

CSX55: DISTRIBUTED SYSTEMS [PEER-TO-PEER SYSTEMS]

Overlays

Traffic patterns divorced from
network topology

Breaking out of
hierarchical addressing shackles

With spaces that are flat
and more to go around

To route is to get closer
To what to you seek

Shrideep Pallickara
Computer Science
Colorado State University

COMPUTER SCIENCE DEPARTMENT



COLORADO STATE UNIVERSITY

1

Frequently asked questions from the previous class survey

- Pitfalls of P2P systems?
- Ongoing research in P2P systems?
- Degree of anonymity in P2P systems?



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.2

2

Topics covered in this lecture

- Napster
- 3rd generation P2P Systems



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.3

3

Term Project

- Will be released soon
 - ▣ You can choose your teammates: 2 for CS555, and 2-3 for CS455
 - ▣ Online/on-campus teams strongly encouraged
- Spatial data
- Tomorrow's lecture [On doing term projects well]
 - ▣ Ideation, etc.
- Term project pitches
 - ▣ Tuesday (March 5th) and Friday (March 8th)



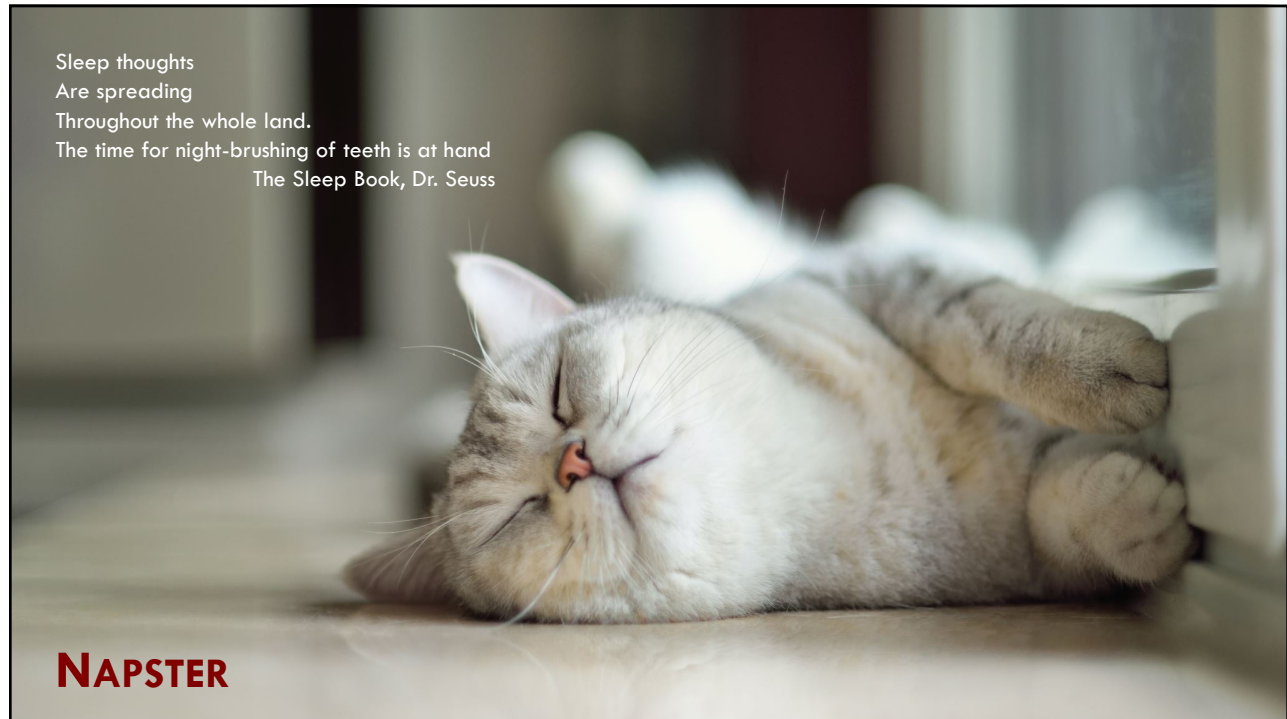
COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.4


4



5

Napster

- First application in which demand for massively scalable storage and retrieval arose
 - ▣ Downloading of digital music files
- Became very popular soon after its launch
- At its peak
 - ▣ **Several million** users
 - ▣ Thousands swapped music files *simultaneously*

 **COLORADO STATE UNIVERSITY** Professor: SHRIDEEP PALICKARA
COMPUTER SCIENCE DEPARTMENT PEER-TO-PEER SYSTEMS L13.6

6

Key features of the architecture

- Centralized indexes
- Users supplied the files
 - ▣ Stored and accessed on their personal computers
- Clients add their own music files to the pool of shared resources
 - ▣ Transmit a link to Napster's indexing service for each available file
 - ▣ Shared resources at the *"edge of the internet"*



COLORADO STATE UNIVERSITY

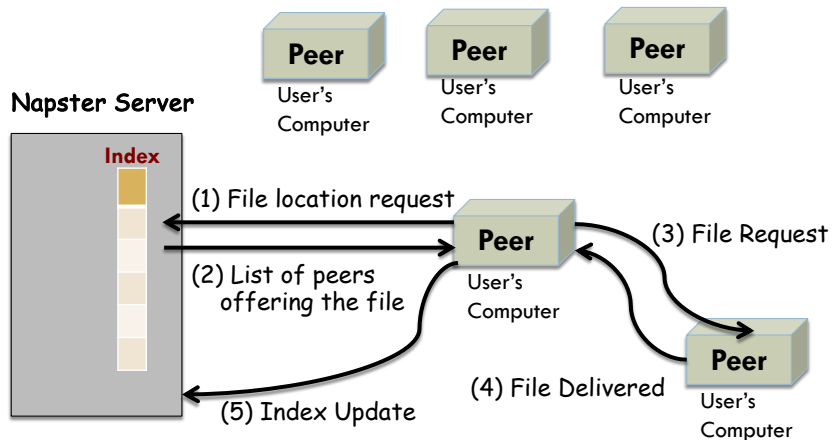
Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.7

7

Napster Architecture



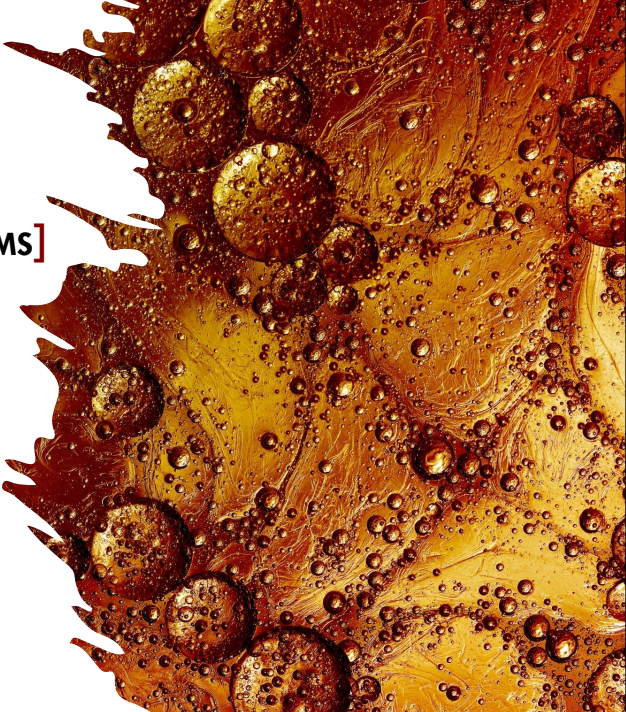
COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.8

8




OVERLAYS
[USED IN 3RD GENERATION P2P SYSTEMS]

COMPUTER SCIENCE DEPARTMENT

9

Overlays

- A distributed algorithm takes the responsibility of locating nodes and objects
 - ▣ This is the *routing overlay*
- Denotes that the **middleware** is a layer that is responsible for routing requests
 - ▣ From a client to the host that holds the requested object

 COLORADO STATE UNIVERSITY Professor: SHRIDEEP PALICKARA
COMPUTER SCIENCE DEPARTMENT PEER-TO-PEER SYSTEMS L13.10

10

But why call it an overlay?

- Denotes that it implements a *routing mechanism* in the **application layer**
 - This is different from routing mechanisms deployed at the network level, e.g., IP
- A logical hop in the routing overlay, encompasses multiple underlying router hops



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.11

11

What does the routing overlay do?

- Ensures any node can access any object
- Routes requests through a **sequence** of nodes
 - Exploits (local) knowledge at each of the intermediate nodes to locate the destination object
- If there are multiple replicas of objects?
 - Overlay maintains knowledge of all available replicas, and then delivers request to the *nearest* “live” node



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.12

12

OVERLAY VS IP ROUTING



COMPUTER SCIENCE DEPARTMENT

13

Overlay Routing vs. IP Routing

- There are several similarities between the two
- Why have a separate mechanism?
 - ▣ The legacy nature of IP
 - ▣ The legacy's impact is too strong for it to be overcome
 - Hard to support P2P applications directly



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.14

14

IP Routing vs. Overlay routing: Scale

- IP
 - IPv4 is limited to 2^{32} nodes
 - IPv6 is 2^{128}
 - But addresses are **hierarchically** structured
 - Much of the space is pre-allocated to meet administrative requirements
- Overlay
 - Globally Unique Identifiers (GUID) namespace is very large (2^{128} or 2^{160})
 - The namespace is also **flat** allowing for it to be much more fully occupied



IP Routing vs. Overlay routing: Load Balancing

- IP
 - Loads are determined by network topology and associated network patterns
- Overlays
 - Object locations can be randomized, so ...
 - Traffic patterns can be divorced from the network topology



IP Routing vs. Overlay routing: Network dynamics

- IP
 - ▣ Routing tables are *updated asynchronously* on a best-effort basis
 - ▣ Typically, on the order of an hour
- Overlays
 - ▣ Can be updated synchronously or asynchronously
 - ▣ Fractions of seconds



IP Routing vs. Overlay routing: Fault tolerance

- IP
 - ▣ Redundancy provided by network managers
 - To handle router or network connectivity failure
 - ▣ N-fold replication is costly
- Overlay
 - ▣ Routes and object references can be replicated n -fold
 - Tolerance of $(n - 1)$ failures of nodes or connections



IP Routing vs. Overlay routing: Target identification

- IP
 - ▣ Each IP address maps to exactly one node

- Overlay
 - ▣ Message can be routed to nearest replica of a target object



Main task of a routing overlay

- ① Routing of requests to objects
- ② Insertion of objects
- ③ Deletion of objects
- ④ Node additions and removals



Calculation of Globally Unique Identifiers (GUIDs)

- This is computed from all or part of the state of the object
- Function delivers a value that is, with a *very high probability*, unique
 - ▣ One way hash functions, such as SHA-1 or MD5 are often used



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.21

21

Why are overlay systems also called Distributed Hash Tables (DHTs)?

- Randomly distributed identifiers are used to determine resource
 - ▣ Placements
 - ▣ Retrievals



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.22

22

In the DHT model, a data item with GUID X

- Is stored at the node whose GUID is *numerically close* to X
- If the replication factor is r ?
 - Then it is stored at the r hosts whose GUIDs are next-closest to it numerically



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.23

23

A quick tour of how different P2P systems solve this

- Prefix routing
- Exploiting distance measures



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.24

24

Prefix routing

- Routes for delivery of messages based on values of GUIDs to which they are addressed
- Narrow search for the next node along the route by applying a **binary mask**
 - ▣ Selects an increasing number of hexadecimal digits from the destination GUID after each hop
- Used in Pastry and Tapestry



Exploiting different measures of distance to narrow search for next hop destination

- Chord
 - ▣ Numerical difference between GUIDs of the selected node and the destination node
- CAN
 - ▣ Uses distance in a d-dimensional hyperspace into which nodes are placed
- Kadmelia
 - ▣ Uses XOR of pairs of GUIDs as a metric for distance between nodes



A final note about GUIDs

- These are not human readable
- Client applications must obtain GUIDs for resources of interest through some indexing service
 - ▣ Human readable names or search requests
- For e.g., BitTorrent
 - ▣ Web index search leads to a sub file containing details of desired resource
 - GUID
 - URL of tracker: Host that holds up to date list of network providers willing to supply the file



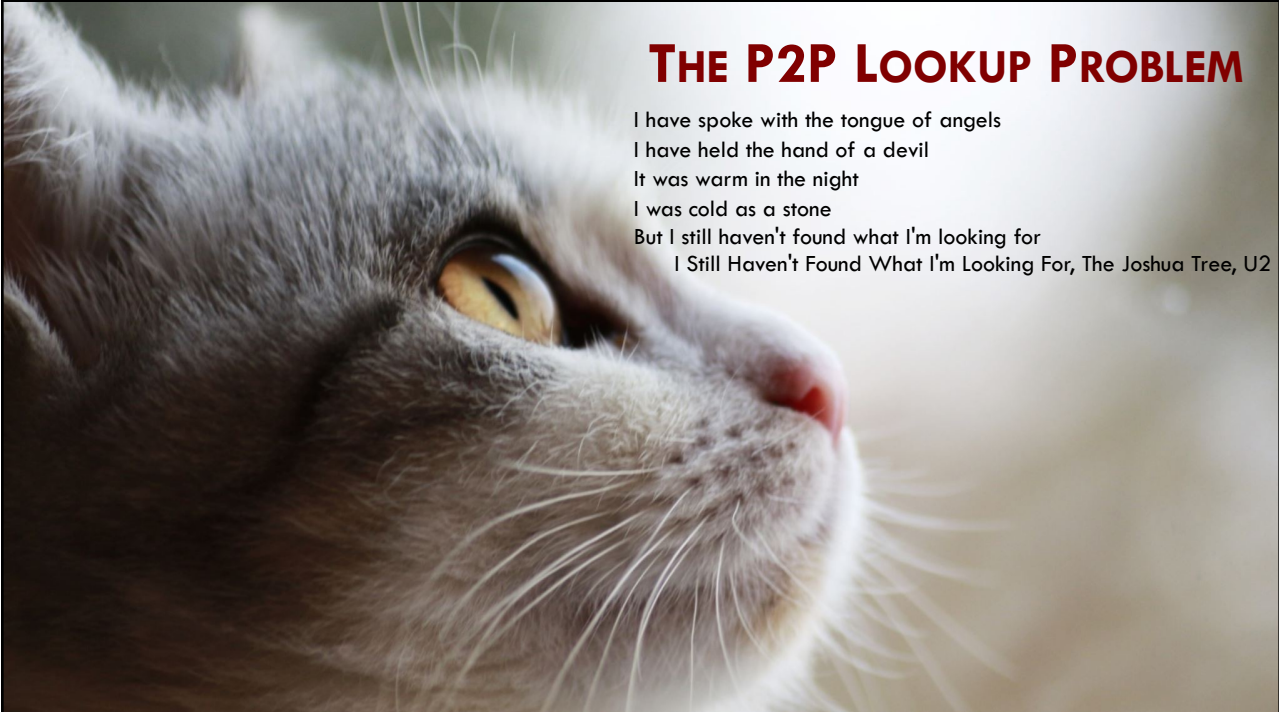
COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.27

27



THE P2P LOOKUP PROBLEM

I have spoke with the tongue of angels
I have held the hand of a devil
It was warm in the night
I was cold as a stone
But I still haven't found what I'm looking for
I Still Haven't Found What I'm Looking For, The Joshua Tree, U2

28

The peer-to-peer (P2P) lookup problem

- How do you **find** a data item in a large collection of peers?
- Lookup must be scalable and decentralized
 - Without hierarchy



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.29

29

The lookup problem: Centralized Approach

- Maintain central database
- Maintain table that maps file name to server that holds content
 - NAPSTER
- Problems
 - Reliability **Single point of failure**
 - Scalability **Database bottleneck for all requests**
 - Vulnerability **Targeted denial of service attacks**



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.30

30

The lookup problem: Broadcast

- **Flood** the network with requests looking for **X**
- When a node receives the request:
 - Check local repository
 - If it has **X**, node responds back with a message
- Scaling problems
 - ALL discovery requests sent to ALL nodes
 - ALL nodes process **every** discovery request



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.31

31

Broadcast costs can be reduced by organizing nodes into a hierarchy

- Searches start at the top
 - Traverse single path to the node that holds the desired data
- Directed traversal more frugal than broadcast
- Problems
 - Nodes at the **top** of the tree take **larger fraction** of load than leaf nodes
 - Requires expensive hardware
 - Loss of tree root (or node close to it) catastrophic



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.32

32

Distributed hash tables

- Few constraints on the structure of the keys
- REQUIREMENTS
 - Data identified using numeric **keys**
 - Nodes must be willing to store keys **for each other**



Storage and retrieval in distributed hash tables

- Data items are *inserted* and *found* by specifying a unique **key** for the data
- Underlying algorithm must determine *which node* is responsible for storing the data



Distributed Storage using DHTs: Publishing a file

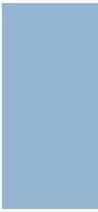
- **Convert** file-name to numeric key
 - Using one-way hash functions like MD5 or SHA-1
- Call **lookup (key)**
 - Returns IP address of node responsible for key
- **Send file** to be stored at node returned by lookup



Distributed Storage using DHTs: Retrieving a file


- ① Obtain name of file
- ② Convert it to a key using one-way hash function
- ③ Call lookup (key)
- ④ Ask resulting node, from (3), for a copy of the file





IMPLEMENTING DHTs

COMPUTER SCIENCE DEPARTMENT




COLORADO STATE UNIVERSITY

37

Implementing DHTs: 3 core elements

- **Mapping** keys to nodes
- **Forwarding** a lookup for a key to the appropriate node
- Building **routing tables**



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.38

38

Implementing DHTs: Mapping keys to nodes

- Must be load balanced
- Done using one-way hash functions
 - ▣ MD5 (128-bit) or SHA-1 (160-bit)
- Ensures that content is distributed **uniformly**



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.39

39

Implementing DHTs Forwarding lookups

- Any node that receives query for key
 - ▣ Must forward it to a node whose ID is **closer** to the key
- Above rule guarantees that query **eventually arrives** at the closest node
- For e.g.:
 - ▣ Node has ID 346, and key has ID 542
 - ▣ Forwarding to node 495 gets it numerically closer



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.40

40

Implementing DHTs: Building routing tables

- Multiple nodes participate in locating content
- Each node must know about **some other** nodes
 - To forward lookup requests
 - SUCCESSOR
 - The node with the **closest succeeding** ID
 - Other nodes
 - For efficiency in routing



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.41

41

Distributed hash tables: Identifiers

- Data items are assigned an identifier from a large random space
 - 128-bit UUIDs or 160-bit SHA1 digests
- **Nodes are also assigned a number from the same identifier space**



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.42

42

Crux of the DHT problem

- Implement an efficient, **deterministic** scheme to
 - Map data items to node
- When you **look up** a data item
 - Network address of node holding the data is returned



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.43

43

The contents of this slide-set are based on the following references

- *Distributed Systems: Principles and Paradigms*. Andrew S. Tanenbaum and Maarten Van der Steen. 2nd Edition. Prentice Hall. ISBN: 0132392275/978-0132392273. [Chapter 5]
- *Distributed Systems: Concepts and Design*. George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair. 5th Edition. Addison Wesley. ISBN: 978-0132143011. [Chapter 10]



COLORADO STATE UNIVERSITY

Professor: SHRIDEEP PALICKARA
COMPUTER SCIENCE DEPARTMENT

PEER-TO-PEER SYSTEMS

L13.44

44