HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Peer-to-Peer Systems

Winter semester 2014

Jun.-Prof. Dr.-Ing. Kalman Graffi

Heinrich Heine University Düsseldorf

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Peer-to-Peer Systems – Chapter 8

Security Issues in P2P Systems

BitTorrent

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Peer-to-Peer Systems

Security Issues in P2P Systems

– Attacks and Countermeasures

DHT Components

DHTs have following components:

- 1. Key identifier space
- 2. Node identifier space
- 3. Rules for associating keys to nodes
- 4. Per-node routing tables that refer to other nodes
- 5. Rules for updating routing tables as nodes join and leave

Any of the above may be the target of the attack

Adversary Model

Adversaries are participants in DHT

- that do not follow protocol correctly (erroneous)
- that follow a protocol that benefits them (selfish)
- That follow a protocol which is harmful for the network (malicious)

Assumptions:

Malicious node can generate arbitrary packets

- Includes forged source IP address

Can receive only packets addressed to itself

- Not able to overhear communications between other nodes

Malicious nodes can conspire together → collusion

- but still limited as above

Attack Layer: Application Layer

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Application layer

- Malicious nodes give false responses
  - e.g., report lower than real bandwidth to have less traffic routed via them
- Freeriding
  - Not contributing / sharing
- Countermeasure:
  - Incentives = rewards for good
  - Penalties for bad behavior (e.g. eMule credit system)
- Example: Mandragora virus
  - Scanned searches of Gnutella client and reported itself as a successful hit

Attack Layer : Network Layer

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Network layer

- Malicious nodes do not follow protocol correctly but provide other nodes with false information about the network state

- Invalid lookup:
  - Messages are forwarded to invalid, non-existing or random nodes instead of the correct ones
    - Waste of bandwidth, time; efficiency is diminished

- Invalid routing:
  - Corruption of routing tables due to false routing information

- Partitioning:
  - Bootstrap to malicious node could lead to the node being kept in a second network consisting of only malicious nodes
    - Good node can be controlled totally
    - All actions can be observed

Identity-based Attacks

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Goal of an attacker

- Take over desired place in DHT

To gain benefit:

- Responsibility for an object ID
  - Might be deleted, modified
- To cut a good node off the network

Requirements for a solution

- Limited possibilities to choose Node ID
- Verification of Node ID

Sybil attack (I)

Sybil is the title of a book about a schizophrenic woman "Sybil"

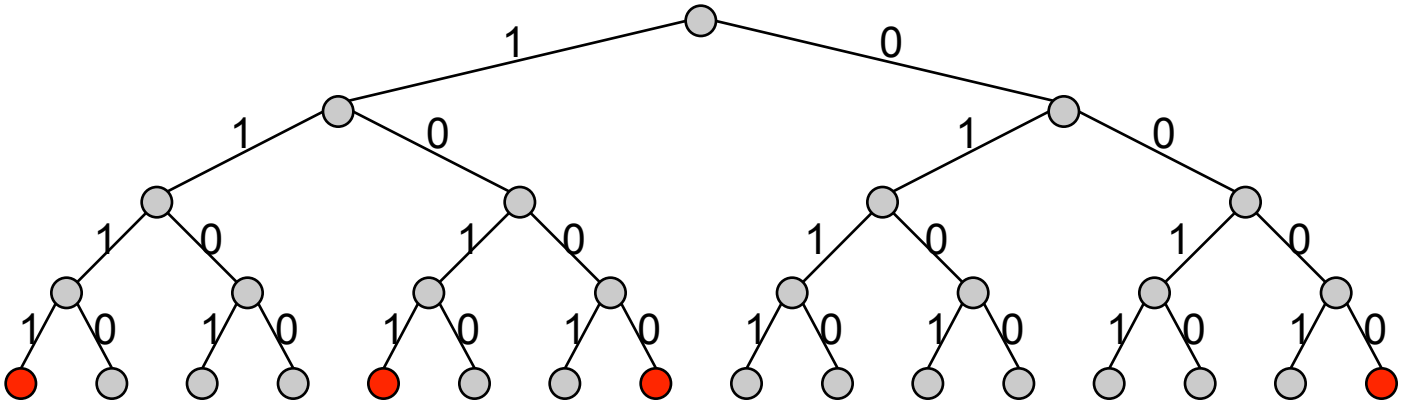
Sybil attack:

- A single node joins the network with multiple identities
- One physical entity possesses multiple logical identities

Result:

- Undermining of replication (data loss) resp. fragmentation (data leakage)
- Entity gets more global view on the network
- Observation of other nodes possible
- Lower robustness: If physical sybil node fails, all logical identities fail
- Disruption of majority decisions, e.g. for join permission

Sybil attack (II)



Example: Kademlia-based network with 4-bit-identifiers

The attacker introduces 4 sybil identities

- Controls thereof 25% of the network

Eclipse attack (I)

Conspiracy of malicious nodes to introduce themselves into the routing tables of good nodes

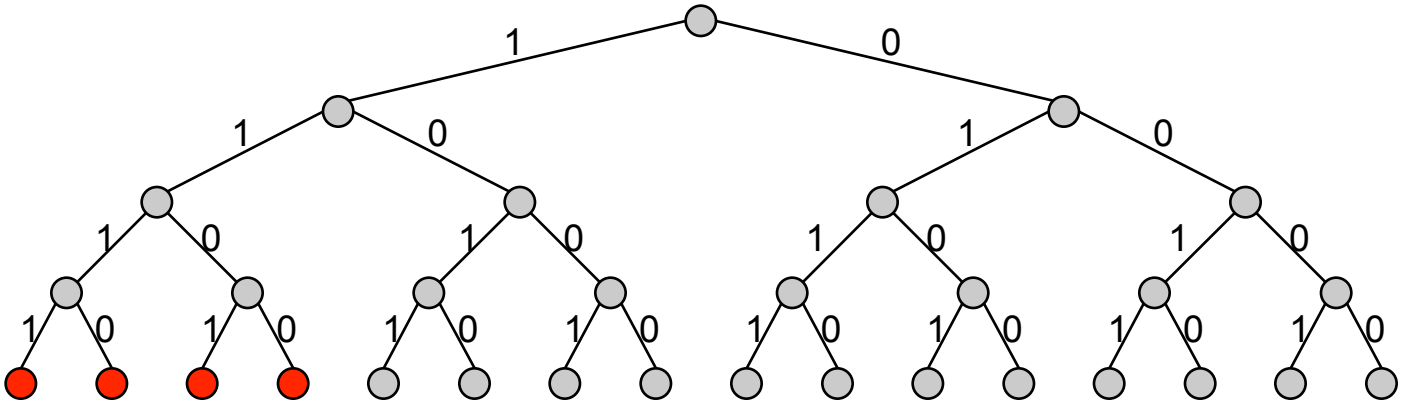
- Manipulation of the routing process
- “Eclipse” parts of the network
  - Prevent correct nodes from contacting each other
  - Censorship

Sybil attack can be used to perform an Eclipse attack

- Sybil attack is not necessary. Even if countermeasures against a Sybil attack are in effect, an Eclipse attack is possible

Free choice of node IDs

Eclipse attack (II)



Example: Kademlia-based network with 4-bit-identifiers

The attacker introduces 4 sybil identities

- Controls thereof IDs 1100 to 1111

Solutions against Identity Attacks

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Required

- Verifiable ID
- ID generation limited (no free picking)
- Some suggest: pay for ID
  - Money
  - Computational time

Secure ID through Trusted Third Party

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

One main solution: trusted third party

- Controls the Node ID assignment

Limits Sybil / Eclipse attacks

Often combined with reputation approach

- All nodes report observed misbehavior to trusted party
- Trusted party analyzes behavior
- Can revoke Node ID

Verifiable ID through Device-bounding

Bind Node ID to device characteristics

Node ID is hash of

- IP, port
- Media Access Control (MAC) address
- Round Trip Time(RTT) between the node, and some randomly selected nodes in the same subnetwork.

E. Sit and R. Morris. Security Considerations for Peer-to-Peer Distributed Hash Tables. Springer, 2002.

H. Wang, Y. Zhu, and Y. Hu. An Efficient and Secure Peer- to-Peer Overlay Network. In Proc. of IEEE LCN, 2005.

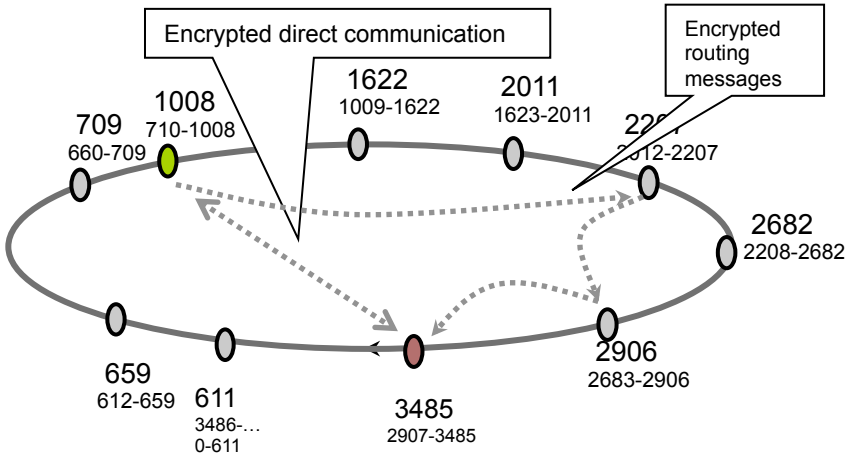
Secure ID: ID-based cryptography

Root of trust

- Identity-based cryptography
  - Asymmetric public key used as node ID
- Goal: Verifiable ID, secure communication

Example:

- Pastry: ID size 128 bits
- Using public keys: ID size 2048 bits



Allows for

- Instant encrypted communication
  - As long as node ID is known
- Verifiable content:
  - Signature by sender with private key
  - Receiver can check signature using the node ID

Drawbacks:

- More traffic overhead (larger messages)

Malicious Network Partitioning

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Attacker can partition network

- If new node contacts attacker first, attacker can partition network
  - can even hijack nodes from real network
- Parallel network is consistent and “looks OK”
  - Attacker can track nodes
- Bootstrap from a trusted source:
  - Hard to get in dynamic networks, public keys might help
- Cross check routing tables with random queries
  - Assumes we were part of network earlier, still not totally safe

Crypto puzzles (I)

Payment with real money difficult to handle without central authority

- Nodes pay with resources instead
  - (storage, bandwidth, computing power, …)

Static crypto puzzle

- To hinder free node ID selection (Eclipse attack)

Dynamic crypto puzzle

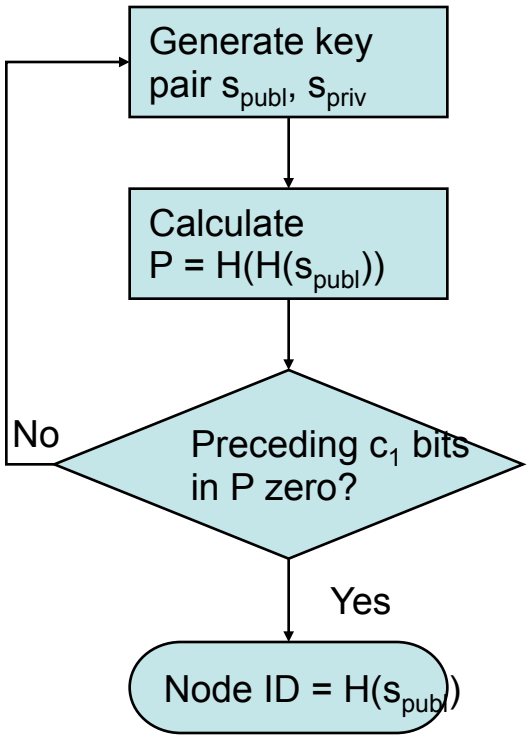
- To hinder the introduction of a large number of nodes by one physical entity (Sybil attack)

Assumption

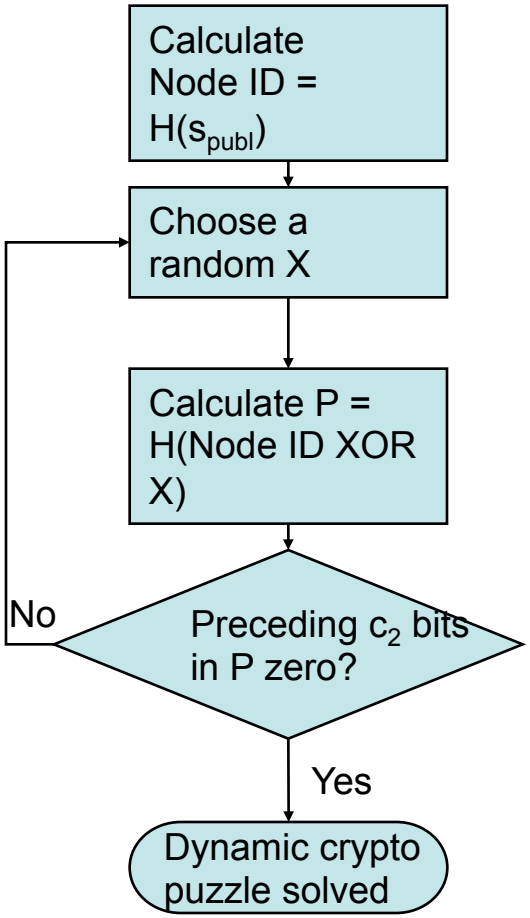
- Computing power rises over time, so computations that take a long time today will perform faster in the future
  - c1 is static. Each node generates its node ID using c1
  - c2 can be changed (increased). Node IDs can be (re)validated later using c2

N. Borisov. Computational Puzzles as Sybil Defenses. In Proc. of IEEE P2P, 2006.

Crypto puzzles (II)



Static crypto puzzle



Dynamic crypto puzzle

Routing Attacks

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Routing is

- responsible for maintaining routing tables
- and sending messages to correct nodes

Attacker can forward messages incorrectly

- But: Each hop should get “closer” to destination
- Malicious node might pick very far next hop

Attacker can claim wrong node is responsible node

- Querying node is “far away”, cannot verify this
- Assign keys to nodes in a verifiable way

Securing Routing

Assumption:

- Verifiable Node IDs (hashes of IP … or Public Keys)

Routing path should be observed

- Allow querying node to observe and verify lookup process
  - For example, processing messages recursively hides this

→ Iterative or tracer routing

No trust in intermediate nodes

- One single path might break

→ Redundant / parallel routing

Routing Attack: Routing Table Poisioning

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Attacker sends incorrect routing updates

- Answers only with other malicious nodes
  - To spread influence of them
  - With time, bad nodes dominate in routing table
- Combined with sybil attack
  - Good node might get eclipsed

- Obviously wrong updates can be detected
  - Wrong source node ID, zone sizes …
- If DHT allows several choices for next hop
  - Attacker can pick a “bad” node
  - Not necessarily a problem with correctness, only performance
  - Can be a problem for some applications (anonymity)
- Server selection can be abused

Coutermeasure for Routing Table Poisioning

Have two routing tables:

- Optimized routing table for regular routing
- Verified routing table
  - Only to be used if regular routing table fails
  - To check validity of regular routing

Erase periodically the optimized routing table

- To hinder malicious nodes from dominating

Artificially induced churn

- All peers change their ID periodically
- Thus they are harder to eclipse
- (Significantly larger costs)

Storage and Retrieval Attacks

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Attacker can deny existence of data

- Or return wrong data

Must implement replication at storage layer

- Who creates replicas?
- Clients must be able to verify that all copies were created

Avoid single points of responsibility

- Replication with multiple hash functions is one good way

Big problem if system does not verify IDs

- Any node can become responsible for any data
- For example, Chord allows virtual nodes

Disjoint path routing

Kademlia uses α parallel queries

Lookup fails if only one adversarial node is queried

- If this node provides false routing information that is "better" (closer to the target) than valid responses

Proposal for increased security:

- Perform α parallel lookups
- Start with taking the k closest known nodes to the target and distribute them uniformly among the α search processes
- Each of the α lookups is performed like a normal Kademlia lookup except for the fact that each encountered node is used by only one search process

Result: Adversarial nodes cannot influence all search processes (if at least 1 of the initial k nodes is good)

Secure and Access Controlled Storage

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

DHTs allow distributed storage and retrieval of data

- No control on where data is stored
- especially when replicated

Requirements for access control

- Goal: storage of data with enforcement of access control list
  - Set of “privileged users” to be added
- No unauthorized user can read / modify content
- Author is authenticated
  - Content integrity can be checked
  - False authorship can be rejected
  - Data objects should be signed

Secure and Access Controlled Storage

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Assumptions:

- Existence of public key infrastructure
  - E.g. node ID = public key

Data object must be replicable

- Replication
  - Multiple storage of the same content
  - Find copy to read
  - Find all copies to write consistently
- Replication independent of security
  - Efficient mechanisms usable

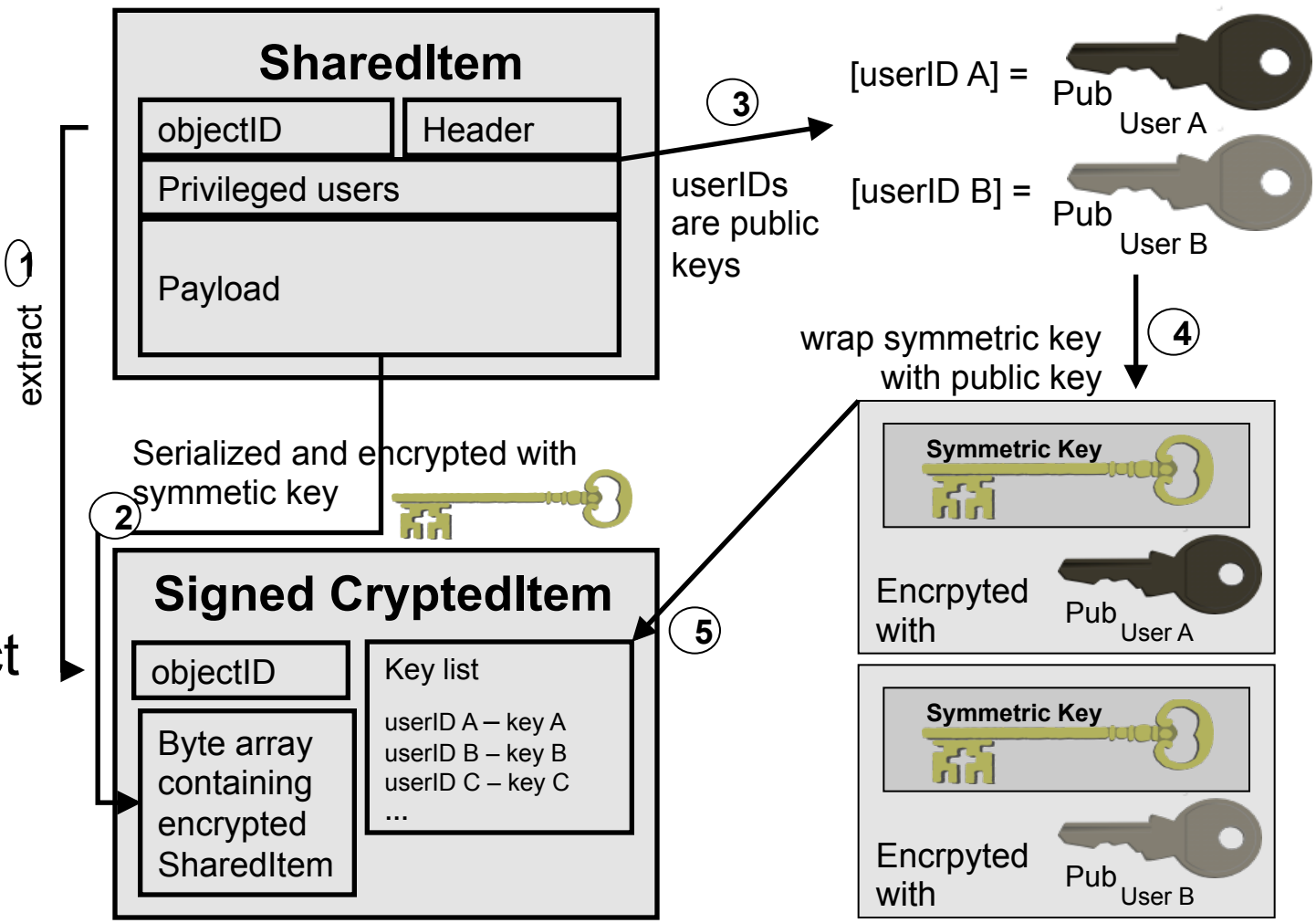
Practical Distributed Access Control

Step 1:

PeerID
= UserID
= PublicKey

Step 2:

Attach
enc. keys
to
data object



Access Control: Data Overhead

Overhead depends on number of privileged users

- Each additional privileged user causes data overhead of about 420 bytes

Comes from added UserIds and wrapped keys to the key list

Much more overhead than in message encryption but acceptable

Privil. Users	Item Size (bytes)	Encryp. Item Size (bytes)	Overhead abs. (bytes)	Overhead rel. (%)
1	346697	348159	1462	0,42
10	346715	351892	5177	1,49
50	346819	368524	21705	6,26
100	346969	389318	42349	12,21
200	347269	430922	83653	24,09

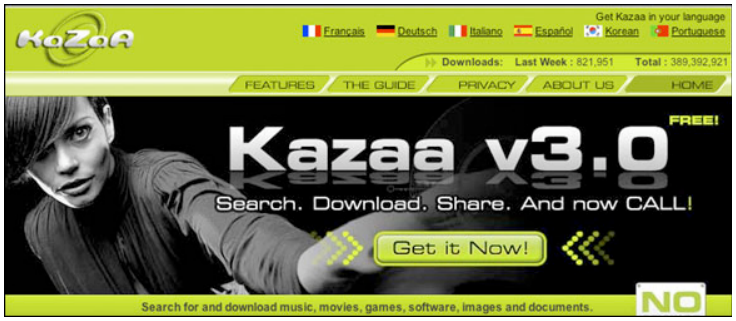
HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Peer-to-Peer Systems

Content Distribution and Multimedia Streaming
– BitTorrent

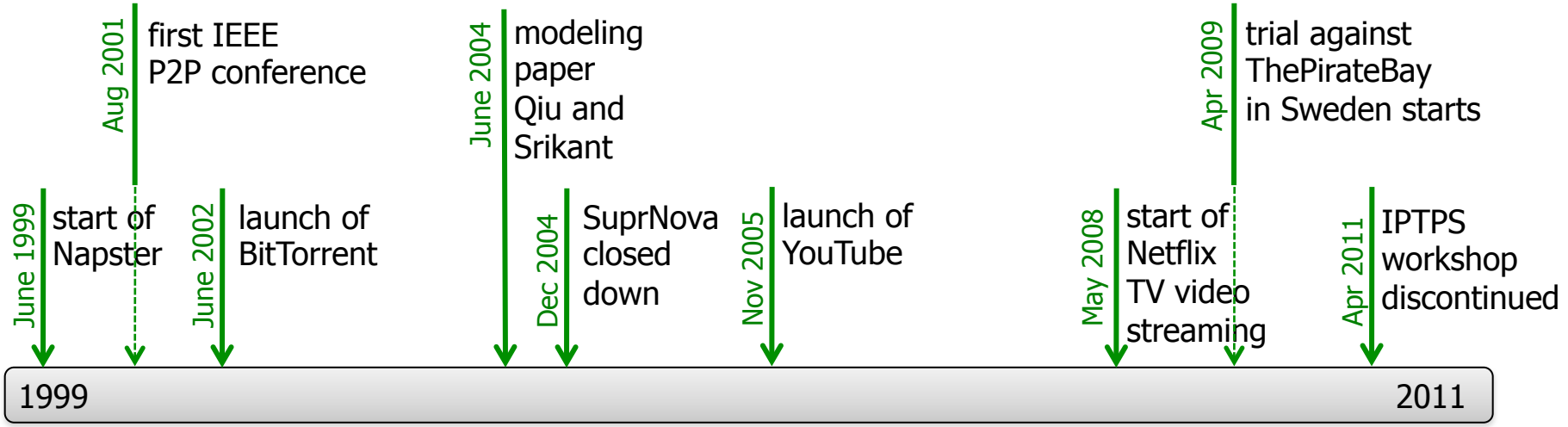
This slide set is based on the lecture "Communication Networks 2" of Prof. Dr.-Ing. Ralf Steinmetz at TU Darmstadt

P2P file sharing: a mixed bag



Slide based on Dick Epema, Tribler, “Peer-to-Peer File Sharing: Past!–Present–Future? A Delft View" presented at IEEE P2P 2011

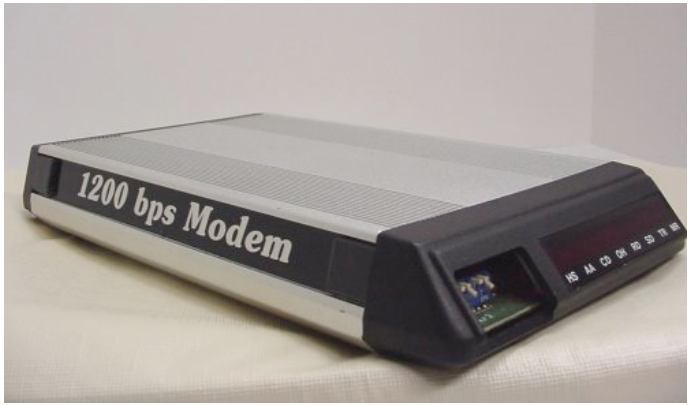
A Little History of File Sharing



Bandwidth (1/3): 3000-fold increase

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

1200 bps dial-up modem



1988

500 MB movie: **40 days**

4 Mbps/1Mbps ADSL

2011

20 minutes

Slide based on Dick Epema, Tribler, “Peer-to-Peer File Sharing: Past!–Present–Future? A Delft View" presented at IEEE P2P 2011

Video-on-demand vs file sharing (US)

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

	Upstream Traffic		Downstream Traffic		Total Traffic	
Rank	Application	Share	Application	Share	Application	Share
1	BitTorrent	52.01%	Netflix	29.70%	Netflix	24.71%
2	HTTP	8.31%	HTTP	18.36%	BitTorrent	17.23%
3	Skype	3.81%	YouTube	11.04%	HTTP	17.18%
4	Netflix	3.59%	BitTorrent	10.37%	YouTube	9.85%
5	PPStream	2.92%	Flash Video	4.88%	Flash Video	3.62%

SOURCE: SANDVINE NETWORK DEMOGRAPHICS

sandvine

Slide based on Dick Epema, Tribler, "Peer-to-Peer File Sharing: Past!–Present–Future? A Delft View"

BitTorrent

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Involved nodes / roles:

Tracker

- Non-content-sharing node
- Actively tracks
  - Swarm: all peers (seeders and leeches)
  - Status of downloaded data - volume
- One swarm per file

Seeders

- Have complete copies of the desired content

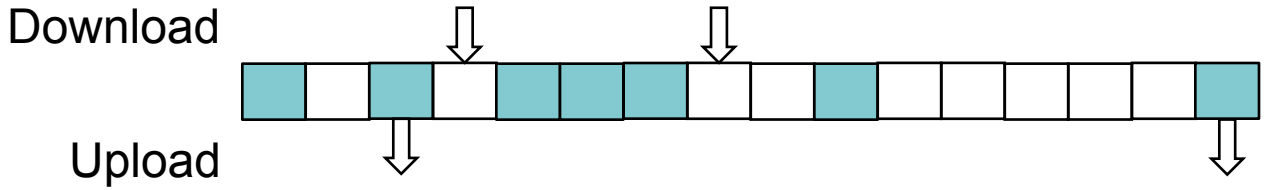
Leechers

- Incomplete copies of the desired content
- Leeches try to download missing chunks

BitTorrent Design (1/3): Swarms

File is split into chunks

- Typical size 256kB
- Allows download from parallel sources
- Allows parallel up- and downloads of same file



One network (=swarm) per file

- Search & lookup of chunk providers
- NOT: search for file providers

Tit-for-Tat: chunk exchange strategy

- Assuming all peers in swarm want file
- Chunks: download requires upload to the same peer

BitTorrent Design (1/3): Swarms

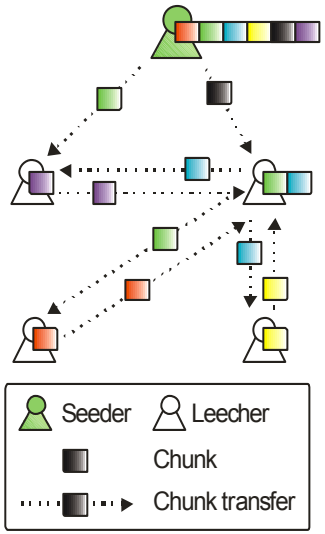
File divided into chunks (size 256 KB, 1000 per file)

Swarm – the set of peers downloading the same file

Seeders – online peers with the complete file who upload for free

Leechers – peers whose download is in progress

Chunks exchanged between peers according to tit-for-tat strategy

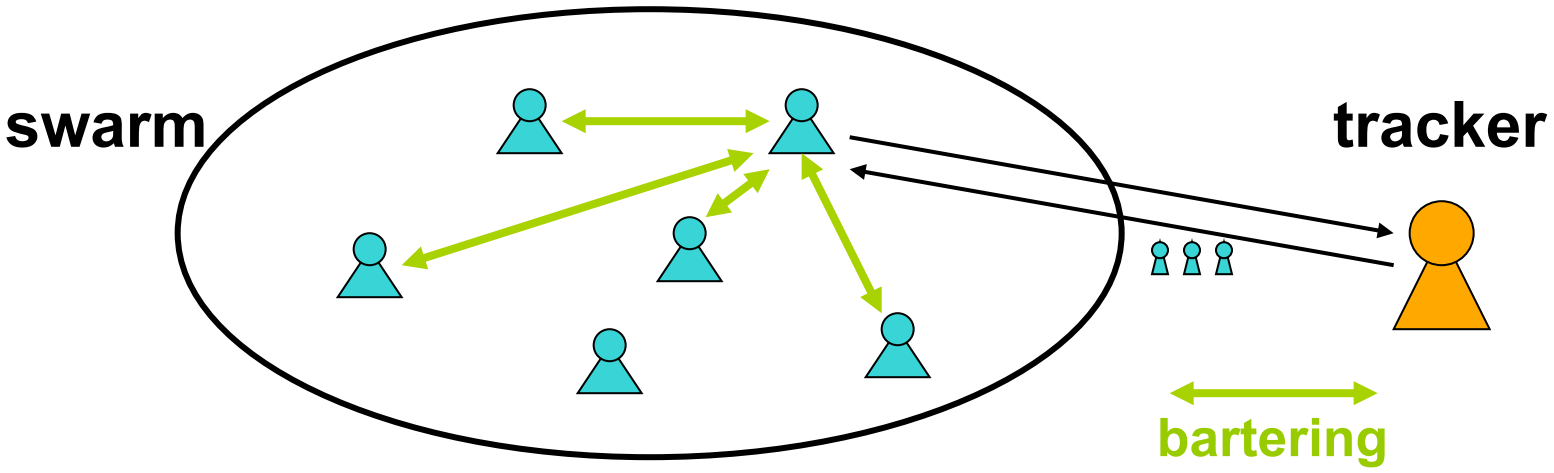


BitTorrent Design (2/3): Trackers

Torrent metadata files contain tracker URL(s)

Trackers centrally manage swarms:

- A peer indicates its interest in a file to a tracker
- Peers periodically contact a tracker
  - to obtain the IP numbers of other peers downloading the same file
  - Info on ca. 40-50 peers obtained
- A peer selects the best other peers as bartering partners
  - Share group: 4 nodes



Slide based on Dick Epema, Tribler, "Peer-to-Peer File Sharing: Past!–Present–Future? A Delft View"

BitTorrent Design (3/3): Downloading

Peers in a swarm exchange the IDs of the chunks they have

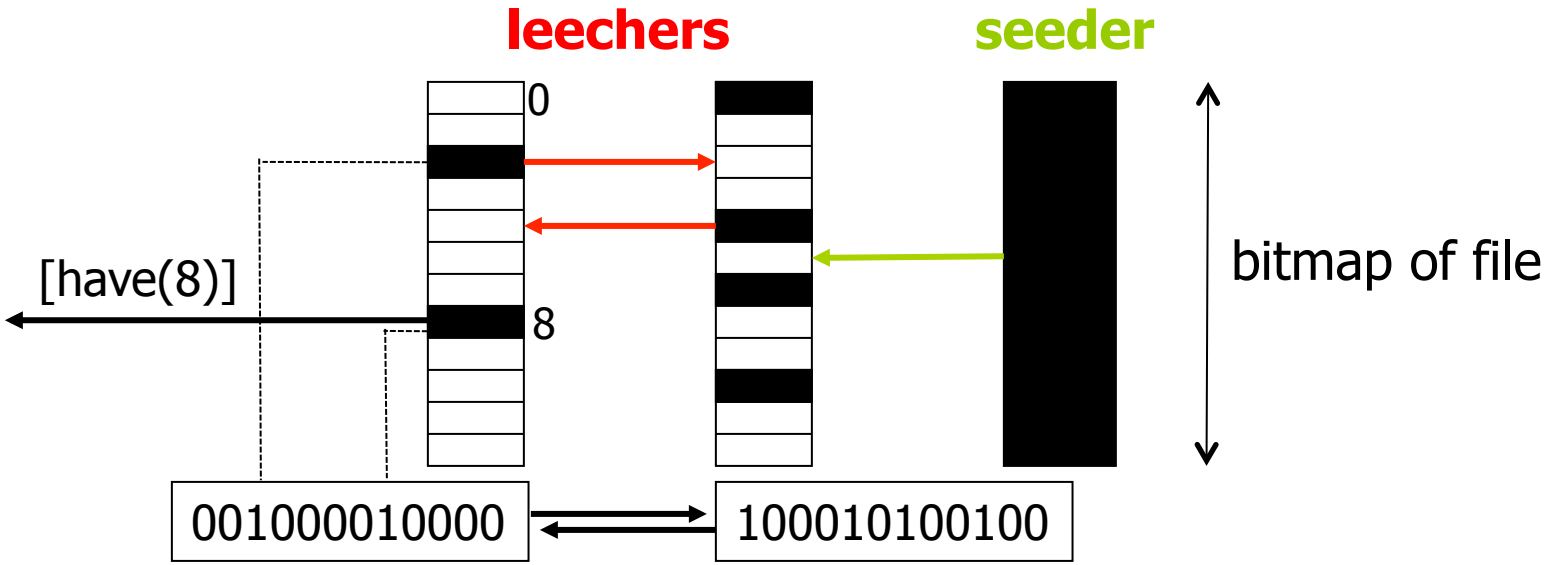
- With a bitmap or with have-messages

Leechers barter for chunks of the file with other leechers

- And get chunks for free from seeders

Chunk selection uses the rarest-first policy

Periodic optimistic unchoke to find better bartering partners



Slide based on Dick Epema, Tribler, "Peer-to-Peer File Sharing: Past!–Present–Future? A Delft View"

BitTorrent – Tit-for-Tat Exchange Strategy

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Principle:

- While I give you, you give me
  - Hinders free riding
  - Supports contributing peers to get good service
- Being optimistic about unknown nodes

Share group:

- Every peer exchanges chunks with only 4 peers
- Chosen due to fastest upload speed

Choking and Optimistic Unchoking

- Periodically (30s) slowest of the 4 is released
- Random new peer is chosen (unchoked)

Chunk Selection Policy

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Start strategy:

- Fresh node without chunks announces presence over tracker
- Hopes for optimistic unchoking
- Picks random chunk to download

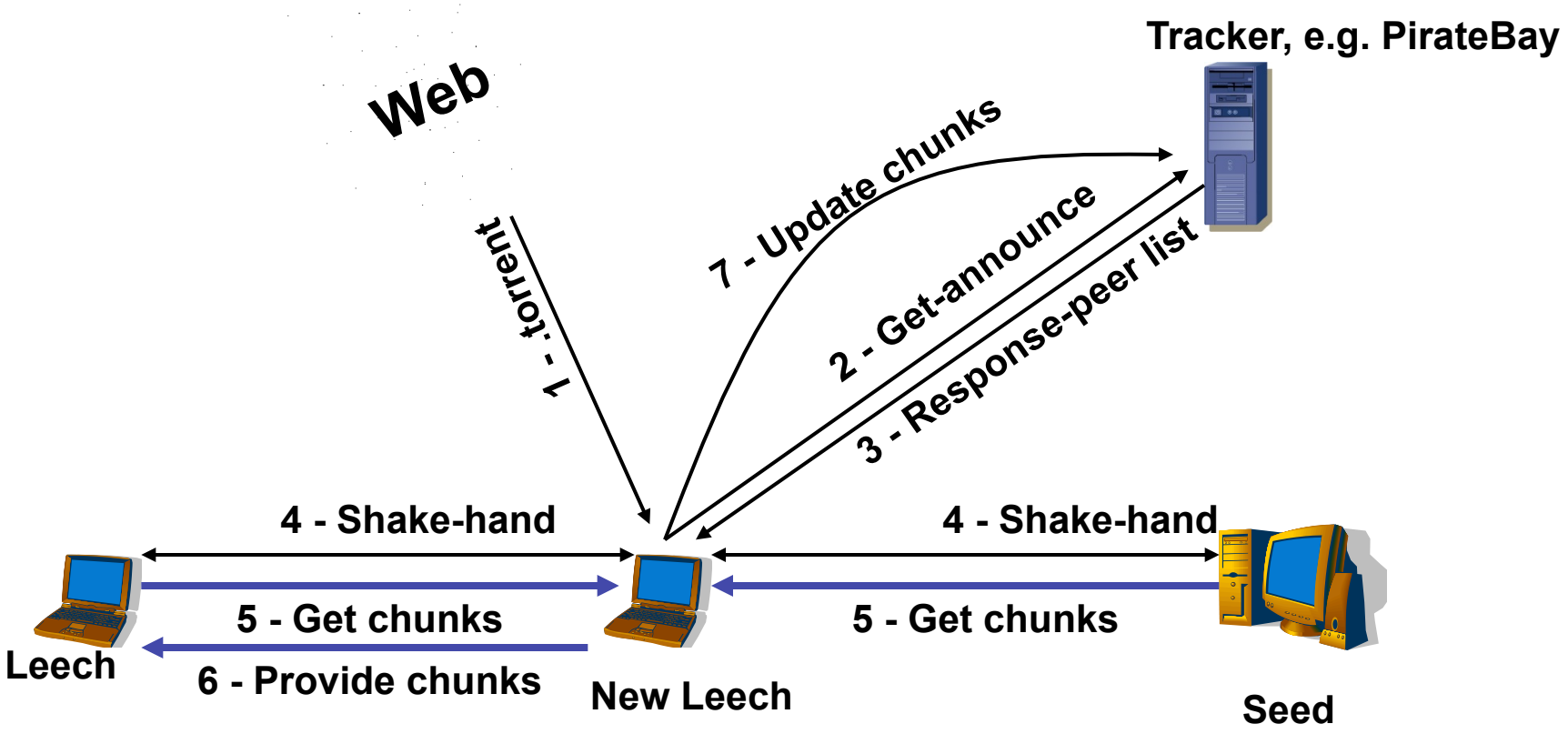
Tit-for-Tat strategy (normal case)

- Rarest first policy
- Supports the availability of the file, by sharing more rare parts

End game

- If only a few chunks are missing,
- Last missing chunk may take long to find
- Requests are sent to known peers

BitTorrent: Components



Phases of BitTorrent

1. Get torrent from Web
   - Several websites offer torrents (a file type)
   - Torrent contains information about the file: filename, size, checksum, and a corresponding tracker

2. Get announce, contact tracker
   - Tracker maintains a list of active peers sharing the file
   - One tracker per file, one network (swarm) per file

3. Response peer list
   - Peer list contains up to 50 peers
   - Seeders and leechers
   - Large number of peers sharing the same file → allows for parallel downloads of chunks

Phases of BitTorrent

4. Shake-hand
   - Peer establishes connection to 20-40 peers from the peer-list
   - Checks periodically their bandwidth capabilities → prioritized list
   - Uses just 4 for exchanging chunks
   - Various replacement strategies

5. Get chunk
   - Peers request chunks using the rarest first policy
   - Parallel chunk download allows for faster file downloads

6. Provide chunk
   - Tit-for-tat strategy, peers give as long as they receive some chunks
   - Slow startup phase, as not chunks provided

7. Update chunk information at tracker

BitTorrent Extensions

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Tracker is single point of failure

- See case: PirateBay
  - Lawsuites to hinder hosting of torrent files
  - Remember: torrents only contain metadata

DHT-based tracker

- Uses KAD to store torrents
- Now widely deployed and used approach

Further extensions

- Locality-aware BitTorrent: Prefer peers from same area
  - Lowers costs for Internet Service Provider, as traffic remains local
- Multi-swarm BitTorrent:
  - Tit-for-tat-exchange extended to over 2 swarms
- Streaming over BitTorrent
  - Adapted chunk selection allows watching a stream while downloading

Conclusion on BitTorrent

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Whole BitTorrent design in three slides

But:

- tens of millions of users
- large fraction of Internet bandwidth consumed

Reasons:

- simplicity
- effectiveness
- timeliness

Very strong assumptions in mathematical model:

- Poisson distributions
- steady state

Still this analysis reveals important insights

Mathematical analysis may be (in)validated

- by simulations and measurements, and may steer design

BitTorrent: Evaluation

Strengths

- Good bandwidth utilization
- Limit free riding
  - tit-for-tat
- Limit leech attack
  - coupling upload & download
- Spurious (“unechte”) files not propagated
- Open source implementations
- Preferred selection
  - for legal content distribution

More information:

- BitTorrent is a community project

http://wiki.theory.org/BitTorrentSpecification

Drawbacks

- Small files
  - lead to latency, overhead
- Central tracker server needed to bootstrap swarm
  - Single point of failure
  - Scalability issue
- Robustness
  - System progress dependent on altruistic nature of peers
- Cannot totally avoid malicious attacks at leeches
- Centralized supervising tracker – trust?

P2P Traffic: An Issue for ISPs

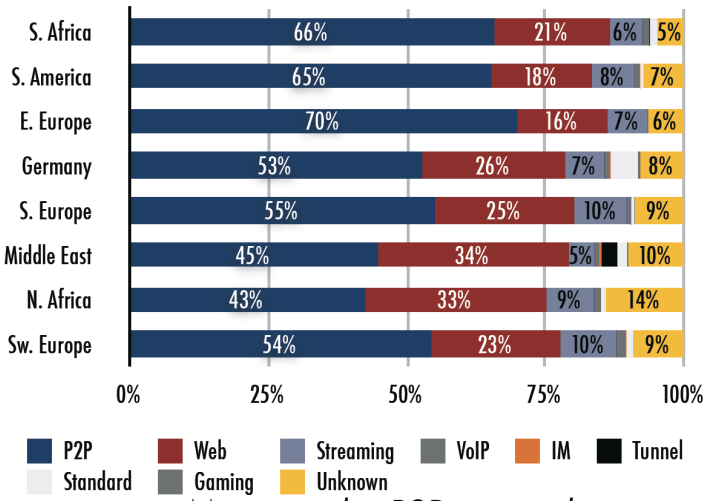
2009: P2P was dominating

- P2P makes 43%-70% of overall Internet traffic
- BitTorrent makes 30%-80% of p2p traffic worldwide
- Lots of intercontinental traffic

Problems for ISPs

- Tit-for-tat strategy optimizes for low download times
- Typically chunk provider is far away
- Generates inter-ISP traffic → expensive
- Local providers are more ISP-friendly

Distribution of protocol classes 2008/2009



Most popular P2P protocols

Protocol	Southern Africa	South America	Eastern Europe	Northern Africa	Germany	Southern Europe	Middle East	South-western Europe
All P2P	65,77%	65,21%	69,95%	42,51%	52,79%	55,12%	44,77%	54,46%
Ares	0,29%	42,63%	0,00%	2,24%	0,84%	0,16%	0,11%	1,80%
BitTorrent	48,34%	30,02%	80,83%	74,51%	70,77%	48,94%	78,85%	58,20%
DirectConnect	0,01%	0,00%	17,87%	0,08%	0,85%	0,00%	0,12%	0,30%
eDonkey	2,48%	25,99%	1,16%	7,70%	24,22%	47,17%	15,37%	35,99%
Gnutella	18,60%	0,36%	0,14%	14,21%	1,75%	1,66%	5,00%	2,75%
iMesh	13,60%	0,02%	0,00%	0,47%	0,00%	0,03%	0,00%	0,14%
Thunder	14,04%	0,80%	0,00%	0,69%	0,77%	1,64%	0,52%	0,62%
Other	2,64%	0,19%	0,00%	0,10%	0,68%	0,41%	0,03%	0,21%

Sources: ipoque Internet Study 2008/2009

Strategies to reduce ISP costs

Filter P2P traffic → provide worse quality

Usage based billing (differentiated services)

- Introduce fair flat rates
- volume based flat rates

Caching of P2P content by ISPs

Cooperate of ISPs with P2P application providers

- Building P2P applications aware of the network locality
- P4P: Provider Portal for Applications
- Idea: ISP intervene in step 3
  - Peer requests peer list
  - Tracker contacts P4P server to localize peer list
  - Localized peer list is delivered

Result:

- Chunk providers are in same / near ISP network
- More P2P traffic remains in the ISP network
- Lower costs for ISPs
- But: ISP knows what you download