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 1.4 Other Issues

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

Overview

- Maintenance / Churn (revisted)
- Optimization
  - Locality (Distance)
  - Load
- Technical Issues
  - NAT / Firewall

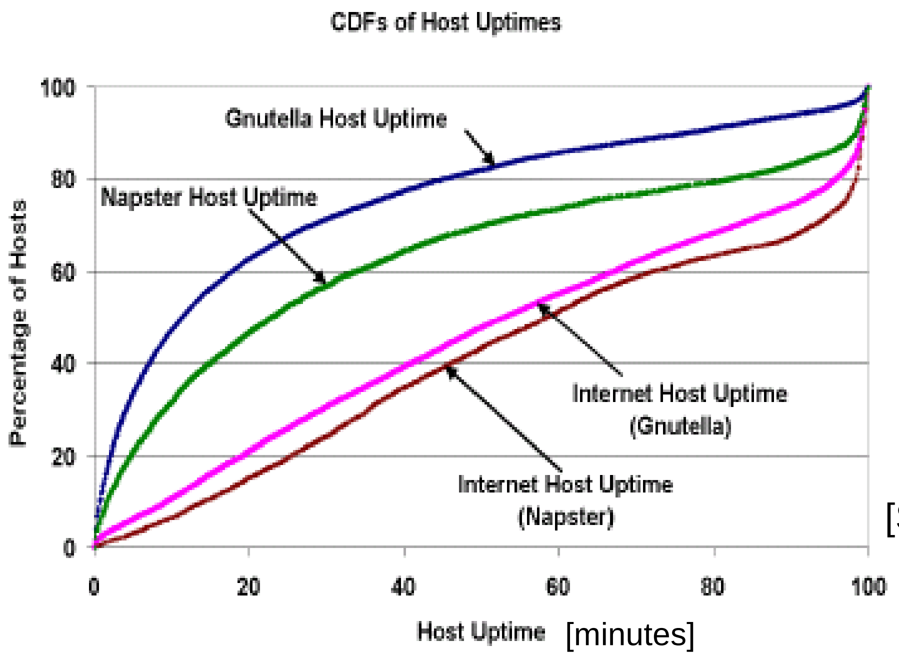
Maintenance / Churn

Maintenance

Maintenance

- Problem
  - Joins and leaves (= „Churn“)
  - Changes in network connectivity
- Goals
  - Keep the network alive
  - Keep the data stored
  - Provide a good quality of service
- Solutions
  - Know more nodes
    - Successor/neighbor lists to recover.
    - Buckets instead of single links.
  - Be up-to-date
    - Frequently check if nodes still exist and routing table is correct.
  - Store data on nodes of a replica set (= group of nodes)
  - Use networks with low join/leave overhead if churn is high.

Churn – Filesharing

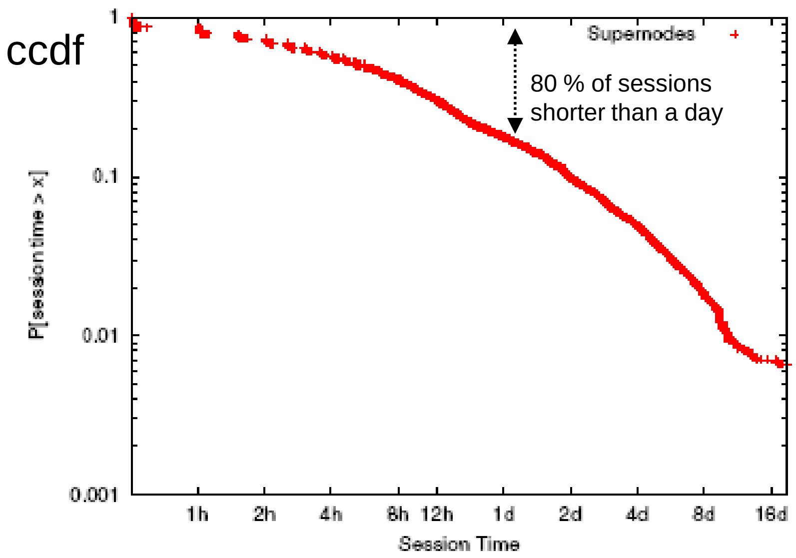


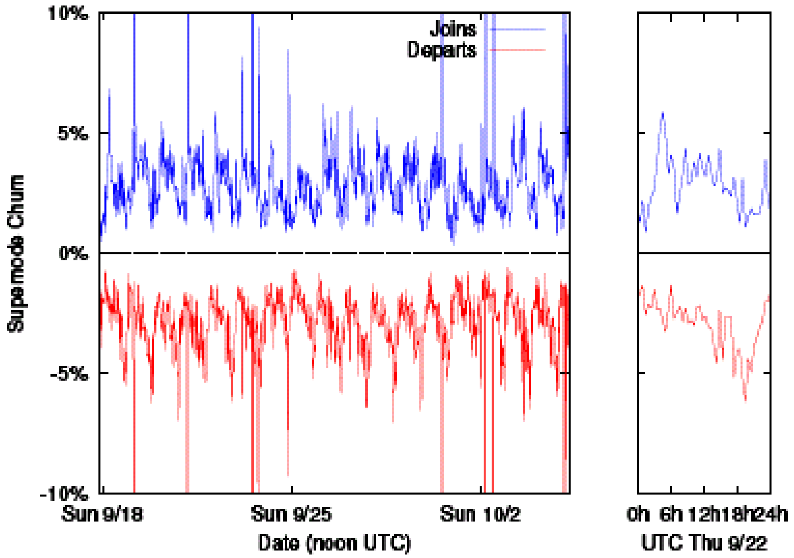
[Saroiu, Gummadi, Gribble, 2002]

Filesharing

- The Internet Host Uptime figures show how long the host is online. The Gnutella and Napster uptime are how long the application is running on the host.
  - Obvious: Gnutella/Napster Uptime <= Host Update
- Most nodes are only available for a short time. However, some are online for several days.
- A measurement by Chu, Labonte and Levine reports that short short sessions are even more likely. They speculate about the session time being an overlap of Zipf distributions (heavy-tail / power law distribution).

Churn – Skype





[Guha, Daswani, Jain, IPTPS 2006]

Skype

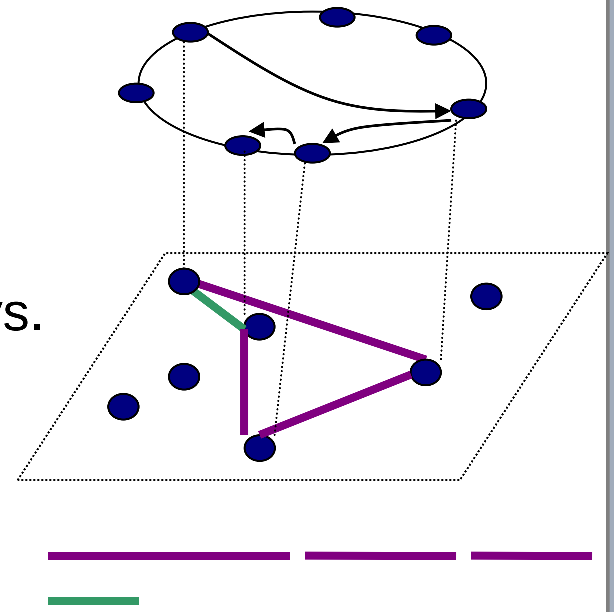
- Super nodes
  - Relatively stable network (session time less than a day does not imply that the node won't be back).
  - Possible approx: Poisson arrival, session time heavy-tail
  - Variations over daytime and weekday.
  - 95 % of super nodes will be there 30 min later

Optimization (Locality, Load)

Optimization – Locality

Locality

- Basic Peer-to-Peer network construction ignores geographic or underlay relationships.
  - → Long path with hops all over the world may be used to contact a local neighbor. *Inefficient!*



Stretch

- Metric for locality in and performance of overlays.
- Idea: Compare distance over overlay with distance in the underlay.

$$Stretch = \frac{DISTANCE\_via\_overlay}{DISTANCE\_in\_underlay}$$

- Any distance metric can be used,
  latency (default) or IP hops are common.

Locality – Approaches

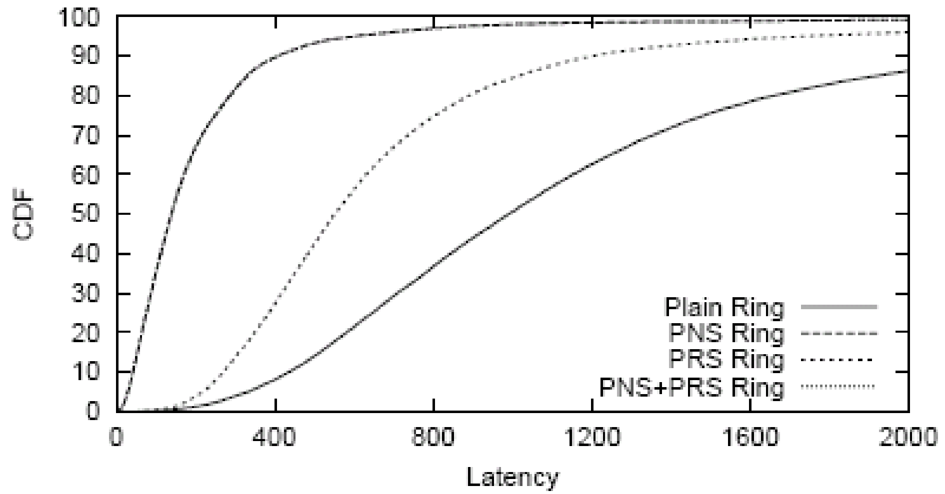
Basic approaches to improve locality

- Proximity Neighbor Selection (PNS)
  - Given multiple potential nodes for a routing table entry, select the best from the candidates according to proximity metric (e.g. latency, RTT).
- Proximity Route Selection (PRS)
  - Use routing table as candidate set and select next hop as trade-off between latency and reduction of distance in ID space.
  - Alternative understanding of PRS (almost identical to PNS): Given multiple routing entries into one direction, chose the next hop with best proximity metric.
- Proximity Identifyer Selection (PIS)
  - Select identifyer in a way to minimize proximity to neighbors in overlay.
    - Internet Coordinate Systems are examples.
- Special solutions
  - Pastry: Neighbor Set with local nodes, join via local node
  - Toplus: Use IP adresses as overlay IDs.

Locality – Comparison & Discussion

General Results

- PNS better than PRS
  - Reason: PNS uses #candidates * size(routing table) nodes to optimize.
    PRS only uses size(routing table) nodes.
- PNS+PRS only slightly better than PNS, in Figure on the right the two curves are indistinguishable.



Gummadi et al, Sigcomm 2003

Discussion

- Optimizing locality may conflict with the idea of diversity, i.e. to improve robustness by relying not only on geographically local nodes, but on internationally distributed ones.

Optimization – Load Balancing

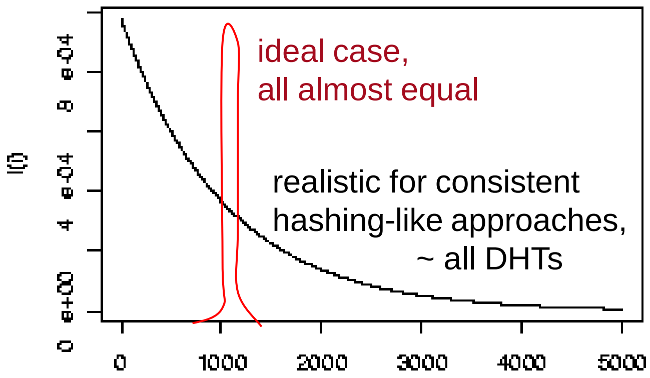
What is load?

- Items stored
- Computational effort
- Traffic and work imposed by requests.
- Maintenance traffic

Problem?

- Peers may be assigned more work due to larger intervals.
  - Low probability of bad luck in randomized scenarios.
- Non-random approaches for work and ID assignment
- Requests may not be uniformly distributed among items. → Zipf's Law

Interval size distribution



Zipf's Law

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

- Zipf distribution is Power-Law.
- Zipf-like popularity can be found for websites, words in natural languages, movies.
- However, with static content that is only requested once, there is usually a saturation for popular files and the behaviour not fully Zipf (measurements of Kazaa, Gnutella,…).
- Popularity of items is application-dependent. Not all applications create Zipf-like distribution.

→ Extreme differences in popularity are one argument for the need for load balancing.

Necessity of Load Balancing and basic ideas

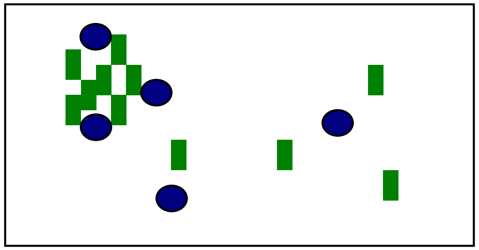
Random item IDs

- Load of each node follows binomial distribution with low variance.
- Variation of the size of ID ranges is dominating.

→ Node Balancing is important

Non-random item IDs

- Need to balance load as items may cluster in certain areas and build hotspots.



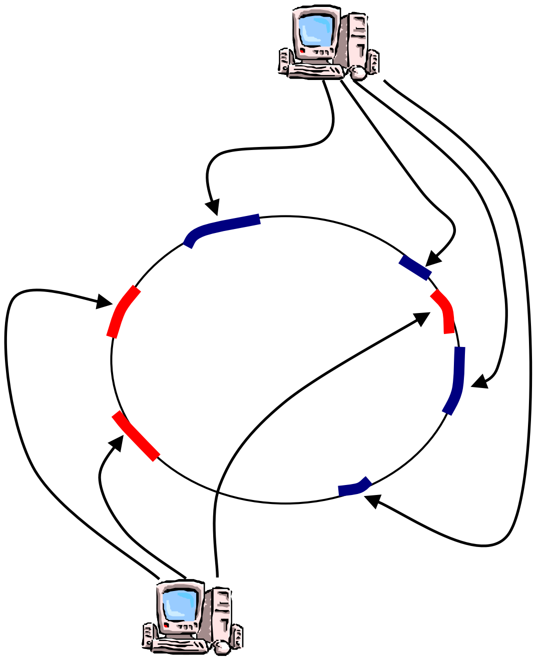
Basic ideas for load balancing

- Methods to reduce variation in number of items and interval size.
- Share load at hotspots by using nodes in parallel.

The Virtual Server approach

Virtual Server

- Each node is represented by O(logn) virtual nodes called virtual servers.
  - Having O(logn) intervals already averages out some imbalance. Theory of consistent hashing suggests this number (fair with high probability).
- Virtual servers can be transfered from overloaded (heavy) to light nodes
  - The light node does not become heavy.
  - The transfered virtual server is the lightest to make the heavy node light.
  - If no virtual server is heavy enough to make the heavy node light, transfer the heaviest virtual server.
- How do they find each other?
  - Randomly contacting other nodes.
  - Directories with information about heavy and light nodes.



Simple Efficient Load Balancing

Simple Efficient Load Balancing (Karger/Ruhl)

- Address-Space Balancing
  - Instead of having O(logn) virtual servers, each node has a fixed set of O(logn) possible positions, only one of them is active.
  - The active virtual server is selected according to interval size.
    - To avoid the selection of mini-intervals, the „smallest“ address range is selected according to the order 1 < ½ < ¼ < ¾ < 1/8 < 3/8 < 5/8 < …
- Item Balancing
  - Nodes randomly connect each other.
  - IF one of them has ε times less items than the other THEN
    - It moves to the interval of the other node. Both nodes fairly divide the interval so that they share the same number of items.
- Results
  - Approach has provable bounds for address-space and item balancing.
  - Item Balancing needs Ω(logn) random connections for each node per half-life and when the number of items on a node doubles.

Dealing with Hotspots

Hotspots

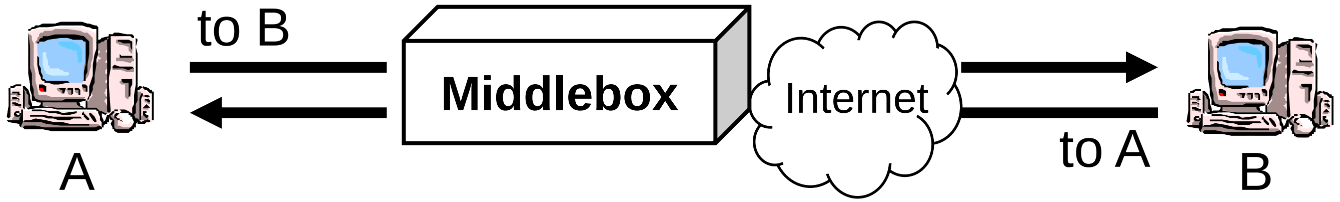
- Zipf‘s Law indicates that there may be items with extreme weight, say „60 % of the users want this file“. Such cases cannot be solved with item balancing alone.

Dealing with Hotspots

- Replica sets
  - A set of nodes is storing an item and any node can answer to queries for reading the item.
  - Basicallly, replica sets are proposed for fault tolerance, but they can be used to relief hot spots. This only works, however, if nodes in the replica set are on paths towards the item ID and are only hit by a fair share of the queries.
    - Leaf set in Pastry, Predecessors on paths in Kademlia are good choices.
    - Successor list in Chord is a bad choice as they are not hit by queries.
- Structural Replication
  - Allow multiple nodes per key-space. Similar to replica sets, but directly built into the system design, e.g. in the DHT P-Grid (Aberer et. al 2001) or CAN.

Technical Issues

Technical Issues – Middleboxes

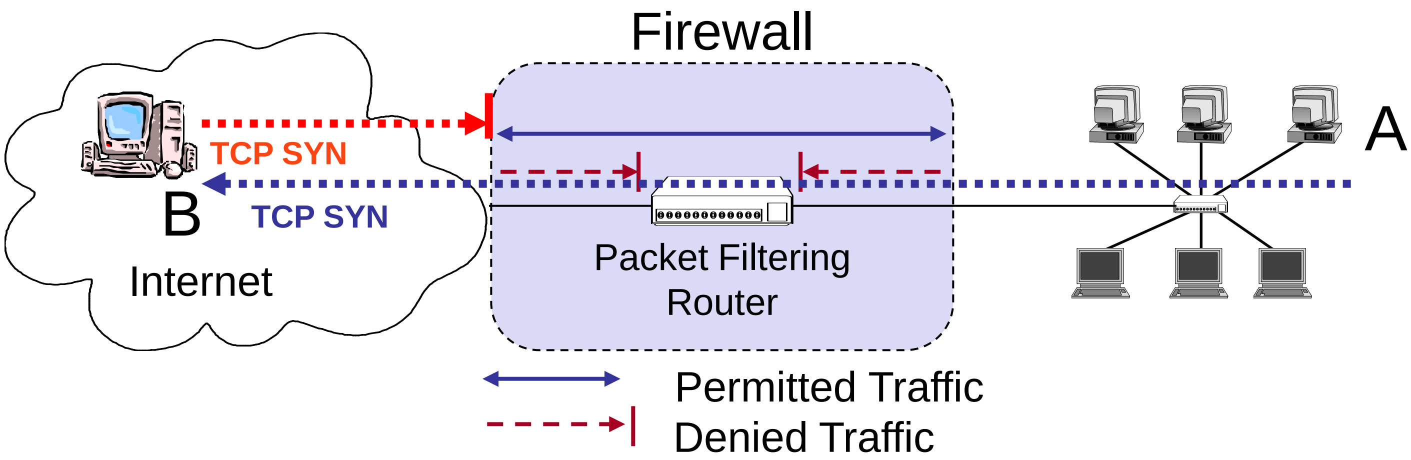


„A middlebox is defined as any intermediary device performing functions other than the normal, standard functions of an IP router on the datagram path between a source host and destination host. “ (RFC 3234, Middleboxes: Taxonomy and Issues, February 2002, www.ietf.org)

Middleboxes

- may prevent hosts on the Internet from connecting to each other
- Firewall
- Network-Adress-Translators (NAT)
- …
- Results for the Kad network [Brunner, 2006]
  - 44 % of peers firewalled or NATed.

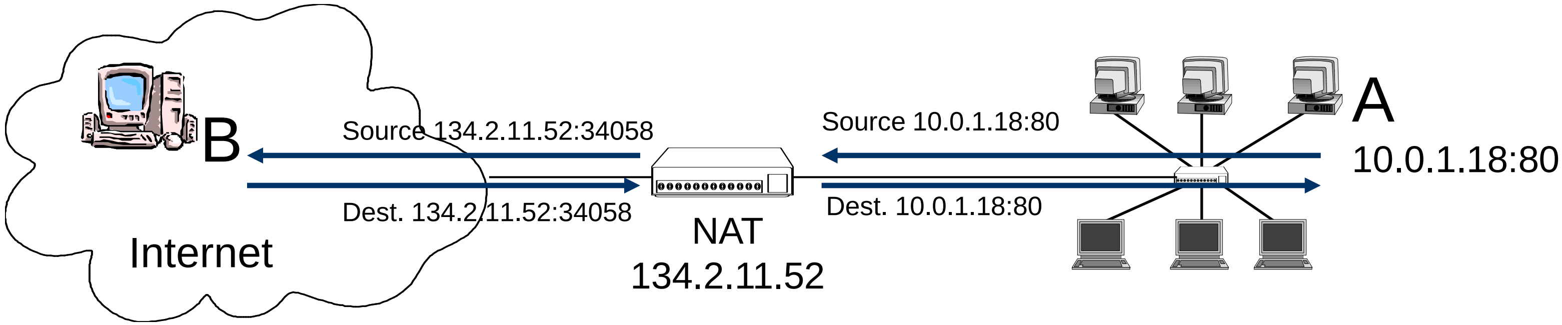
Firewall



Firewalls

- Usually at the edges of a local network firewalls are used to filter traffic according to a set of rules.
  - Rules are usually based on IP:Port combinations, e.g. only allow Port 80 (http) traffic to the webserver (IP)
- Typical problem for Peer-to-Peer
  - Connection establishment may only be allowed from inside to the Internet and not from the Internet to the hosts in the local network.

Network Adress Translator (NAT)



Network Adress Translation

- Typical
  - Local computers have private IP adresses (10/8,192.168/24) which are reserved for usage in local networks and which are not routed by routers. → computers cannot be adressed with their private IP adress
- Adress Translation used to translate local private adresses to global public adresses.
  - usually: Local_IP:Local_Port <-> Public_IP:Public_Port
- Problems
  - Public adress different from machine adress and application port.
  - NATs usually work dynamically, so application or computer are unreachable before they connect and get a public IP:Port pair.

Simple solutions

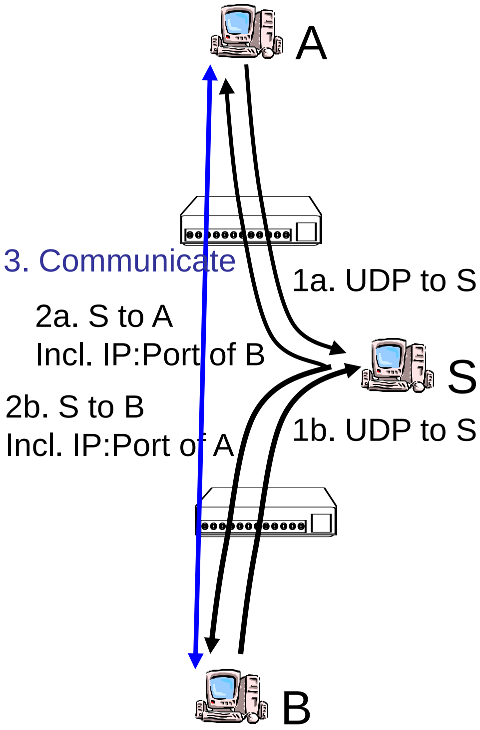
Simple solutions

- PUSH in Gnutella
  - An existing connection is used to signal the request to the firewalled peer, the firewalled peer then initiates the connection.
- In super peer networks
  - Super peer approach
    - Super peers may not be firewalled or behind NAT.
  - Use super peer as relay
- Use reachable peer as relay
  - Peers need to know peers with public IP adresses.
- Hide behind common ports
  - Many firewalls may allow connections to some common ports (HTTP, SMTP, etc.) and block new connections to unknown ports (typically used by P2P).
  - e.g. Skype tends to listen at port 80 if available.

Hole Punching (for NAT Traversal)

UDP Hole Punching

- A and B use a third party S to directly connect to each other.
- Algorithm
  - Let A and B be the two hosts, each in its own private network; N1 and N2 are the two NAT devices; S is a public server with a well-known globally reachable IP address.
  - A and B each begin a UDP conversation with S; the NAT devices N1 and N2 create UDP translation states and assign temporary external port numbers
  - S relays these port numbers back to A and B
  - A and B contact each others' NAT devices directly on the translated ports; the NAT devices use the previously created translation states and send the packets to A and B



- Does not work with all types of NATs, but with the most common ones.
- There is also TCP Hole Punching.

Standards & Protocols (for NAT Traversal)

STUN, TURN, ICE

- IETF standards for NAT Traversal
- Based on Hole Punching and/or Relaying
- Transparent approaches – no interaction with the NAT

UPnP (Universal Plug and Play)

- Developped by Microsoft, very common protocol in home network and consumer devices.
- Interaction with NAT to determine public IP adress, configure port mappings, etc.
  - Many current filesharing network use it to open ports if necessary.
- Insecure, not suitable for larger/company networks

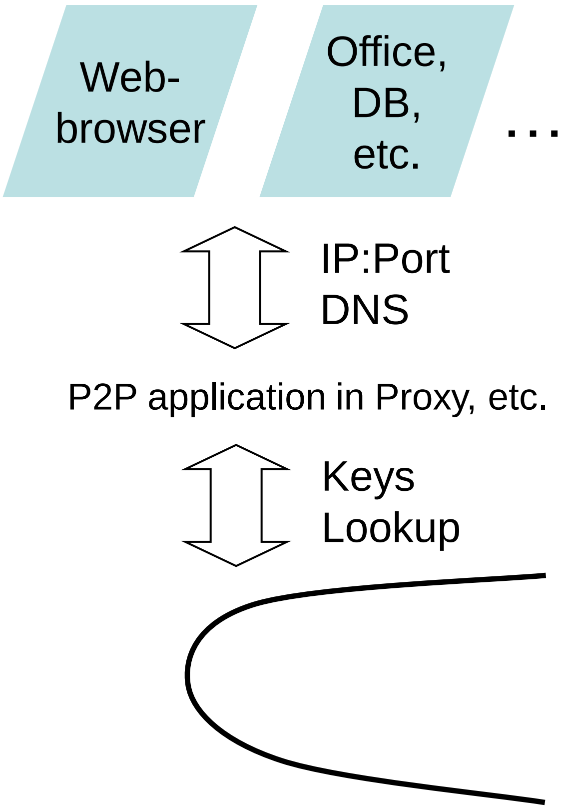
Integration of legacy applications

Legacy Applications

- Applications that are commonly used and that are not developped for the P2P network.
- TCP / UDP sockets used to communicate.
- Resolution of IP adresses via DNS.

Peer-to-Peer for legacy applications

- Accept TCP/UDP traffic with IP:Port adressing
- Intercept DNS to resolve destinations and map IP:Port pairs to keys in the P2P network
  - E.g. based on special non-existing top-level domains, e.g. „`schoenerdienst.p2p`"
- Technical solutions
  - Proxy
    - The application sends the packets to the configured proxy. The proxy then modifies and inserts them into the Peer-to-Peer network.
  - Virtual Network Devices (tun/tap)
    - Similar, but Layer 2 approach.



Literature

- Ananth Rao, Karthik Lakshminarayanan, Sonesh Surana, Richard Karp, Ion Stoica: "Load Balancing in Structured P2P Systems", IPTPS 2003.
- David Karger, Mathias Ruhl: "Simple Efficient Load Balancing for Peer-to-Peer Systems", IPTPS 2004.
- Miguel Castro, Peter Druschel, Y.C. Hu, Anthony Rowstron: "Exploiting Network Proximity in Peer-to-Peer Netwokrs", Technical Report, Microsoft Research, 2002.
- Krishna P. Gummadi, Ramakrishna Gummadi, Steven D. Gribble, Sylvia Ratnasamy, Scott Shenker, and Ion Stoica: „The Impact of DHT Routing Geometry on Resilience and Proximity", Proceedings of the ACM SIGCOMM 2003, Karlsruhe, Germany, August 2003.