Peer-to-Peer Networks

Chapter 1: Introduction

Thorsten Strufe

Note: these slides have been prepared with influence by material of Prof. Jussi Kangasharju

Lecture Outline

- Who are we?
- Course outline
- Organizational matters

- The P2P scenario
- Reprise of DS models
- Reprise of DNS
- P2P in a nutshell
- What‘s also hard?
- History of P2P
- Current state
- The future of P2P

Course Outline and Goals

- Course topic is peer-to-peer systems
- Take a look at current state in P2P systems, both in “real world” and in research work

- What does P2P mean?
- Why does P2P work?
- What makes a good P2P system, how can their quality be evaluated?
- How is the P2P principle reflected in today’s systems?
- What are new challenges / applications for P2P?

Who we are…

- Fachgebiet „Peer-to-Peer Netzwerke“
  - Prof. Thorsten Strufe (Lectures)
    - Piloty A310 (*forget it, write me an email!* ☺)
    - strufe [at] cs.tu-darmstadt.de
  - Stefanie Roos
    - roos[at]cs.tu-darmstadt.de

  - Dominik Fischer
  - Jan-Michael Heller

  - http://www.p2p.tu-darmstadt.de

Organizational matters

- Courses
  - Wed 9.50 – 11.30 (*semi c.t.*)
  - S2|02/C110
- Exercises
  - Tue 14.25 – 16.05
  - S2|02/C110
  - Both written and programming
  - Can be done in groups (of 2 ppl)
- Exams
  - Date: to be announced
  - Hopefully oral exams (if < 32 students)
  - Successfully completing exercises can buy you a bonus

- „The P2P-lecture“
    - http://www.p2p.tu-darmstadt.de/teaching/winter-term-2012/p2p-networks-lecture/
  - Mailing list: p2p-ws12 [at] informatik.tu-darmstadt.de (please subscribe)
  - Forum „D120“ at „Fachschaft“

Material

- Slides will be available on the web site

  (please send emails to push me, there are millions of other things in my head! :-/)

- Literature
  - scholar.google.com/citeseer, „p2p“,“peer-to-peer“,…
  - google (avoid wikipedia, unless you only want a very first impression)
  - Books:
    - Coulouris, Dollimore, Kindberg: „Distributed Systems“
    - Raj Jain: „The art of computer systems performance evaluation“
    - Booth, Colomb, Williams: „The craft of research“
    - Steinmetz/Wehrle „P2P systems“ (free on the web, no close relation)

Questions?

Peer-to-Peer?

- What is it?

- What does the word “peer” mean?
  - Merriam-Webster: one that is of equal standing with another **:** EQUAL; *especially* **:** one belonging to the same societal group especially based on age, grade, or status

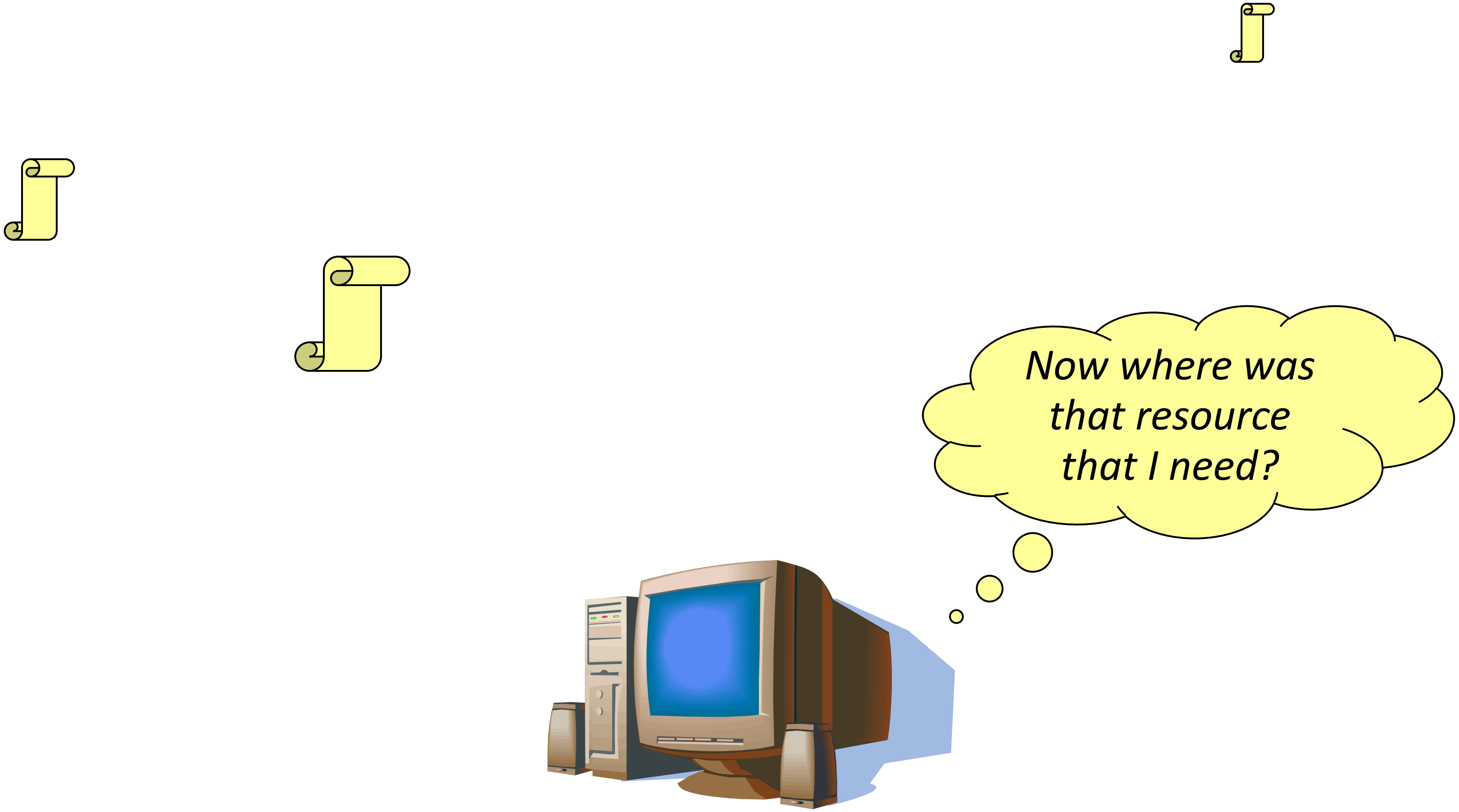
- Peer-to-peer: From one equal partner to another?

How do you define peer-to-peer?

What do you consider to be the main problems?

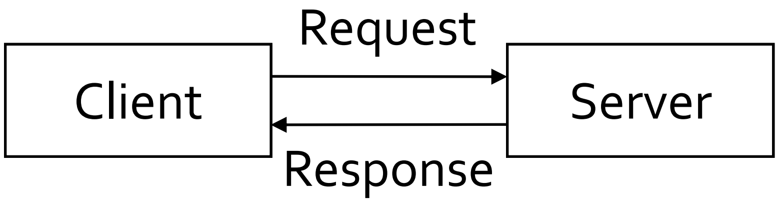
What do you think are the solutions (how does P2P work?)

The Peer-to-Peer Problem



Reprise: The client/server model

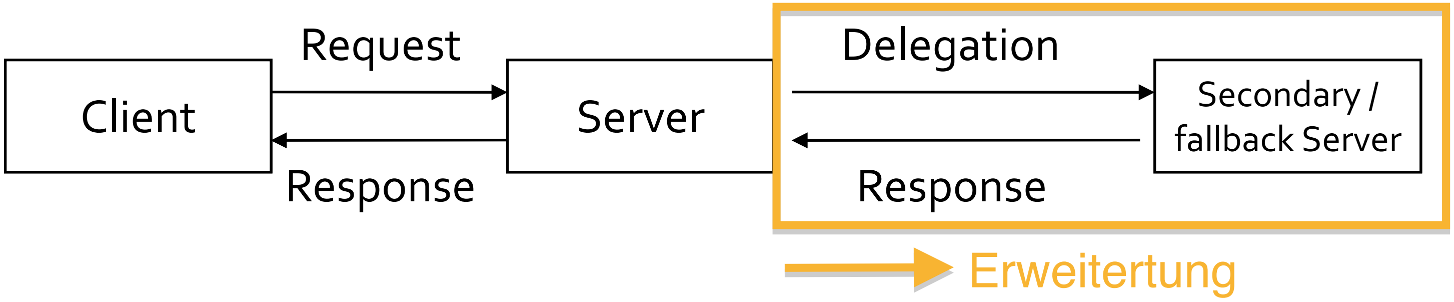
- Communication in request-response pairs
- The **role model** (the roles in the communication) is asymmetric
  - The client arrives and requests a service at any given point in time
  - The server is dedicated to the service and available and it responds immediately (well, after a processing time…)



- The **communication model** is asymmetric, too:
  - The clients send **requests** to a server
  - The server provides a solution and sends the results back in a **response**

Extended Client/Server Model

The client/server model is sometimes extended to more than two roles („client/server with delegation“)



- Definition of the clients remains the same:

  A **client** is, with respect to a considered service, the original and **initial source** of a **request** and the **sink for the response**.

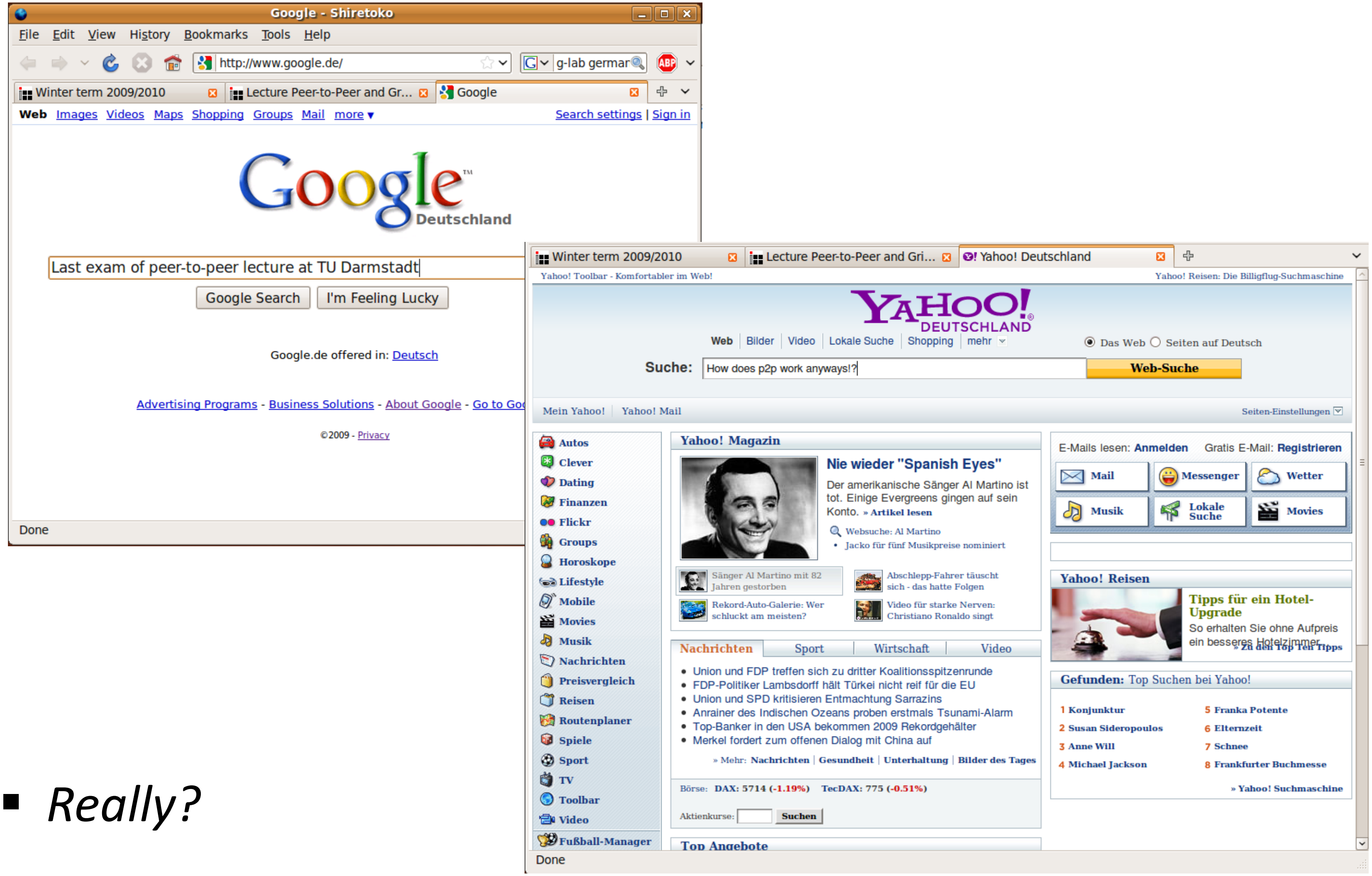
- Sidenote: Delegation may be *recursive* or *iterative*

- ***Beware!:*** *DNS calls or other services /external/ to the system are not part of the model!*

Locating Resources?

So how is it done in the Internet?

Real life resource location



Last exam of peer-to-peer lecture at TU Darmstadt

Google.de offered in: Deutsch

Advertising Programs - Business Solutions - About Google - Go to Goo

©2009 - Privacy

Yahoo! Toolbar - Komfortabler im Web!

Yahoo! Reisen: Die Billigflug-Suchmaschine

● Das Web ○ Seiten auf Deutsch

Suche: How does p2p work anyways!?

Yahoo! Magazin

Nie wieder "Spanish Eyes"

Der amerikanische Sänger Al Martino ist tot. Einige Evergreens gingen auf sein Konto. » Artikel lesen

Websuche: Al Martino

- Jacko für fünf Musikpreise nominiert

Sänger Al Martino mit 82 Jahren gestorben

Abschlepp-Fahrer täuscht sich - das hatte Folgen

Rekord-Auto-Galerie: Wer schluckt am meisten?

Video für starke Nerven: Christiano Ronaldo singt

- Union und FDP treffen sich zu dritter Koalitionsspitzenrunde
- FDP-Politiker Lambsdorff hält Türkei nicht reif für die EU
- Union und SPD kritisieren Entmachtung Sarrazins
- Anrainer des Indischen Ozeans proben erstmals Tsunami-Alarm
- Top-Banker in den USA bekommen 2009 Rekordgehälter
- Merkel fordert zum offenen Dialog mit China auf

» Mehr: Nachrichten | Gesundheit | Unterhaltung | Bilder des Tages

Börse: DAX: 5714 (-1.19%) TecDAX: 775 (-0.51%)

Aktienkurse:

Top Angebote

E-Mails lesen: Anmelden Gratis E-Mail: Registrieren

Yahoo! Reisen

Tipps für ein Hotel-Upgrade

So erhalten Sie ohne Aufpreis ein besseres Hotelzimmer. » Zu den Top Ten Tipps

Gefunden: Top Suchen bei Yahoo!

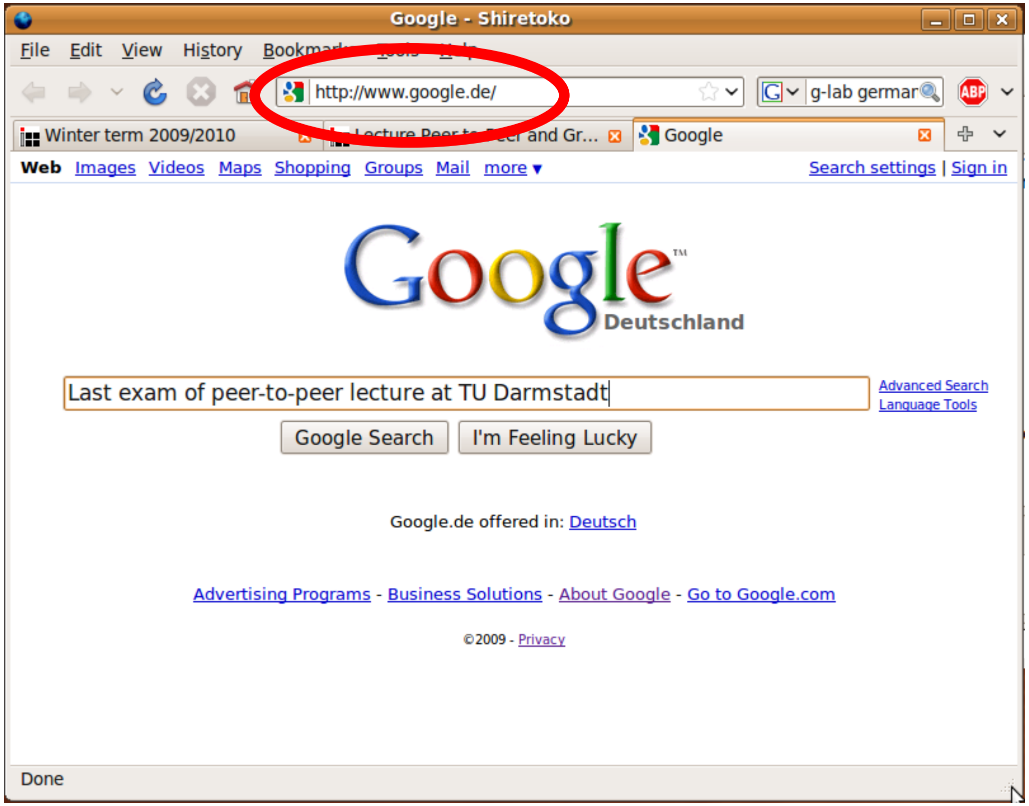
1 Konjunktur	5 Franka Potente
2 Susan Sideropoulos	6 Elternzeit
3 Anne Will	7 Schnee
4 Michael Jackson	8 Frankfurter Buchmesse

» Yahoo! Suchmaschine

- *Really?*

Reprise2: DNS – The Domain Name System

- Naming Service for (almost all) Internet traffic
- Lookup of (resolve)
  - Host-Addresses
  - Mail-Servers
  - Alias Names
  - Alternative Name Servers
  - …
- Distributed Database consisting of multitude of servers



http://www.google.de/

Last exam of peer-to-peer lecture at TU Darmstadt

DNS – Names

People: many identifiers:

- SSN, name, passport #

Internet hosts, routers:

- IP address (32 bit) - used for addressing datagrams
- “Name”, e.g., www.yahoo.com - used by humans

Q: Map between IP addresses and name ?

Domain Name System:

Distributed database implemented in hierarchy of many *name servers*

Application-layer protocol: hosts, routers, name servers communicate to *resolve* names (address/name translation)

- Note: core Internet function, implemented as application-layer protocol
- Complexity at network’s “edge”

DNS – what does it do?

DNS services

- Hostname to IP address translation
- Host aliasing
  - Canonical and alias names
- Mail server aliasing
- Load distribution
  - Replicated Web servers: set of IP addresses for one canonical name

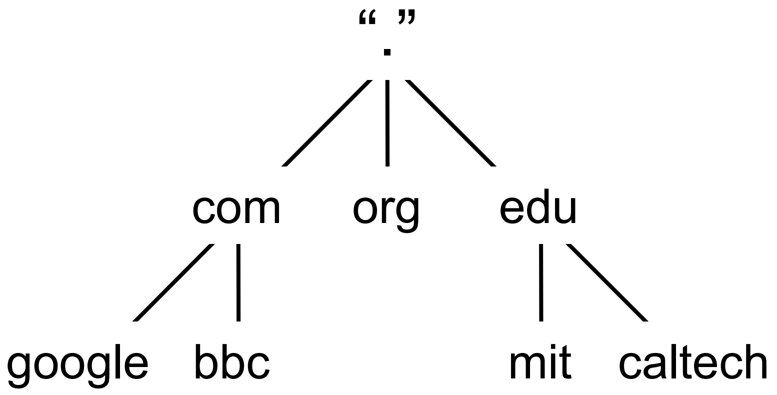
Why not centralize DNS?

- Single point of failure
- Traffic volume
- Distant centralized database
- Maintenance

does not *scale!*

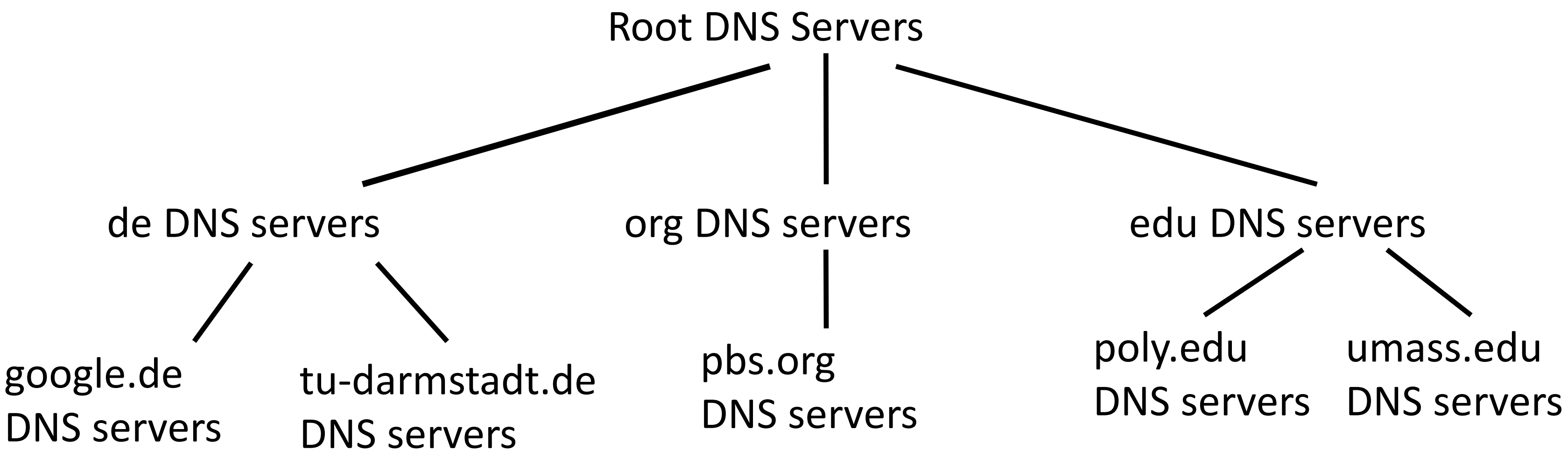
What does this „it scales“ mean anyways!?

DNS – Data Organization: Domains / Zones



- Structured Namespace
- Hierarchical organization in sub domains/zones
- Sourced at “root zone” (“.”)
- Parent zones maintain pointers to child zones (*“zone cuts”*)
- Zone data is stored as “Resource Records” (RR)

Distributed, Hierarchical Database

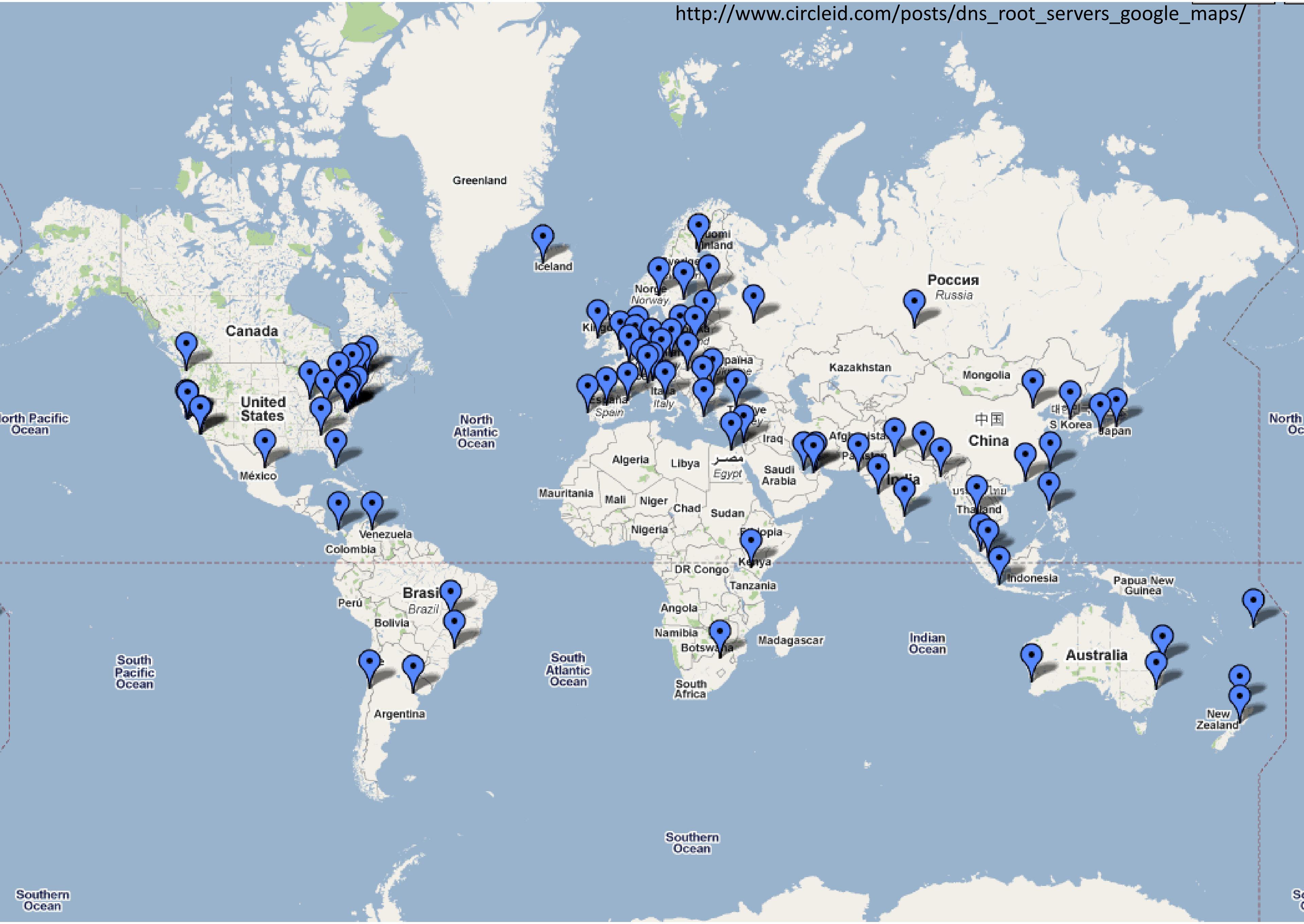


Client wants IP for www.p2p.tu-darmstadt.de; 1st approx:

- Client queries a root server to find **de** DNS server
- Client queries de DNS server to get **tu-darmstadt.de** DNS server
- Client queries tu-darmstadt.de DNS server to get IP address for www.p2p.tu-darmstadt.de

DNS: Root Name Servers

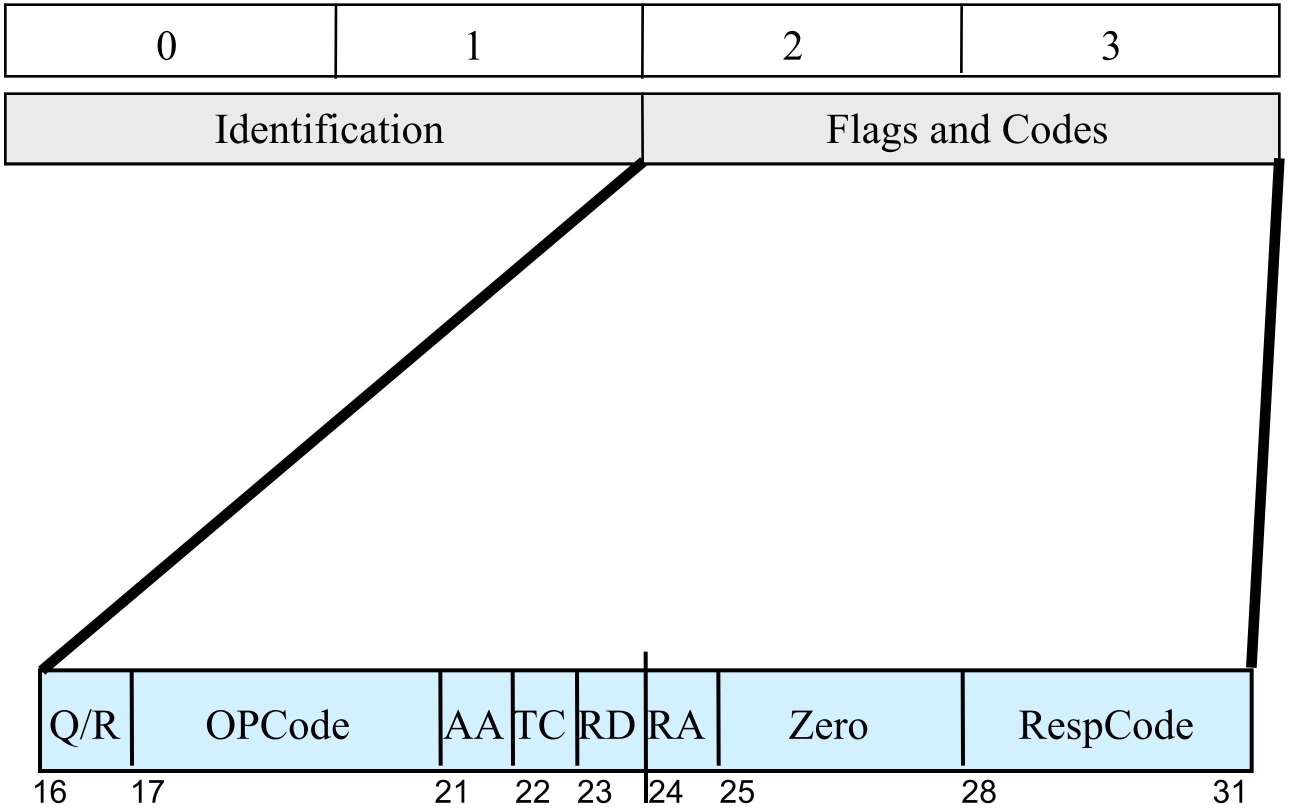
So, how many root nameservers are there actually? (physically) http://www.circleid.com/posts/dns_root_servers_google_maps/



DNS – Components

- Authoritative Server
  - Server maintaining authoritative content of a complete DNS zone
  - Top-Level-Domain (TLD) servers & auth servers of organization’s domains
  - Pointed to in parent zone as authoritative
  - Possible load balancing: master/slaves
- Recursive (Caching) Server
  - Local proxy for DNS requests
  - Caches content for specified period of time (soft-state with TTL)
  - If data not available in the cache, request is processed recursively
- Resolver
  - Software on client’s machines (part of the OS)
  - Windows-* and *nix: Stub resolvers
    - Delegate request to local server
    - Recursive requests only, no support for iterative requests

DNS – Message Format



- Q/R *Query/Response Flag*
- *Operation Code*
- AA *Auth. Answer Flag*
- TC *Truncation Flag*
- RD *Recursion Desired Flag*
- RA *Recursion Available Flag*
- Zero (three resv. bits)
- *Response Code*

DNS – Header Fields

- *Identifier:* a 16-bit identification field generated by the device that creates the DNS query. It is copied by the server into the response, so it can be used by that device to match that query to the corresponding reply
- *Query/Response Flag:* differentiates between queries and responses (0 ~ Query, 1 ~ Response)
- *Operation Code:* specifies the type of message (Query, Status, Notify, Update)
- *Authoritative Answer Flag (AA):* set to1 if the answer is authoritative
- *Truncation Flag:* When set to 1, indicates that the message was truncated due to its length (might happen with UDP, requestor can then decide to ask again with TCP as transport service)
- *Recursion Desired:* set to 1 if requester desired recursive processing
- *Recursion Available:* set to 1 if server supports recursive queries

TLD, Authoritative and Local DNS Servers

- Top-level domain (TLD) servers:
  - responsible for com, org, net, edu, etc, and all top-level country domains uk, fr, ca, jp
  - Network solutions maintains servers for **`com`** TLD
  - Educause for **`edu`** TLD
- Authoritative DNS servers:
  - organization's DNS servers, providing authoritative hostname to IP mappings for organization's servers (e.g., Web and mail).
  - Can be maintained by organization or service provider
- Local DNS servers:
  - Does not strictly belong to hierarchy
  - Each ISP (residential ISP, company, university) has one
    - Also called "default name server"
  - When a host makes a DNS query, query is sent to its local DNS server
    - Acts as a proxy, forwards query into hierarchy

DNS – Resource Records

- Atomic entries in DNS are called “Resource Records” (RR)
- Format:

  <name> [<ttl>] [<class>] <type> <rdata>

  - `name` (domain name of resource)
  - `ttl` (Time-to-live)
  - `class` (used protocol): IN (Internet), CH (Chaosnet)…
  - `type` (record type): A (Host-Address), NS (Name Server), MX (Mail Exchange), CNAME (Canonical Name), AAAA (IPv6-Host-Address), DNAME (CNAME, IPv6)
  - `rdata` (resource data): Content! (What did we want to look up?)

DNS Records

DNS: Distributed DB storing resource records (RR)

RR Format: **(name, value, type, ttl)**

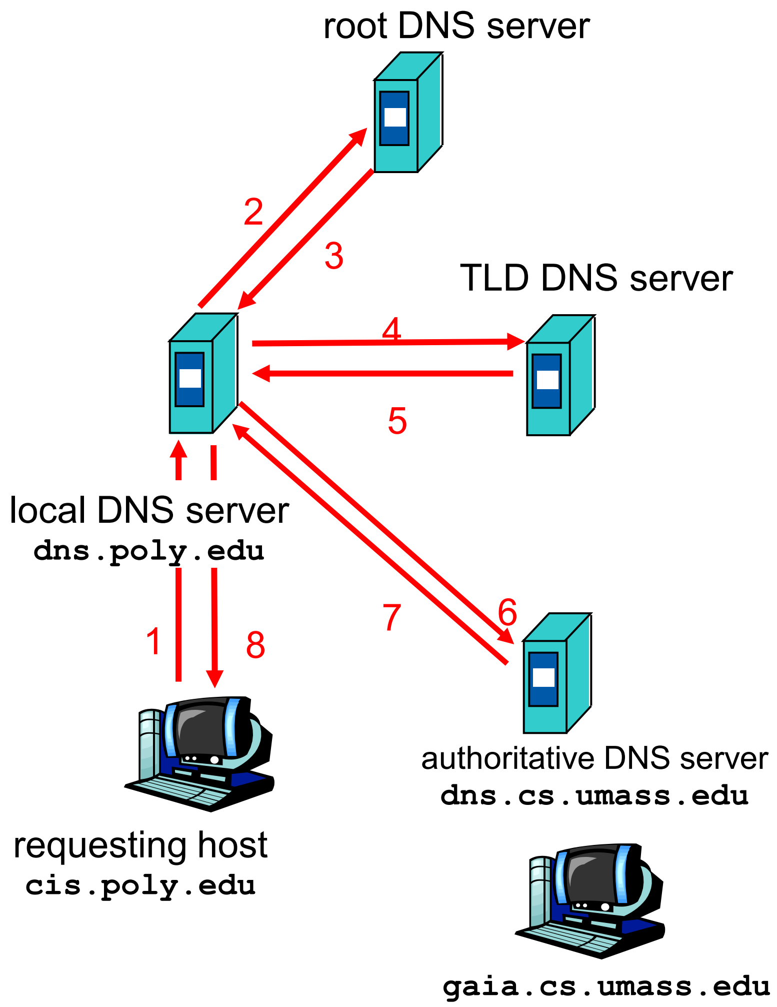
- Type=A
  - **`name`** is hostname
  - **`value`** is IP address
- Type=NS
  - **`name`** is domain (e.g. foo.com)
  - **`value`** is IP address of authoritative name server for this domain
- Type=MX
  - **`value`** is name of mailserver associated with **`name`**
- Type=CNAME
  - **`name`** is alias name for some "canonical" (the real) name

    `www.ibm.com` is really

    `servereast.backup2.ibm.com`
  - **`value`** is canonical name

DNS – Example of Iterative Queries

- Host at cis.poly.edu wants IP address for **`gaia.cs.umass.edu`**



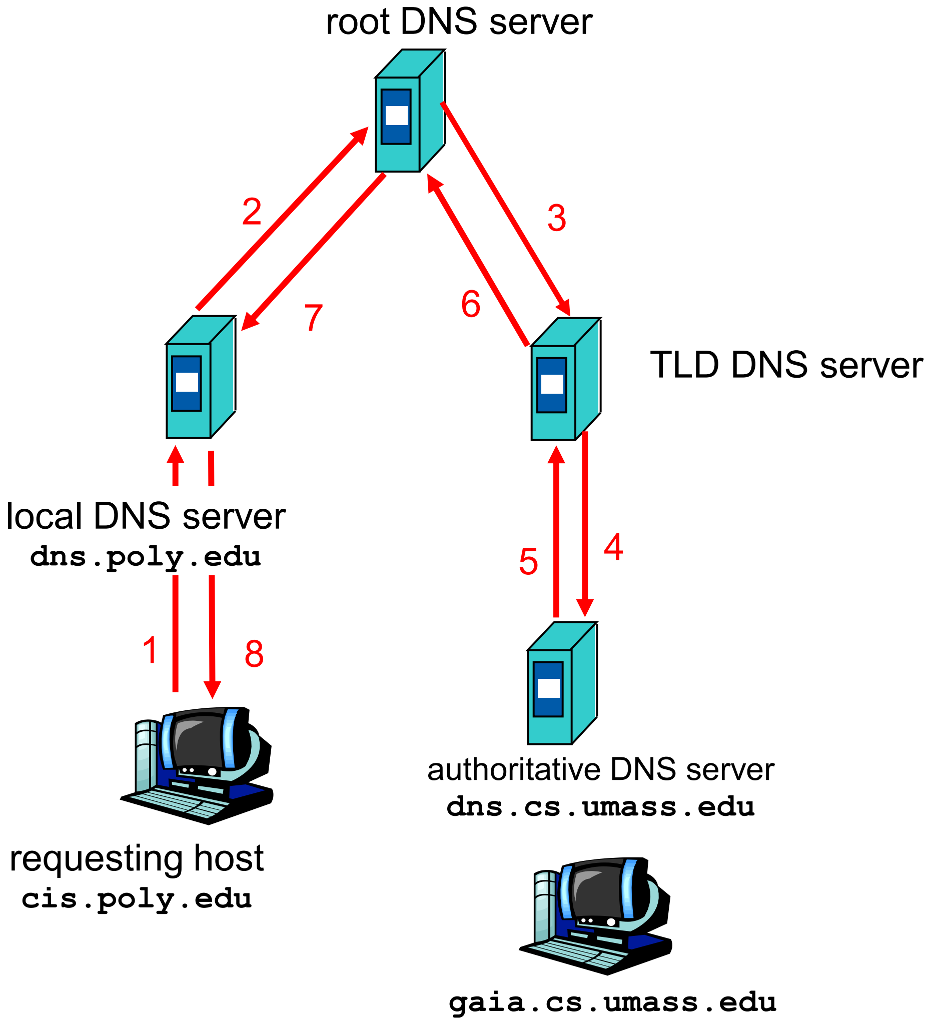
DNS – Recursive Queries

Recursive query:

- Puts burden of name resolution on contacted name server
- Heavy load?

Iterated query:

- Contacted server replies with name of server to contact
- “I don’t know this name, but ask this server”



DNS: Caching and Updating Records

- Once (any) name server learns mapping, it *caches* mapping
  - Cache entries timeout (disappear) after some time
  - TLD servers typically cached in local name servers
    - Thus root name servers not often visited

- Update/notify mechanisms under design by IETF
  - RFC 2136
  - http://www.ietf.org/html.charters/dnsind-charter.html

Inserting Records Into DNS

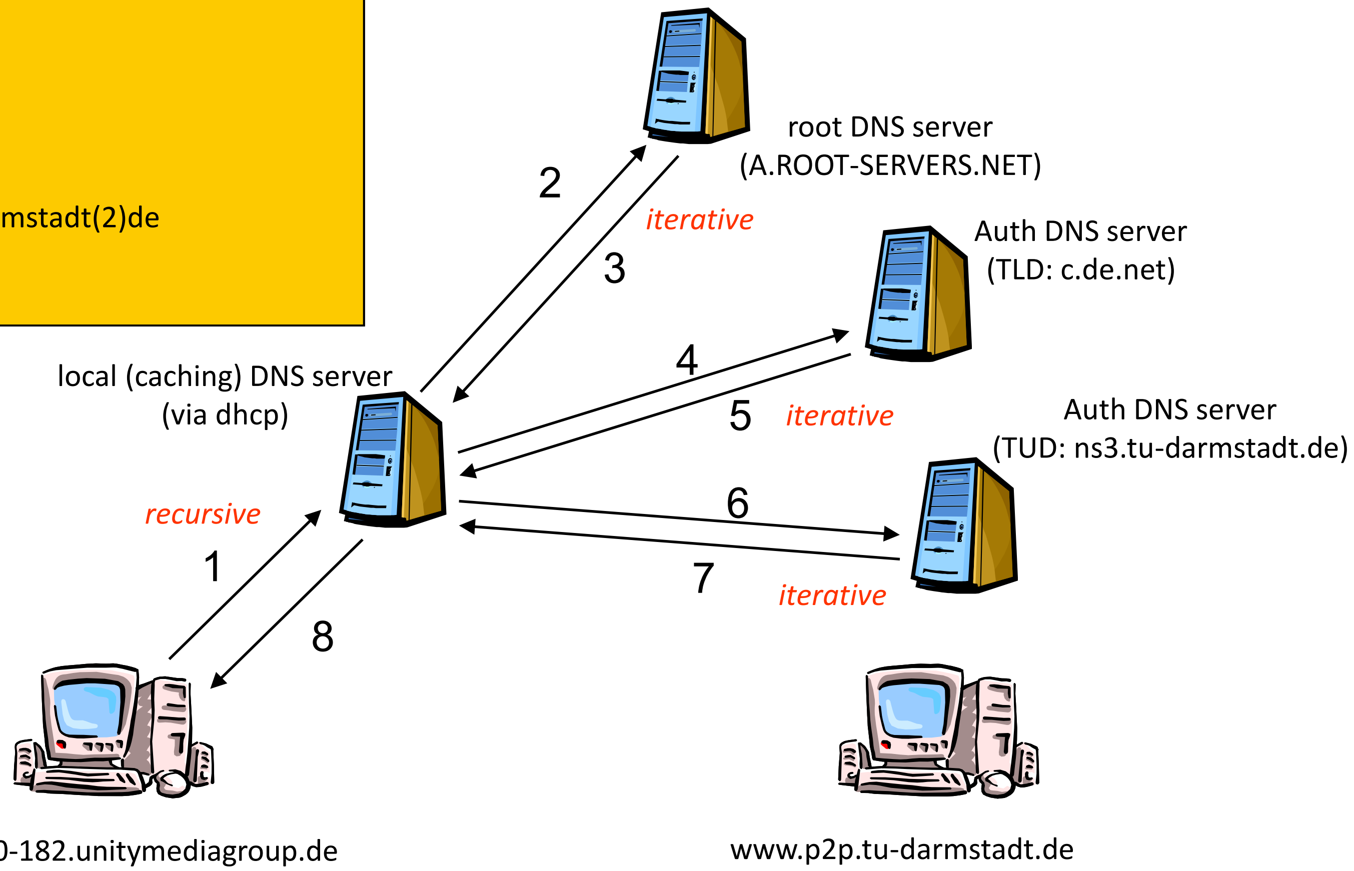
- Example: just created startup “Network Utopia”
- Register name networkutopia.com at a registrar (e.g., Network Solutions)
  - Need to provide registrar with names and IP addresses of your authoritative name server (***primary*** and ***secondary***)
  - Registrar inserts two RRs into the com TLD server:

```
(networkutopia.com, dns1.networkutopia.com, NS)
(dns1.networkutopia.com, 212.212.212.1, A)
```

- Put in authoritative server Type A record for www.networkuptopia.com and Type MX record for networkutopia.com

DNS – Recursive and Iterative Queries

```
S HEADER (send)
lentifier:          0x3116
ags:               0x00 (Q )
pcode:              0 (Standard query)
eturn code:         0 (No error)
umber questions:    1
umber answer RR:    0
umber authority RR: 0
umber additional RR: 0
JESTIONS (send)
ueryname:           (3)www(3)p2p(12)tu-darmstadt(2)de
ype:               1 (A)
lass:              1 (Internet)
```



A Quick Example…

```
strufe@eris:~$ dnstracer -v www.p2p.tu-darmstadt.de
Tracing to informatik.tu-darmstadt.de[a] via 130.83.163.141, maximum of 3 retries
130.83.163.141 (130.83.163.141) IP HEADER
-Destination address:  130.83.163.141
-DNS HEADER (send)
-- Identifier:         0x3116
-- Flags:              0x00 (Q )
-- Opcode:             0 (Standard query)
-- Return code:        0 (No error)
-- Number questions:   1
-- Number answer RR:   0
-- Number authority RR: 0
-- Number additional RR: 0
-QUESTIONS (send)
-- Queryname:          (3)www(3)p2p(12)tu-darmstad
-- Type:               1 (A)
-- Class:              1 (Internet)
-DNS HEADER (recv)
-- Identifier:         0x3116
-- Flags:              0x8080 (R RA )
-- Opcode:             0 (Standard query)
-- Return code:        0 (No error)
-- Number questions:   1
-- Number answer RR:   2
-- Number authority RR: 0
-- Number additional RR: 0
-…….
```

```
QUESTIONS (recv)
- Queryname:         (3)www(3)p2p(12)tu-darmstadt(2)de
- Type:              1 (A)
- Class:             1 (Internet)
ANSWER RR
- Domainname:        (6)charon(7)dekanat(10)informatik(12)tu-darmstadt(2)de
- Type:              1 (A)
- Class:             1 (Internet)
- TTL:               1592 (26m32s)
- Resource length:   4
- Resource data:     130.83.162.6
ANSWER RR
- Domainname:        (3)www(3)p2p(12)tu-darmstadt(2)de
- Type:              5 (CNAME)
- Class:             1 (Internet)
- TTL:               49817 (13h50m17s)
- Resource length:   28
- Resource data:     (6)charon(7)dekanat(10)informatik(12)tu-darmstadt(2)de
Got answer [received type is cname]
```

So where is the Info?

```
strufe@eris:~$  dnstracer -v -qns tu-darmstadt.de
Tracing to tu-darmstadt.de[ns] via 130.83.163.130
130.83.163.130 (130.83.163.130) IP HEADER
- Destination address:  130.83.163.130
DNS HEADER (send)
- Identifier:          0x4C45
- Flags:               0x00 (Q )
- Opcode:              0 (Standard query)
- Return code:         0 (No error)
- Number questions:    1
- Number answer RR:    0
- Number authority RR: 0
- Number additional RR: 0
QUESTIONS (send)
- Queryname:           (12)tu-darmstadt(2)de
- Type:                2 (NS)
- Class:               1 (Internet)
DNS HEADER (recv)
- Identifier:          0x4C45
- Flags:               0x8080 (R RA )
- Opcode:              0 (Standard query)
- Return code:         0 (No error)
- Number questions:    1
- Number answer RR:    5
- Number authority RR: 0
- Number additional RR: 9

…….
```

```
QUESTIONS (recv)
- Queryname:           (12)tu-darmstadt(2)de
- Type:                2 (NS)
- Class:               1 (Internet)
ANSWER RR
- Domainname:          (12)tu-darmstadt(2)de
- Type:                2 (NS)
- Class:               1 (Internet)
- TTL:                 70523 (19h35m23s)
- Resource length:     6
- Resource data:       (3)ns1(3)hrz(12)tu-darmstadt(2)de
ANSWER RR
- Domainname:          (12)tu-darmstadt(2)de
- Type:                2 (NS)
- Class:               1 (Internet)
- TTL:                 70523 (19h35m23s)
- Resource length:     5
- Resource data:       (2)ns(6)man-da(2)de
ANSWER RR
- Domainname:          (12)tu-darmstadt(2)de
- Type:                2 (NS)
- Class:               1 (Internet)
- TTL:                 70523 (19h35m23s)
- Resource length:     6
- Resource data:       (3)ns2(3)hrz(12)tu-darmstadt(2)de

…….
```

Answer ctd…

```
…….
ADDITIONAL RR
- Domainname:        (3)ns1(3)hrz(12)tu-darmstadt(2)de
- Type:              1 (A)
- Class:             1 (Internet)
- TTL:               17335 (4h48m55s)
- Resource length:   4
- Resource data:     130.83.22.63
ADDITIONAL RR
- Domainname:        (2)ns(6)man-da(2)de
- Type:              28 (unknown)
- Class:             1 (Internet)
- TTL:               38386 (10h39m46s)
- Resource length:   16
- Resource data:     2001:41b8:0000:0001:0000:0000:0000:0053
ADDITIONAL RR
- Domainname:        (2)ns(6)man-da(2)de
- Type:              1 (A)
- Class:             1 (Internet)
- TTL:               38386 (10h39m46s)
- Resource length:   4
- Resource data:     82.195.66.249
ADDITIONAL RR
- Domainname:        (3)ns2(3)hrz(12)tu-darmstadt(2)de
- Type:              28 (unknown)
- Class:             1 (Internet)
- TTL:               17335 (4h48m55s)
- Resource length:   16
- Resource data:     2001:41b8:083f:0022:0000:0000:0000:0063
…….
```

```
…….
ADDITIONAL RR
- Domainname:        (3)ns2(3)hrz(12)tu-darmstadt(2)de
- Type:              1 (A)
- Class:             1 (Internet)
- TTL:               17335 (4h48m55s)
- Resource length:   4
- Resource data:     130.83.22.60
ADDITIONAL RR
- Domainname:        (3)ns2(6)man-da(2)de
- Type:              1 (A)
- Class:             1 (Internet)
- TTL:               38386 (10h39m46s)
- Resource length:   4
- Resource data:     217.198.242.225
ADDITIONAL RR
- Domainname:        (3)ns3(3)hrz(12)tu-darmstadt(2)de
- Type:              28 (unknown)
- Class:             1 (Internet)
- TTL:               17335 (4h48m55s)
- Resource length:   16
- Resource data:     2001:41b8:083f:0056:0000:0000:0000:0060
ADDITIONAL RR
- Domainname:        (3)ns3(3)hrz(12)tu-darmstadt(2)de
- Type:              1 (A)
- Class:             1 (Internet)
- TTL:               17335 (4h48m55s)
- Resource length:   4
- Resource data:     130.83.56.60
Got answer
```

DNS – Lessons Learned

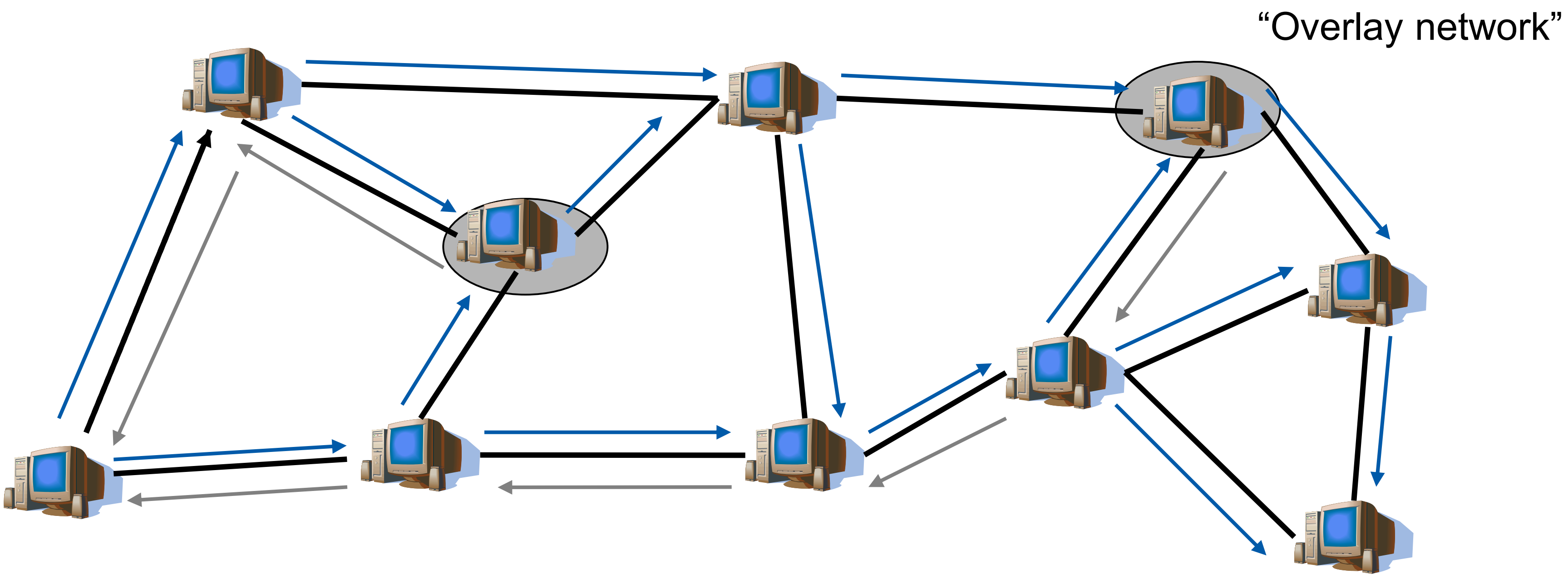
1. Structure name space (divide et impera)
2. Simple „routing“ b/c of structured (hierarchical) namespace

1. Store information at multiple locations
2. Maintain multiple connections
3. Be redundant! (Replicate…)
   - primary and secondary server, multiple TLD servers

4. Delegation using iterative or recursive forwarding
   (Btw: what are the pros and cons of each?)

What does this „routing“ mean anyways!?

Back to P2P!



P2P Environment and Problems

- Properties of (pure) P2P: *„All peers are equal“*
  - no dedicated service, no central entity
  - no a-priori knowledge / structure / hierarchy
  - highly dynamic behavior of nodes

  → Flat system architecture, flat namespace, unreliable service providers

What‘s a Name (or ID, Reference?)

- Main primary problems of P2P:
  - Staying connected
  - Discovery/resource lookup (name resolution, location of replica, especially selecting a good next hop for the delegation → routing)
  - Can‘t trust anyone

P2P in a Nutshell

- *A P2P overlay is a subset of links of a clique*
- Resources are unreliable, yet plentiful

- Standard Solutions (p2p – the executive summary)
  - ***Connectivity****:* select enough fall-back „servers“
  - ***Name resolution****:* unstructured P2P (flooding) or external search engine
  - ***Resource location****:* registry and lookup in structured P2P (DHT!) (or the above…)

  - We can introduce ***identifiers*** based on distributed algorithms (hashes)
  - We can introduce ***structure*** using distributed algorithms (supernodes, etc.)

Peer-to-Peer (a Definition by Models)

- **Communication model:** asynchronous (request-response)

- **Role model:** a single role (?)
  - symmetric behavior, all peers in general (can) do the same
  - ***BUT***: considering an interaction there is one requesting and *n* responding peers.

- **Organisational model:** completely unstructured (*„it‘s a mess!“*)
  - Other than bootstrapping no knowledge whatsoever about the context, no knowledge about the structure

- No **Identifiers**, only names

- *Closely related fields*
  - *DTN, WSN (Ad-hoc-, opportunistic-, pocket-switched-, vehicular-,…, networks)*
  - *Epidemic-, Content-/ Context-based routing, Named Data Networking*

Some Peer-to-Peer Requirements (Terminology)

- A property and its implications:
  - ***Abundance of unreliable resources***

- Functional requirements: discover *and* retrieve
  - Later on extended to services of implemented applications
    - Messaging
    - Session establishment/initiation
    - Content/information dissemination
    - Service migration/placement
    - …

- ***Let's define a few things, before introducing non-functional requirements…***

Terminology and Concepts

- ***Nodes, Peers, Servents, Participants, Users, Individuals***

- ***Service:*** functionality an individual (utilizing entity) is using
- ***Protocol:*** specification according to which cooperating entities communicate in order to provide a service

- ***Distribution vs. Decentralization***

- ***Assumptions:*** expected behavior (actions and reactions) of system components and environment

Terminology and Concepts (Failures)

- ***Churn*** (*exact*): „rate at which individuals are leaving a collective“
  - Commonly: Arrival and departure of nodes (usually: rate)

- ***Failures:*** unintended deviation from expected behavior

- ***Failures***
  - *Component*: Link vs node failures (communication vs. Process)
  - *Type*: Crash (fail-stop, fail-resume), ommission, timing
  - *Time*: Intermittent, permanent
  - ***Here usually: crashing (silently leaving) node***

Terminology and Concepts (Attacks)

- ***Attack***: intended (malicious) deviation from assumption for a purpose

- **Adversaries**: entities willingly deviating from assumptions
  - ***Scope:*** global/local
  - ***Participation:*** internal/external
  - ***Power:*** capabilities, resources

- ***Attack Surface:*** explicit and implicit interfaces exposed to adversaries (within *service*, *protocol*, *underlying infrastructure*)

- ***Adversities***: union of (assumed) stress, failures and attacks

Non Functional Requirements for P2P Systems

- Non functional requirements
  - **Scalability**
    - Mainly: complexity of signalling overhead
  - **Robustness**
    - ***Churn*** and errors do not affect the service provision
  - **Resilience**
    - Service remains available/useable (potentially with **graceful degradation**) under ***attack***
  - **Resistance**
    - Service degradation through attacks is minimized

Properties of P2P Systems

- P2P systems typically have the following properties:

1. Unreliable, uncoordinated, unmanaged
   - No central authority, peers are completely independent
   - Increases flexibility of individual peers, but makes the overall system (possibly) unreliable
2. Resilient to attacks, heterogeneous
   - Large number of peers in the system, hard to bring it down?
   - Heterogeneous peers make viruses and worms harder to write?
3. Large collection of resources
   - Voluntary participation, global reach
   - Millions of simultaneous users

History of P2P

- What was the first P2P system and when?
- Answer: ARPANet 1969
- Later: USENET, 1979 (also FidoNet 1984, other BBSs)
  - Current Internet routing (BGP) is P2P
- The term P2P was coined by Napster in 1999
- Napster was a huge hit, brought P2P to general attention
- Illegal sharing of copyrighted material by users was the main driver behind Napster's success and the reason for its downfall
- Other systems followed Napster quickly, based on other design choices
- Research community followed suit quickly
  - Many deployed systems proprietary, hard to examine well…

Current State of P2P

- Where are we now?
- P2P networks going strong, all over the world
  - Many networks highly popular and widely used
  - Different networks in different countries
- P2P networks currently mostly used for illegal sharing of copyrighted material
  - Music, videos, software, …
  - Note: Can be used for legal sharing too (see BitTorrent)
- Other applications starting to emerge (see below)
- Content providers not so happy
  - Sue companies making P2P software (e.g., Napster), sue software developers (Winny), sue users sharing material
  - But also providing alternate means: iTunes & friends

Newer P2P Systems

- File sharing was first P2P application
- Other applications are coming to light
- BitTorrent more content distribution than file sharing
- P2P extending beyond file sharing: Skype
  - We will look at Skype closer in Chapter 2
- Skype is a P2P telephone “system”
  - Can call other computers, or normal phones
- Skype is based on the KaZaA network (see Chapter 2)
- Similar to VoIP services (e.g., Vonage), but fully based on the individual peers
  - Skype requires a computer, VoIP services often do not
- Using resources: ***Games, Video Streaming***; Controlling data: ***OSN***;

P2P: Some Statistics

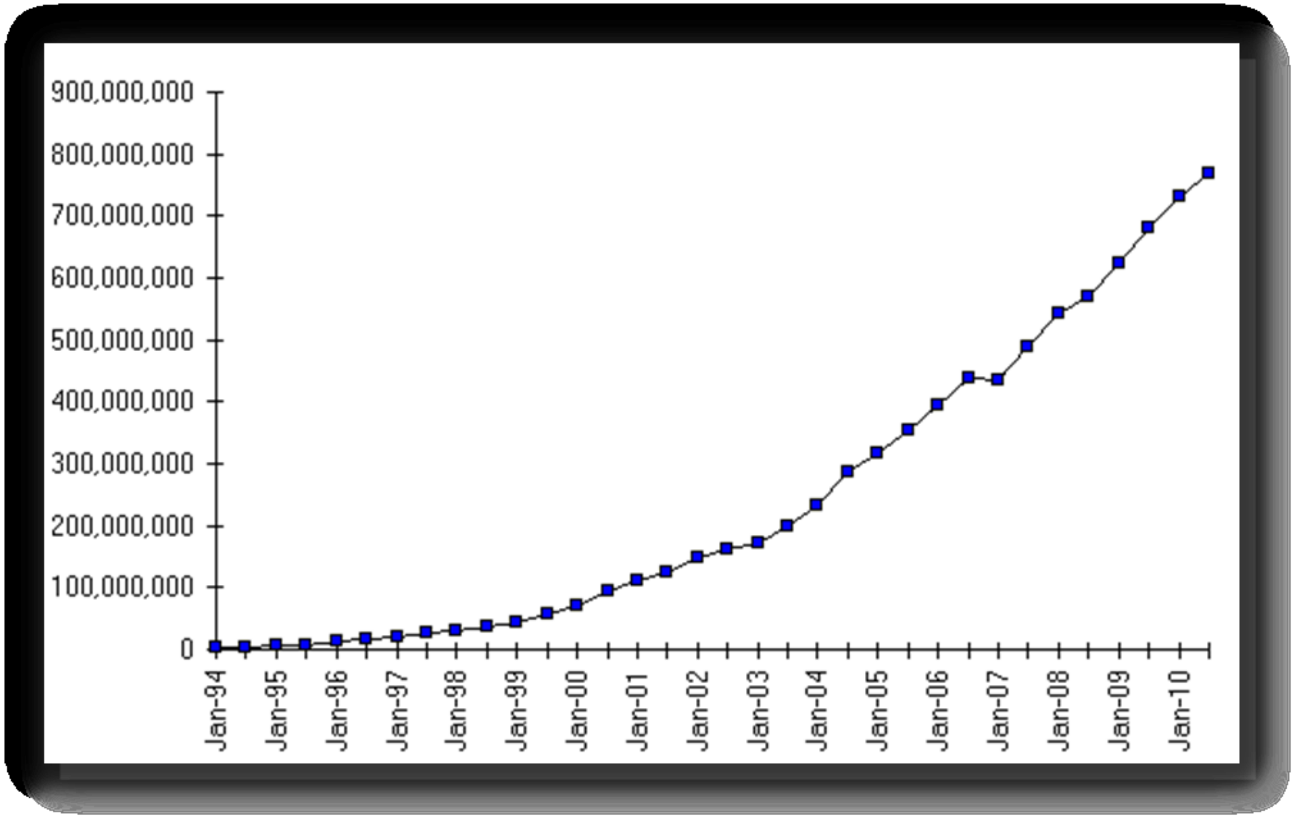
- Currently P2P accounts for 40% (*) of network traffic
  - A bit different in different networks
  - Hard to measure accurately
- Network providers (ISP) not too happy about this
  - But: Often traffic internal to ISP! (e.g., T-Com)
- Some numbers: (take with a grain of salt…)
  - KaZaA had 60 million users total, 1-5 million online at any time
    - 85 million downloads/day
    - Software downloaded over 230 million times
  - Google has 1 billion unique users / month, 400 million queries/day (50mio/200mio in 2006)
  - Skype has over 300 million users, over 20 million concurrently

(*) over 70% including file hosters and usenet

P2P: Some Statistics

- Currently P2P accounts for 40% (*) of network traffic
  - A bit different in different networks
  - Hard to measure accurately
- Network provide
  - But: Often traffi
- Some numbers:
  - KaZaA had 60 m
    - 85 million do
    - Software do
  - Google has 100

    (50mio/200mio
  - Skype has over



(*) over 70% including file hosters and usenet

Why Does P2P Work?

- Why are P2P file sharing networks so successful?

1. Easy to use
   - P2P software readily available, simple to use
2. Provide something useful (for free)
   - Until recently, only alternative to P2P content was “buy a CD”
   - Online music stores may change this?
3. Anyone can contribute
   - Contributions not tied to geographical location; user in Brazil can provide files for everyone (compare with ad hoc networks!)
   - Enough “altruistic” users to make P2P networks useful

- Some systems (Skype) completely hide the P2P-part
  - Will this become the future trend?

P2P: Traps and Pitfalls

- What could render current P2P networks useless?
  - In particular, file sharing networks

1. Removal of desirable content
   - Stricter enforcement of copyright laws?
2. Alternative ways of getting same content
   - Online music stores?
3. Blocking of P2P traffic by ISPs
   - Or making users pay for bandwidth they use?
4. Viruses or worms on P2P networks
   - Exploit bugs in P2P software
5. Frighten the users away…

When P2P and When Not P2P?

- So, when is P2P the right solution?
- Or, when is P2P the wrong solution?
- *Claim:* A general P2P vision is technically feasible
  - In other words, possible to build everything on Internet without any dedicated servers
- Gotcha: Just because it's technically feasible, it doesn't necessarily make sense…
- In other words, just because we can do it P2P, doesn't mean that we should do it P2P
  - True in many areas of life…
- So, when *is* P2P the right solution?!?

Some Criteria

- Let's consider the following criteria

1. Budget
   - How much money do we have?
2. Resource relevance
   - How widely are resources interesting to users?
3. Trust
   - How much trust is there between users?
4. Rate of system change
   - How fast does "something" in the system change
5. Criticality
   - How critical is the service to the users

Analysis

Budget

- If you have enough money, build a centralized system
- Look at Google if you doubt this claim ;-)
  - Any system can be made to scale with enough money
- P2P is therefore useful when budget is not unlimited
  - In other words, most real-world situations…
  - From the rest of this analysis, we assume limited budget

Resource relevance

- If shared resources are highly relevant to a large number of users, P2P makes sense
- Easier to build a distributed solution when interest is widely spread

Analysis, Continued

Trust

- If other users can be trusted, P2P is a good solution
  - For example, corporate network or any closed network
- Building a fully distributed, ***trusted*** network is still very much a research problem (and may remain so…)

Rate of system change

Btw: what does „trust“ mean in this context!?

- How high are the system dynamics?
  - Rate of peers joining and leaving, rate of information change in system, rate of change in network topology, …
- If the rate of change is too high, a distributed P2P solution might not be able to keep up
- Again, research problem

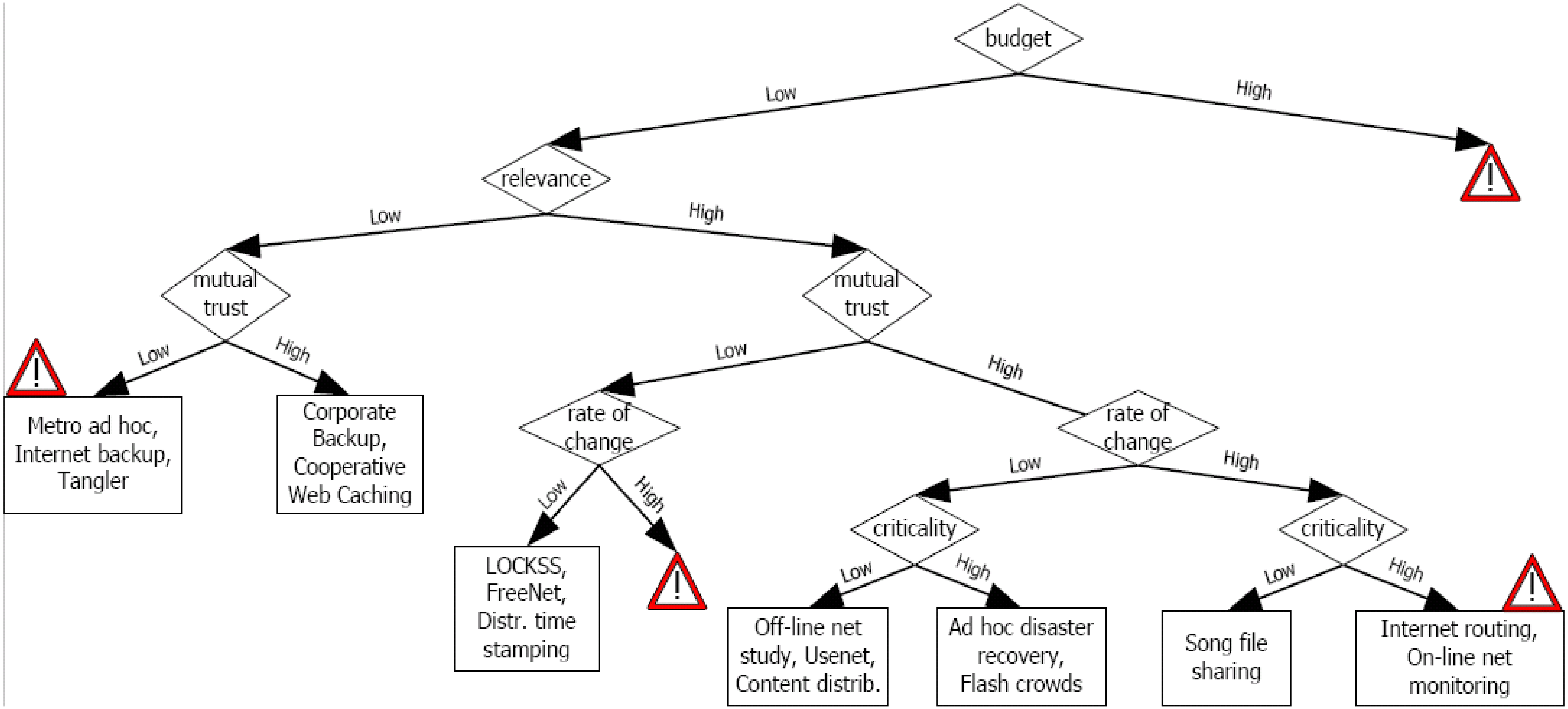
Analysis, End

Criticality

- How important is the service to the users?
- If you “can live without it”, P2P is acceptable
- If “it must work”, then consider other solutions…

- Summary: P2P is good when:
  - Budget is limited
  - Resources have wide interest and relevance
  - Trust between participants is high
  - Rate of change is manageable
  - Criticality is low
- Note: Again, no need to fulfill every point!

P2P Suitability Tree and Examples



budget
Low
High
relevance
Low
High
mutual trust
mutual trust
Low
High
Low
High
Metro ad hoc, Internet backup, Tangler
Corporate Backup, Cooperative Web Caching
rate of change
rate of change
Low
High
Low
High
LOCKSS, FreeNet, Distr. time stamping
criticality
criticality
Low
High
Low
High
Off-line net study, Usenet, Content distrib.
Ad hoc disaster recovery, Flash crowds
Song file sharing
Internet routing, On-line net monitoring

Taken from M. Roussopoulos et al. “2 P2P or not 2 P2P?”, IPTPS 2004

What does Future Hold for P2P?

- Take out crystal ball and look 5 years into future?
  - P2P has been around for just over 10 years now…
- Where will file sharing be in 5 years?
  - Still popular? Underground activity?
- P2P content distribution? (BitTorrent and others)
  - Microsoft building their system for software patches?
  - Some other systems patch via BitTorrent
- How about Skype and others?
  - Will Skype be around in 5 years?
  - Will Internet telephony be taken over by telcos?
- Research efforts in P2P?
  - More mature, concentrate on fundamental principles
  - What makes P2P different from other systems?

Future of P2P?

- Global P2P networks?
  - Besides file sharing, “Skype”, and research prototypes?
- Taking P2P concepts for other means and applications
  - Load balancing at S3 (inherently won)
  - Online Social Networking (remove central access to data)
  - Create resilient distributed systems (bot nets..)
- Insight on future trends: (at the example of Korea)
  - High bandwidth residential and wireless access
  - Online gaming (50% of network traffic!) main source of traffic
  - File sharing moved to pay models
  - Online communities gaining importance

Chapter Summary

- Peer-to-peer principle of self-organization and resource sharing
- Case Study of DNS to see it working the engineering way

- P2P systems exhibit specific characteristics:
  - Autonomy from central servers
  - Use of edge resources
  - Intermittent connectivity

- Hard to define clearly the limits of P2P
  - Quite some areas are closely related…
  - Different people working in different areas have different definitions

Outline of the Remainder of the Course

- Current P2P Systems
- Networks, Searching, and DHT
- Some Theory: Tools and Methods
- Novel Applications for P2P
  - Online Gaming
  - Online Social Networks
  - Application Level Multicast (P2P IPTV, Live Streaming)
  - P2P Botnets
- P2P and Security