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
1.2b Unstructured Systems

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

Overview

1.2b) Systems

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

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

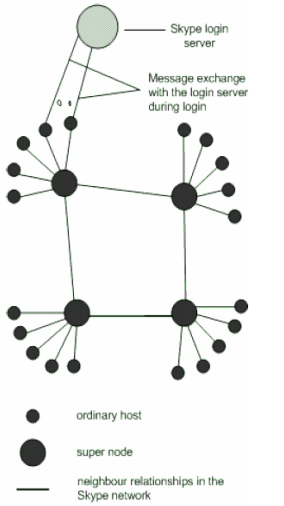
Popular

- 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?



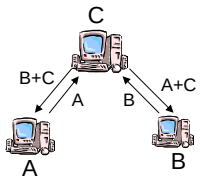
[Baset and Schulzrinne, 2004]

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

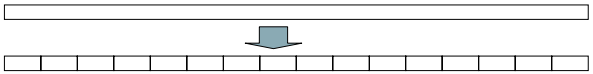
Swarm Intelligence (SI)

- 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

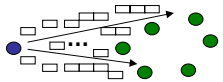
….so, what about swarms in P2P networking?

Swarming

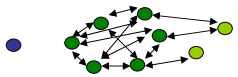
- Idea
  - Peers with the same interest form a swarm and cooperate, instead of individually getting each the same service from one responsible peer or server. → *Goal: a better service*
- For File Transfer / File Distribution
  - Split the file into small chunks. Identify the chunks in some way.



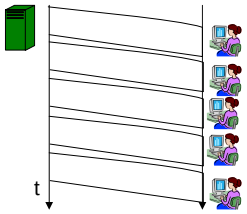
  - Send the chunks to the swarm (= group interested in the file).



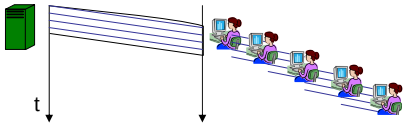
  - Members of the swarm download the chunks from each other in parallel.
    New nodes join and get the data from the swarm.



- 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

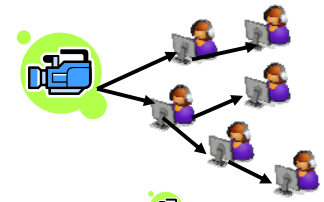
Trackerless BitTorrent

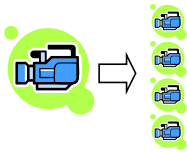
- File transfer: swarming as in tracker-based BitTorrent.
- Tracker is replaced with a Peer-to-Peer network.
  - Structured network (Kademlia) with routing to the torrent ID (info-hash).
  - 8 peers (replica set) can be found via the torrent ID and operate as tracker.
- Join to a torrent
  - A node announces 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

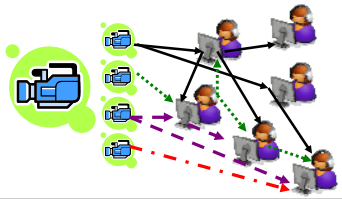
- Send Video from a source to multiple receivers.
- Multicast Problem with requirements of streaming
  - Regular and continous packet flow with high bandwidth
  - If interactive, low latency.
- Common Streaming Problem: One bandwdith for all no good solution.
  - Either low rate or some users cannot 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.

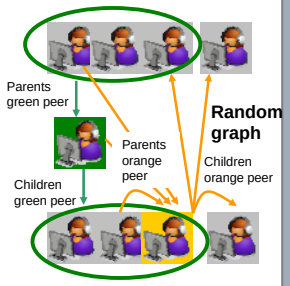


Swarming for Video Streaming (= Mesh-based)

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.