = \frac{5*4}{2*1} = 10$$

- The clustering coefficient $C = \frac{1}{n}\sum_{v \in V} \frac{|E(\Gamma_v)|}{\binom{\deg ree(v)}{2}}$

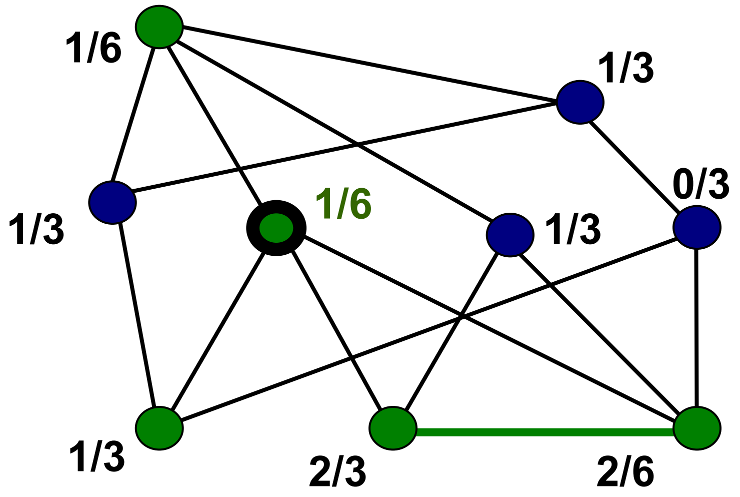
The graph with 2 clusters

- As example we compute the local ● clustering coefficient of a rather central node
  - It has 5 neighbors ● .
  - Their graph has 5 edges / of 10 possible edges.
  - Thus, its coefficient is 5/10 = 0,5.
- The coefficent of the graph **C = 0,759**



The rather unclustered graph

- The example node has 4 neighbors that share only one edge. Its local clustering coefficient is 1/6 = 0,167.
- The coefficient of the graph **C = 0,296**



The Small-World Phenomenon in P2P Networking

Small-World Graph

- A Small-World graph is a graph with a characteristic path length close to that of an equivalent uniform random graph ($L \approx L_{random}$), but with a cluster coefficient much greater ($C >> C_{random}$).

Small-World on the Internet and elsewhere

	Size	Avg. degree	L	L_random	C	C_random
Internet graph (2002) Skitter topology (***)	260.000	3.39	11.4	10.1	0.023	0.000014
Gnutella (2000) Snapshot (**)	n/a	n/a	3.86	3.19	0.045	0.0068
Film collaboration (*)	225000	61	3.65	2.99	0.79	0.00027
Power Grid (*)	4900	2.67	18.7	12.4	0.080	0.005
Neural network of worm C.elegans (*)	282	14	2.65	2.25	0.28	0.05

(*) Watts & Strogatz 1999 (**) Li et. al 2004, (***) Jin & Bestavros 2006

Small-World-Theory and real networks

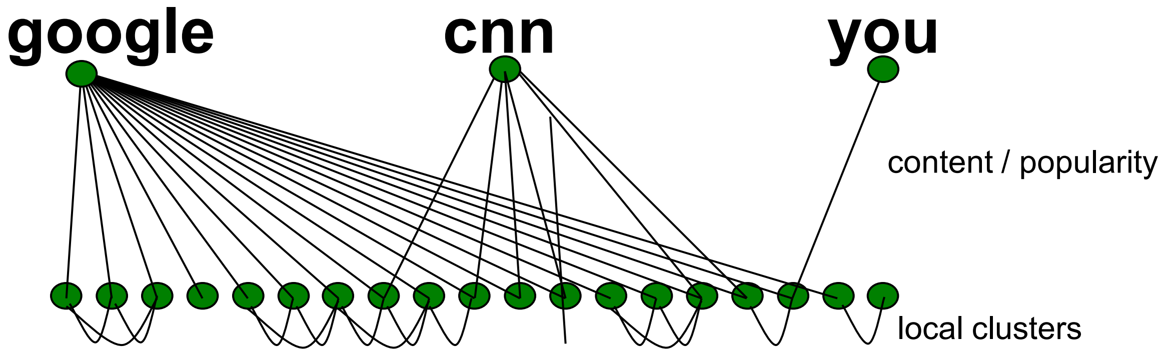
Real networks and Small-World networks

- Real networks (WWW, Gnutella, etc.) often show Small-World properties
  - Characteristic path length is small
  - Clustering coefficient is high

….but…. unlike Small-World networks, they are

- not symmetric
- the peers are way from being equally used.
- In fact, the popularity and the degree of nodes differs extremely .
  - E.g. compare google.com, cnn.com and your webpage.

Zipf‘s Law and Scale-Free networks



Zipf's law: "The popularity of **i**th-most popular object is proportional to $i^{-\alpha}$, α: Zipf coefficient."

- Zipf-like popularity can be found for websites, words in natural languages, movies, …

Node degrees in the example

- Google 18, CNN 6, you 1, other nodes 1-5
- In Filesharing replace the websites with popular content.
- Small-World theory does not explain and contain this variation.

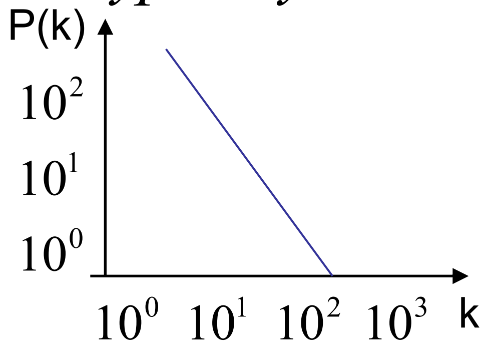
→ Next model: Scale-Free networks

Scale-Free Networks

Scale-Free networks / Power-Law networks

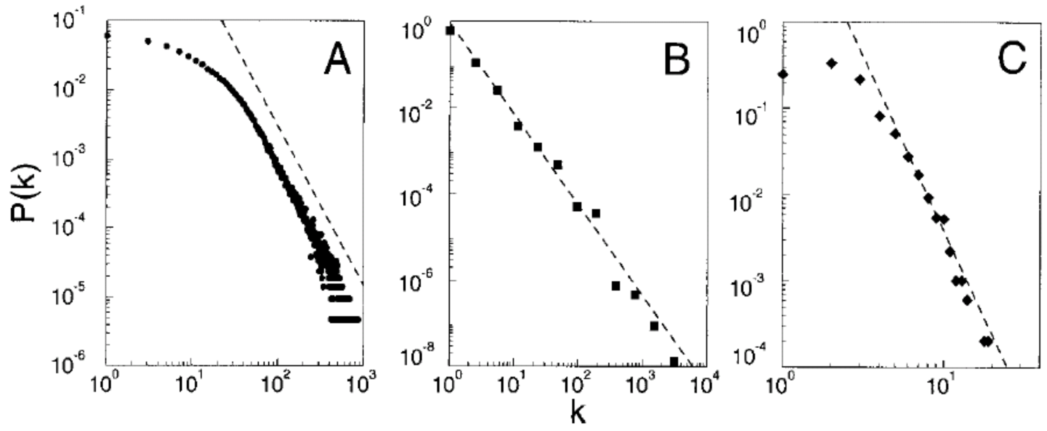
- The term scale-free relates to the fact that the degree distribution is independent of any scale (e.g. no size of the network in it).
- Power Law distribution of the node degree

$$P(k) \sim k^{-\gamma} \quad \textit{typically} \quad \textit{with} \quad \gamma \approx 3 \pm 1$$



- Other definitions for Scale-Free graphs can be found.
- Scale-Free graphs are a likely outcome of random graph construction processes that contain some element with high variability.
  (More on the topic: Li, Alerderson, Tanaka, Doyle, Willinger: „Towards a Theory of Scale-Free Graphs“, 2005)

Scale-Free Networks



Degree distribution for the Actor, WWW, and Power Grid networks taken from

Albert-Laszlo Barabasi and Reka Albert „Emerging of Scaling in Random Network“, Science 1999.

Properties

- The Power Law distribution has extremely high variability.
- A consequence of the extreme variation of node degree, is an existence of few high-degree nodes. Typically, they are called hubs.
  - The hubs are hotspots.
  - Failure or leave of hubs is a problem for these networks („Archilles heel“)
  - Failure of non-hubs is considered less problematic. Unless a hub is hit random failures hardly have an impact on the network, say on the average path length.

High variability / Power-Law distribution

In many fields of networking, there is an element of high variability.

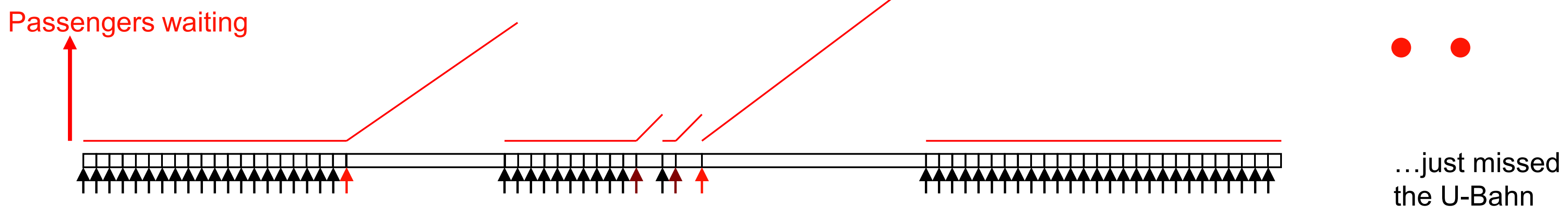
- e.g. network traffic, degree distribution, peer lifetime distribution,…
- High variability („heavy-tail“) means variation coefficient >> 1
  - The values vary more than their mean.

Example

```
1 1 2 45 1 0 1 1023 3 1 2 4 0 1 0 11 …
```

„Bus stop paradox“ (time between buses with variation >> 1)

- „Passenger is happy when she just misses a bus.“



- In most cases when the bus just left the bus stop (arrow), a bus will come within short time (black arrows).
- In most cases when a lot of people are waiting, the arrival of the next bus will still take a while.

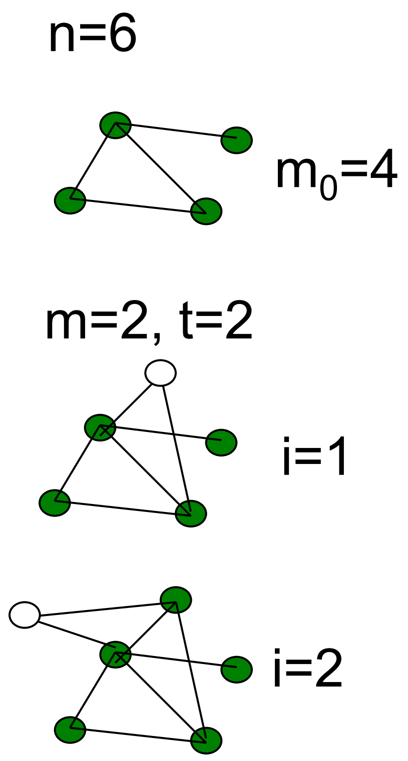
Scale-Free networks (Barabasi-Albert model)

Scale-Free graphs according to Barabasi-Albert's „The rich get richer" model (1999)

- Also called cumulative advantage.
- Given: n nodes
- Start with m_0 unconnected nodes, add random link for each node
  - Minimum degree of each node is 1.
- For i = 1 to t do
  - Add node, connect node to m nodes, select nodes according to the following distribution („linear preferential attachment")

$$p(i) = \frac{k_i}{\sum_{Available\_nodes\ j} k_j}$$

- Result
  - Graph with $t*m+m_0$ edges



Unstructed VoIP / IM Systems / Skype

Unstructed VoIP / IM Systems / Skype

Application: VoIP / IM

Voice over IP / Instant Messaging

- User accounts
- User management
- Search for users
- Keep contact and status with group of users („friends“)
- Start Voice or IM sessions with 2 or more participants

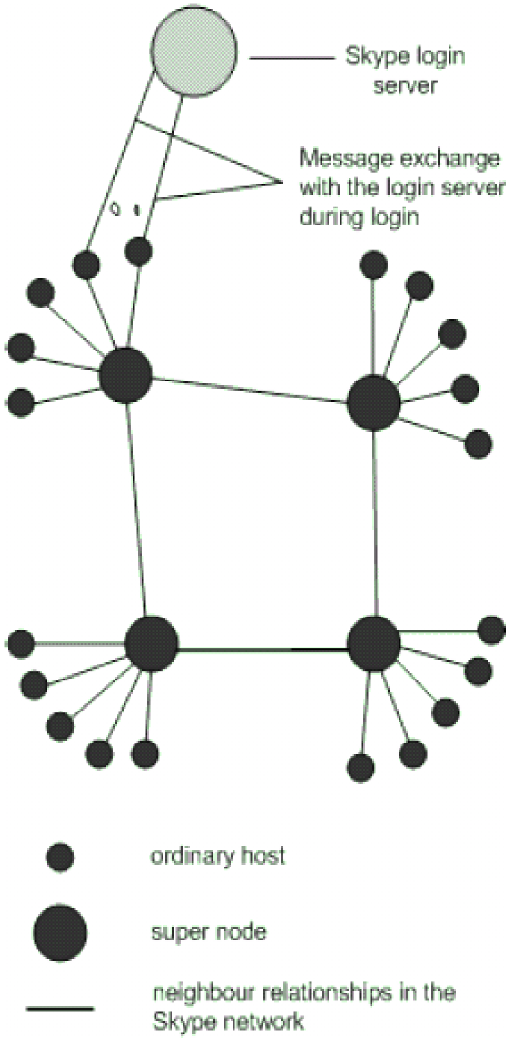
Popular

- Centralized systems like ICQ, AIM, …
- SIP-based or H.323 systems like Netmeeting, …
- **Skype**
  - Code and design of Skype is not published, all information presented is based on analysis of various researchers.
    - First studies network-oriented, by Baset and Schulzrinne 2004.
    - Reverse engineered, Biondi and Desclaux 2006. → also found way to induce a heap overflow.
  - Skype is secured using AES/RSA, closed-source with lots of anti-debugging tricks and obfuscation, and central login servers.

Skype

Skype

- Proprietary Protocol
  - *Protocol may change over time.*
  - *Slides based on analyses by Baset, Schulzrinne, Biondi, Desclaux (version 0.4).*
- Network Structure
  - Login server (manages accounts, login via Username/Password)
  - Supernodes (normal hosts with good connection)
  - Ordinary hosts
- Building up the Connection
  1) HTTP Get
     – Latest Version Check or Installation Notification
  2) Connect to a host in the host cache (HC)
     - Initial bootstrap list hard coded, bootstrap peers provide more hosts for HC
     - Check for Firewall/NAT (variant of STUN protocol)
  3) Connect Login server
     – Login with Username/Password
  4) Exchange message with ~ 20 other Skype nodes
     – For Robustness?


Skype login server
Message exchange with the login server during login
ordinary host
super node
neighbour relationships in the Skype network

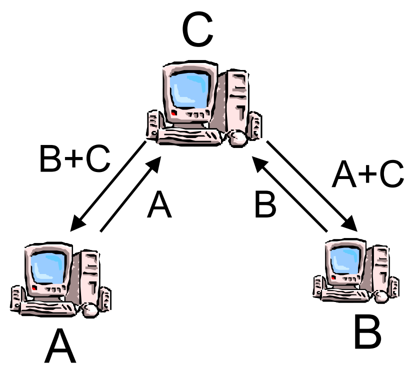
[Baset and Schulzrinne, 2004]

Skype

- Finding other people on the network
  - Way of Searching:
    - First request to the connected supernode
    - If not found: second request to 8 other supernodes
    - If not found: 3rd request to 16 other supernodes
    - ...
  - On average: connect to 24 nodes, 3-4 seconds searching time
  - Option of last resort: ask the login server
  - Behind NAT: Supernode does the search
  - User information caching on the supernodes

Skype

- Media transfer and Codecs
  - Up and downlink bandwith around 40 kbps, Packet size 40-120 Bytes/packet
  - Wide Band Codec (50-8000Hz)
  - UDP preferred for media transport
  - No silence suppression, constant rates
- Conference Calls
  - No full mesh conferencing for three party conference.
  - Most powerful machine is elected host.
- Supernodes (SNs)
  - Normal clients on good connections
  - No way of saying „I don't like to be a supernode“
  - Node sends keep alive to SN every 120s



Skype

Security Aspects of Skype

- Authentication
  - The Certificate Authority public key is a 2048bits key of RSA type.
  - SC session keys: 1024bit RSA key pair and 256-bit AES.
  - An encrypted MD5 hash of login/password provides user authentication and the use of a trusted Skype key identifies the Skype software.
  - If the central authority accepts the login, it signs the couple identity/public key.
- Code
  - Does not start if debugger is running.
  - Several techniques to prevent analysis, lots of dummy code, dynamic calculation of jumps, etc.
  - Binary is encrypted and permanently checks its integrity.
  - All the tricks make it hard to check for correctness → buffer/heap overflows reported.
- General discussion
  - Lots of data exchange with other nodes. Hard to determine if communication is good or bad, say transfering personal data.
  - In February 2007 it was discovered that Skype copied an executable called 1.com in the temp directory of the user which is used to read BIOS data of the PC. Most likely, this was used to bind the use of commercial Skype modules to particular PCs.

Swarming / BitTorrent / Mesh-Based Streaming

Swarming / BitTorrent / Mesh-Based Streaming

Swarm Intelligence

Swarm Intelligence (SI)

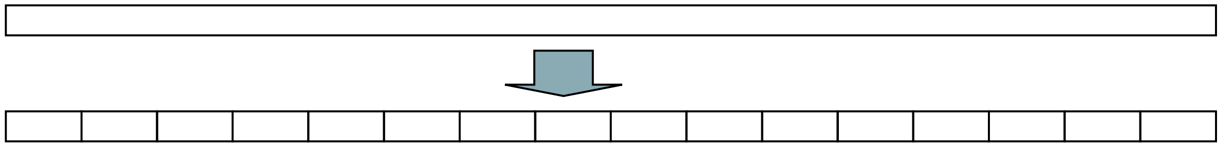
- Beni and Wang introduced the term in 1989 as a form of artificial intelligence.
- Idea
  - Use collective behavior of simple agents in decentralized, self-organizing systems for solving complex tasks.
- From a networking perspective
  - A group of decentralized networked entities cooperates in order to provide a service.
  - Decentralized and self-organizing → Peer-to-Peer

….so, what about swarms in P2P networking?

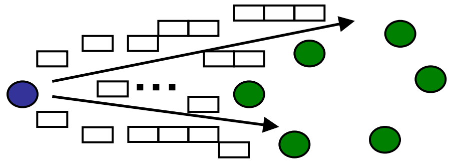
Another unstructured P2P concept – Swarming

Swarming

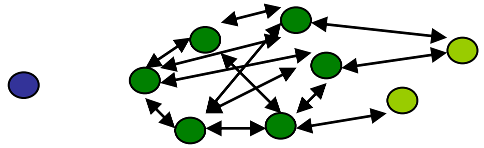
- Idea
  - Peers with the same interest form a swarm and cooperate, instead of individually getting each the same service from one responsible peer or server. → *Goal: a better service*
- For File Transfer / File Distribution
  - Split the file into small chunks. Identify the chunks in some way.



  - Send the chunks to the swarm (= group interested in the file).

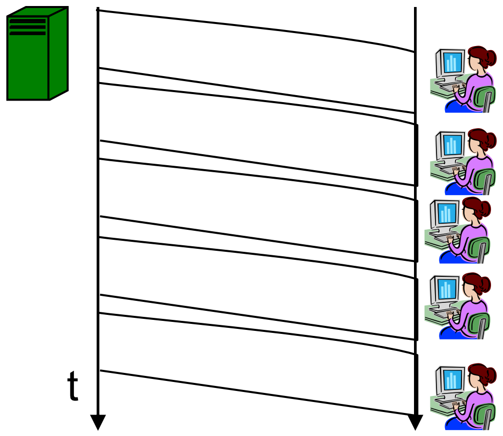


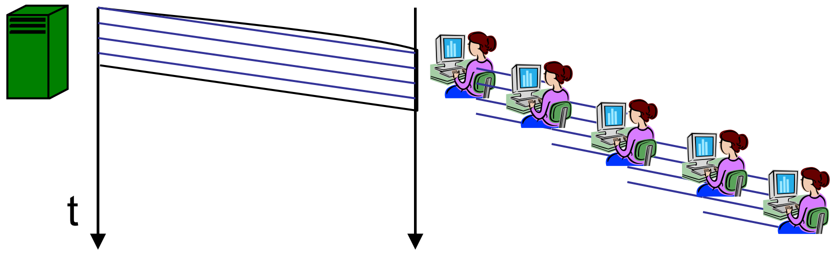
  - Members of the swarm download the chunks from each other in parallel. New nodes join and get the data from the swarm.



File Distribution – Swarming vs Server

- Server approach
  - n clients download complete file from server (sequentially or in parallel)
  - n * filesize of data transfered from server to the clients
  - Time needed to complete > n * filesize / datarate
- Swarming
  - n clients want to download a file
  - 1 client downloads chunks from the server (or more clients do this in parallel)
  - Other chunks are shared between the clients.
  - Time needed approaches filesize / datarate in the limit (simplified and idealized analysis).





BitTorrent

BitTorrent

- … was introduced by Bram Cohen in 2003.
- Goal
  - Efficient and scalable (the more users the better for the throughput) replication and distribution of large amounts of data.
- Approach
  - Swarming approach, typically file is split into 256 kB chunks („pieces“)
    - Chunks are split into 16 KB subpieces for the data transfer.
  - A seed initially creates a torrent for a file, it needs to have the complete file.
  - A metafile (.torrent) is distributed via some out-of-band mechanism, e.g. HTTP
  - A central entitiy (tracker) manages list with the current peers of the swarm.
  - The data transfer is done within the swarm without central coordination.

BitTorrent – Entities

Seed

- Peer that initially created the torrent.

Seeder

- Peer that has the complete file with all pieces.

Leecher

- Peer that is still downloading the file, does not have all pieces.

Tracker

- Central component that keeps track of all members of the swarm.
  - Only knows info hash (= which torrent), leechers and seeders
- Returns random list of nodes in the swarm.
  - Resulting in a random graph.

.torrent file

- Contains filename, size, SHA-1 hashes for all pieces, URL for tracker

BitTorrent – Piece Selection Strategies

- Each downloader reports to the other peer what pieces it has.

→ Local decision for each node: which piece is next?

Next Chunk Selection

- Random First Chunk
  - Select random chunk when you first start to download.
- Rarest First
  - Select to download the chunk that is the rarest among your neighbors.
  - Especially important if original seed is down and only leechers exist.

Next Subpiece Selection

- Strict Priority
  - Download subpieces of current chunk first → complete chunk first before requesting new chunk.
- Endgame mode
  - Ask all peers for subpieces of last chunk. Cancel requests if chunk is finished. (only happens for a short period of time at the end)

Tit-for-Tat / Choking

Resource Allocation

- Peers need to decide how much they send to other peers.
- Each peer is responsible for maximizing its download rate.
- Basic approach
  - Tit-for-Tat: If you are good to me, I am good to you.
  - Evaluate the connections every 10s → Choking and unchoking.
- Choking
  - Temporary refusal to upload to a peer, so that bandwidth can be used for (TCP connections to) other peers.
- Unchoking
  - Every 10s four choked peers are unchoked, usually depending on the download rate.
  - Optimistic unchoking: every third period, one peer is unchoked independent from the download rate.
- Problems
  - Optimistic unchoking can be misused if a client connects to a large enough amount of other peers (only in large torrents).

Trackerless BitTorrent

Trackers

- Single Points of Failures
- Have limitations in bandwidth

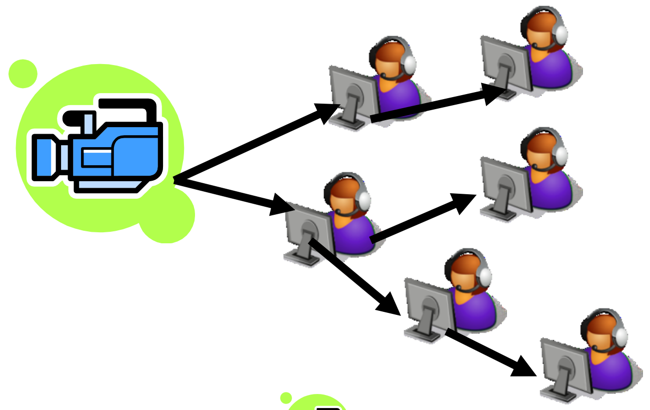
Trackerless BitTorrent

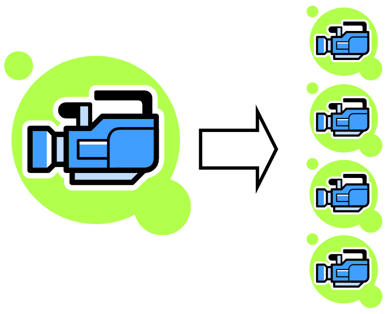
- File transfer: swarming as in tracker-based BitTorrent.
- Tracker is replaced with a Peer-to-Peer network.
  - Structured network (Kademlia) with routing to the torrent ID (info-hash).
  - 8 peers (replica set) can be found via the torrent ID and operate as tracker.
- Join to a torrent
  - A node announces its existences to the replica set of the torrent.
- Get the peer-list
  - Lookup for the torrent ID. In Kademlia the first peer found with a current list will return this list.

Swarming for Video Streaming

Video Streaming

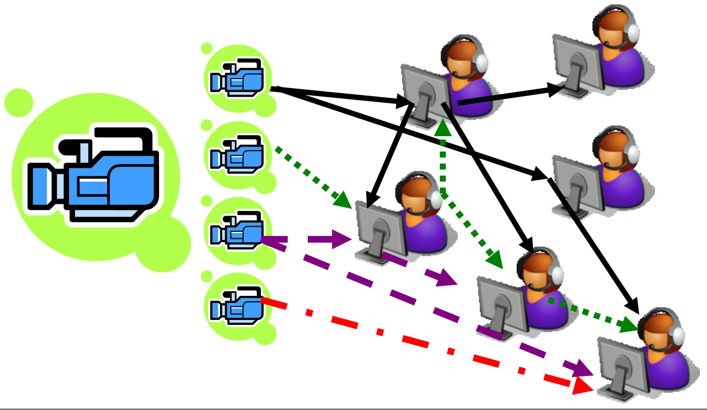
- Send Video from a source to multiple receivers.
- Multicast Problem with requirements of streaming
  - Regular and continous packet flow with high bandwidth
  - If interactive, low latency.
- Common Streaming Problem: One bandwdith for all no good solution.
  - Either low rate or some users cannot watch.
- Multiple Description Coding (MDC)
  - Instead of one stream with fixed quality or bandwidth, the stream is divided into substreams. The more substreams received the better the quality.





Tree-based Streaming approaches

- Organize peers in multiple trees, each streaming a substream.
  - In one tree as internal node for forwarding.
  - In other trees as external leaf node.
- Tree Construction mechanisms
  - Goal: balanced, short, and stable trees.

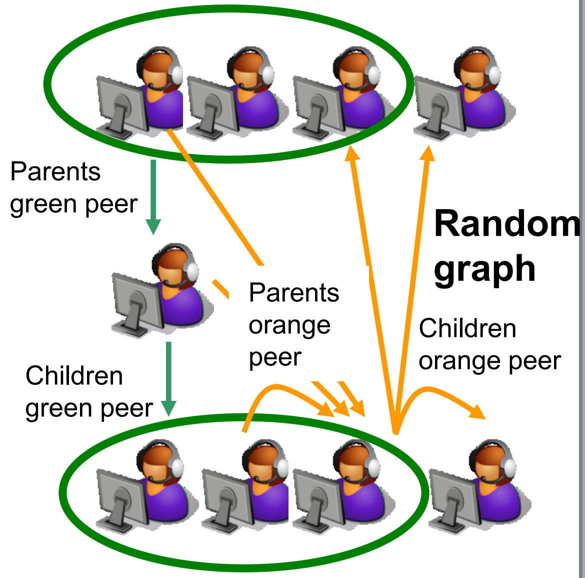


Swarming for Video Streaming (= Mesh-based)

Mesh-based Streaming approaches

- Peers form a randomly-connected overlay (mesh) and use swarming for content delivery.
- Approach (~ PRIME, Magheri et. al, Infocom 2007)
  - Bootstrap node provides random list of peers (potential parents).
  - Maintain certain number of parents, serve a specific number of child peers.
  - For each parent, a child has information about available packets and bandwidth budget.
  - Children download new packets from parents first. The reason is to quickly spread them among peers. They request missing packets if they are still in time and bandwidth is available.
  - Prefer to download from parent with lowest fraction of bandwidth budget utilized.
- Mesh-based streaming seems to outperform tree-based approaches.

  ~ better adaption to available bandwidth of individual peers
  (one reason: better resolution -- packet instead of layer)



Literature

- Judith S. Kleinfeld: Could It Be A Big World After All? The „Six Degrees of Separation“ Myth, Working Paper; Six Degrees: Urban Myth?, Psychology Today, 2001 http://www.uaf.edu/northern/big_world.html
- Albert-László Barabási, Réka Albert: Emergence of scaling in random networks. In: Science. 286, 1999, S. 509–512.
- Stanley Milgram: The Small World Problem. In: Psychology Today. Mai 1967, S. 60–67.
- Duncan J. Watts: Six Degrees – The Science of a Connected Age. Norton & Company, 2003, ISBN 0-393-04142-5.
- Duncan J. Watts: Small worlds. Princeton University Press 1999, ISBN 0-691-00541-9.
- Duncan J. Watts, Stephen H. Strogatz: Collective dynamics of „small-world“ networks. In: Nature. Nr. 393, 1998, S. 440–442.
- Shudong Jin and Azer Bestavros. Small-world Characteristics of Internet Topologies and Implications on Multicast Scaling. Elsevier Computer Networks Journal, vol. 50(5), April 2006.
- Li, Alerderson, Tanaka, Doyle, Willinger: „Towards a Theory of Scale-Free Graphs“, 2005.
- Salman Baset, Henning Schulzrinne: An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. INFOCOM 2006. http://www1.cs.columbia.edu/~salman/skype/
- Bram Cohen: Incentives Build Robustness in BitTorrent. Workshop on Economics of Peer-to-Peer Systems, Berlekeley, CA, 2003. http://www.bittorrent.org/bittorrentecon.pdf
- Nazanin Magharei, Reza Rejaie: PRIME: Peer-to-Peer Receiver-drIven MEsh-based Streaming. *Proceedings of IEEE INFOCOM*, pp. 1415-1423, Anchorage, Alaska, May 2007.

Appendix: German terms

- Unstructured Peer-to-Peer network = unstrukturiertes Peer-to-Peer-Netzwerk
- Structured Peer-to-Peer network = strukturiertes Peer-to-Peer-Netzwerk
- Scale-Free networks = Skalenfreie Netzwerke
- Tit-for-Tat = Wie Du mir, so ich Dir.