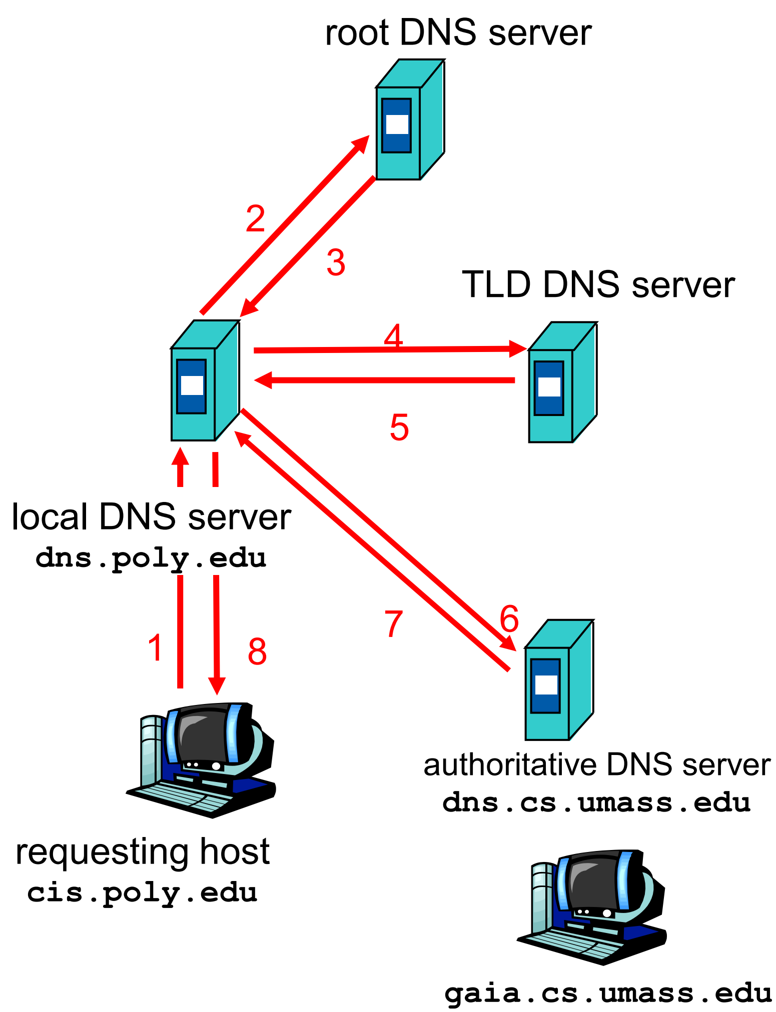
DNS – Example of Iterative Queries

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



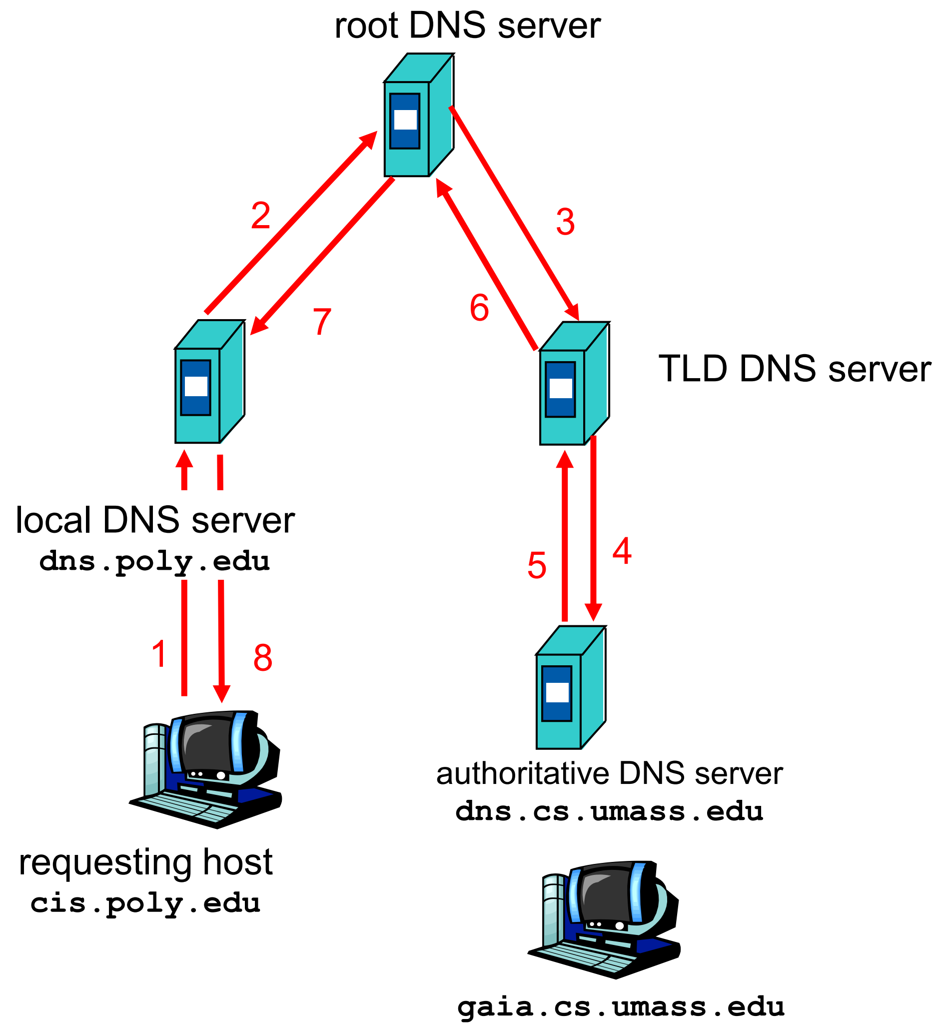
DNS – Recursive Queries

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:

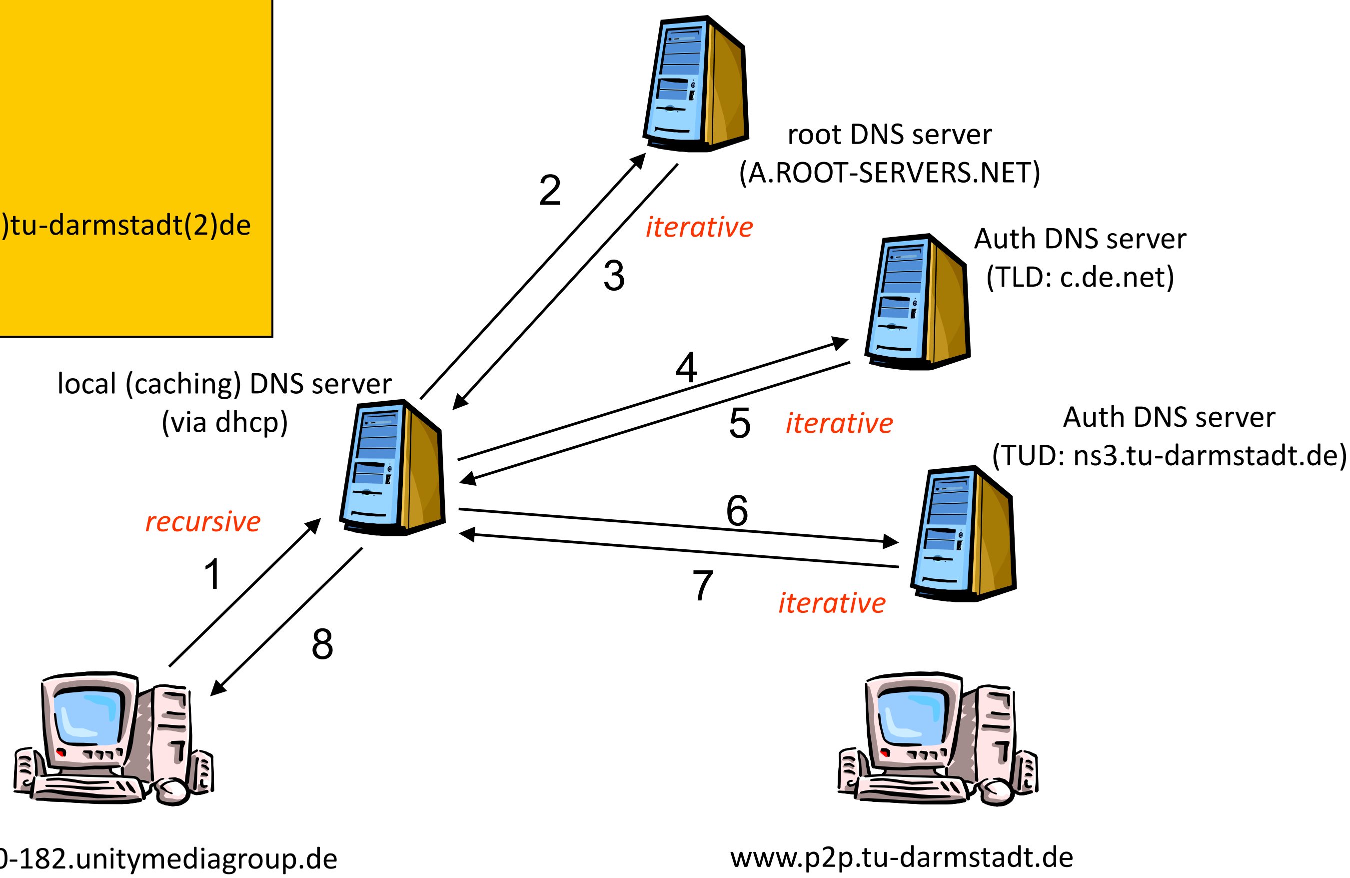
```
(networkutopia.com, dns1.networkutopia.com, NS)
(dns1.networkutopia.com, 212.212.212.1, A)
```

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

DNS – Recursive and Iterative Queries

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-darmstadt(2)de
- Type: 1 (A)
- Class: 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-darmstadt(2)de
-- 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

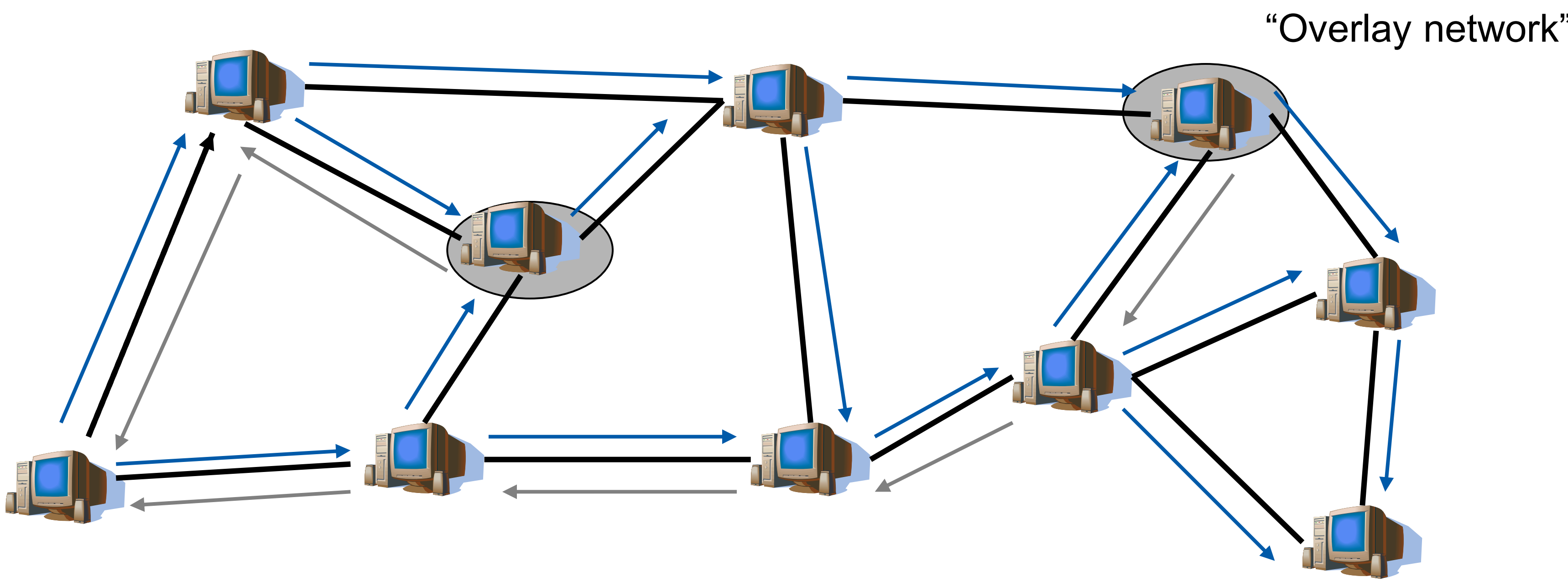
1. Structure name space (divide et impera)
2. Simple „routing“ b/c of structured (hierarchical) namespace

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

6. 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 in a Nutshell

- 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

- Main primary problems of P2P:
  - Staying connected
  - Resource lookup (name resolution, location of replica, especially selecting a good next hop for the delegation → routing)
  - Can't trust anyone

Peer-to-Peer (a Definition)

- **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

- We can introduce identifiers based on distributed algorithms (hashes)
- We can introduce structure using distributed algorithms (supernodes, etc.)
- *A P2P overlay on the Internet is a subset of links of a clique graph*

The P2P Environment

…all this in order to do:

File sharing, content distribution (BitTorrent/iptv), session initiation/chat/voip (skype, jabber), malware distribution/spam (botnets),…

- 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…)

- Closely related fields
  - Delay Tolerant Networks (Ad-hoc-, opportunistic-, pocket-switched-, vehicular-, <you-name-it> networks)
  - Wireless Sensor Networks
  - Epidemic-, Content-/ Context-based routing

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

New 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 date: ***OSN***;

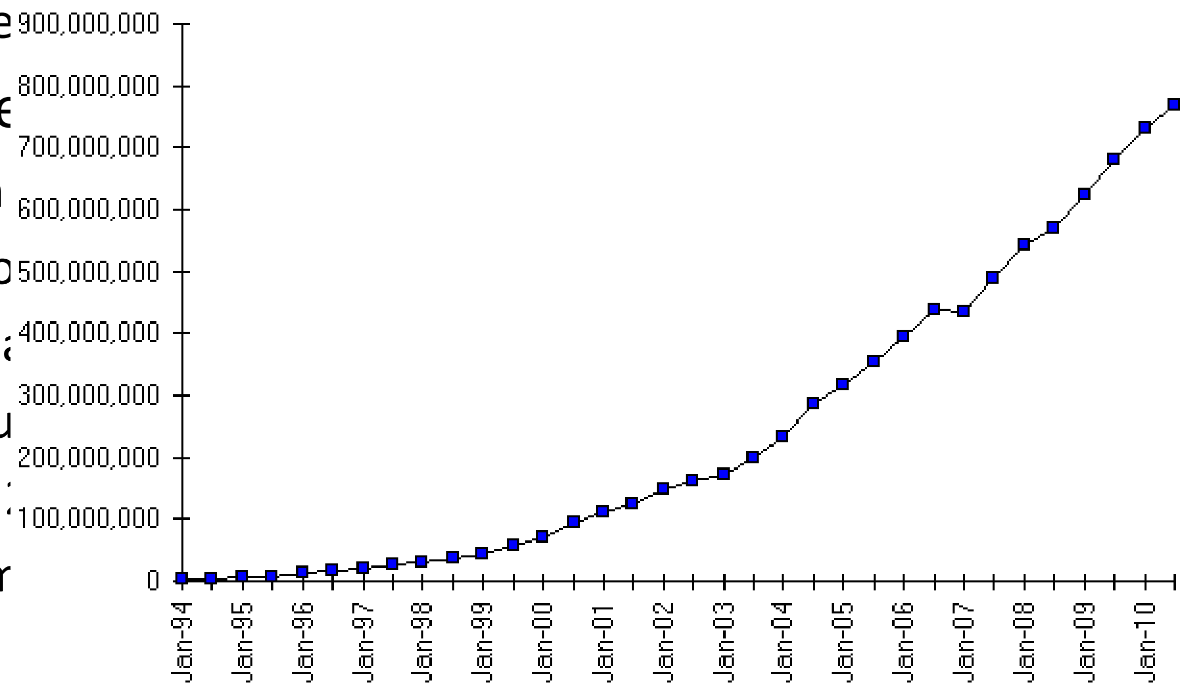
P2P: Some Statistics

- 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 providers (ISP) not too happy about this
  - But: Often traffic inte
- Some numbers: (take
  - KaZaA had 60 million
    - 85 million downlo
    - Software downloa
  - Google has 1 billion u
    (50mio/200mio in 2
  - Skype has over 300 m



(*) 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…)

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

Rate of system change

- 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

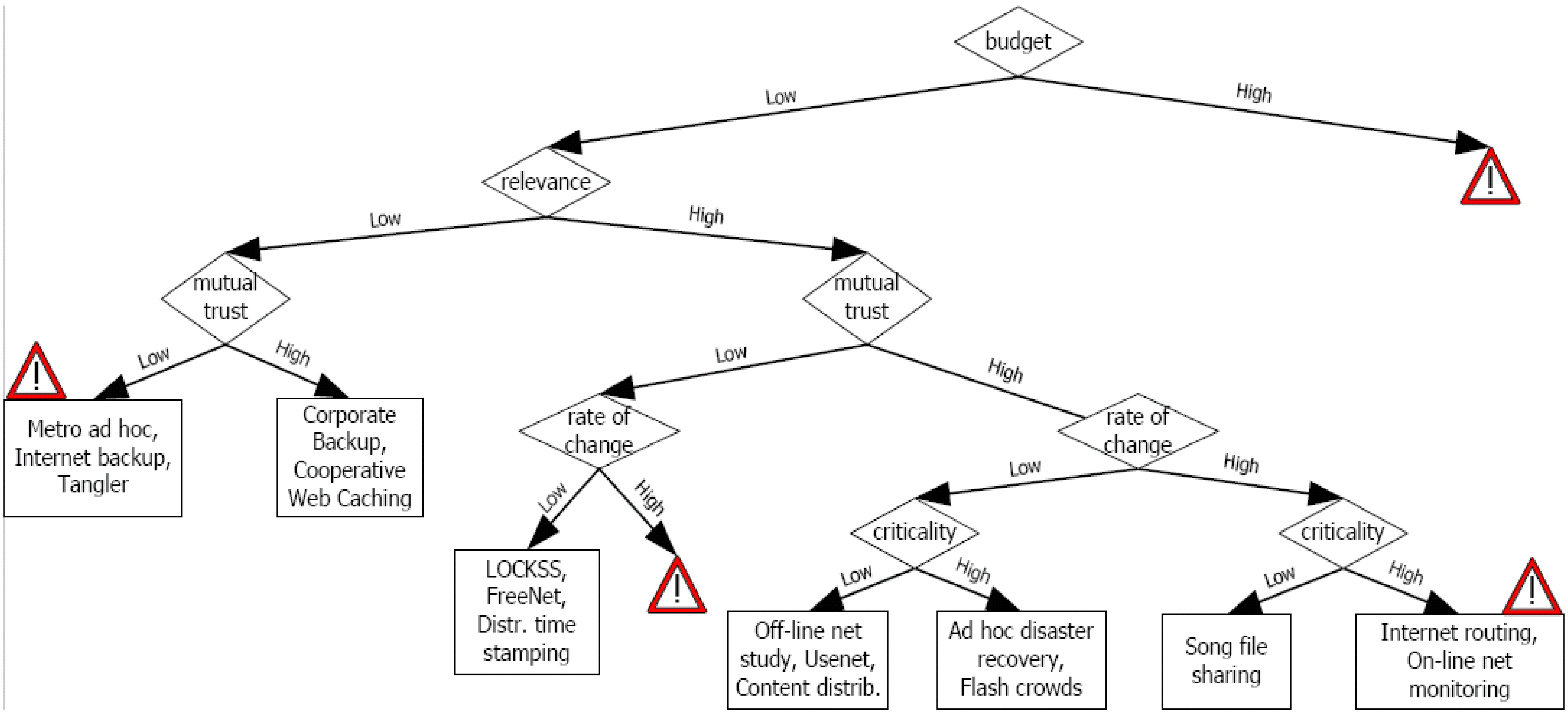
Analysis, End

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



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