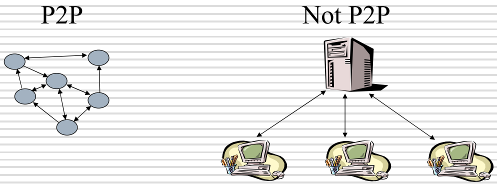
2 P2P or Not 2 P2P



Presented by:

Ghada Alnifie

2 P2P or Not 2 P2P

- What questions should a system designer ask to judge weather a P2P solution is appropriate for his particular problem?

** a heuristic decision tree*

P2P Environments

- Self-organizing

  nodes organize them selves into a network through a discovery process

- Symmetric Communication

  peers are considered equals; they both request and offer services

- Decentralized Control

  there is no central controller that dictates behavior to individual nodes

Problem Characteristics Axes

Decisions to be considered:

- Budget
  - Ample or limited?
- Resource Relevance to Participants
  - Likelihood that a “unit of service” is interesting to many peers
- Trust
  - The cost of handling mutually distrusting peers is high
- Rate of System Change
  - Timeliness and consistency
- Criticality
  - Solving critical problems may need centralized solution

Candidate Problems

□ Routing Problems

takes on p2p characteristics when the scale is large enough or when centralization is ruled out

- Internet Routing
- Ad hoc in Disaster Recovery
- Metropolitan-area Cell Phone Forwarding

Candidate Problems

□ Backup

the process in which a user replicates his files in different media at different locations to increase data availability

- Internet Backup
- Corporate Backup

Candidate Problems

□ Distributed Monitoring

Monitoring in large distributed systems

- simple (publish/subscribe)
- complicated online manipulation (SQL queries)
- the basis for an off-line study

Candidate Problems

□ Data Sharing

peers offer the data to be shared and also search collection to find their interest

- File sharing
- Censorship Resistance

Candidate Problems

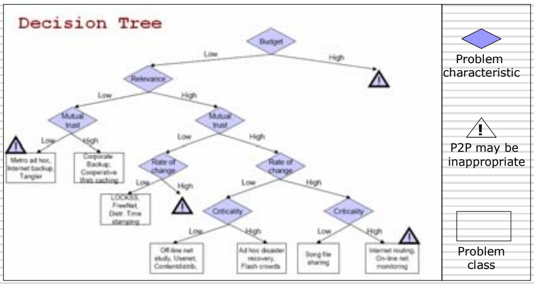
- □ Data dissemination

  same as data sharing, with the difference that data is stored to be spread (forwarded)

  - Usenet

  - Non-critical Content Distribution

2 P2P or Not 2 P2P


Decision Tree
Budget
Low
High
Relevance
Low
High
Mutual trust
Low
High
Metro ad hoc, Internet backup, Tangler
Corporate Backup, Cooperative Web caching
Mutual trust
Low
High
Rate of change
Low
High
LOCKSS, FreeNet, Distr. Time stamping
Rate of change
Low
High
Criticality
Low
High
Off-line net study, Usenet, Contentdistrib.
Ad hoc disaster recovery, Flash crowds
Criticality
Low
High
Song file sharing
Internet routing, On-line net monitoring
Problem characteristic
P2P may be inappropriate
Problem class

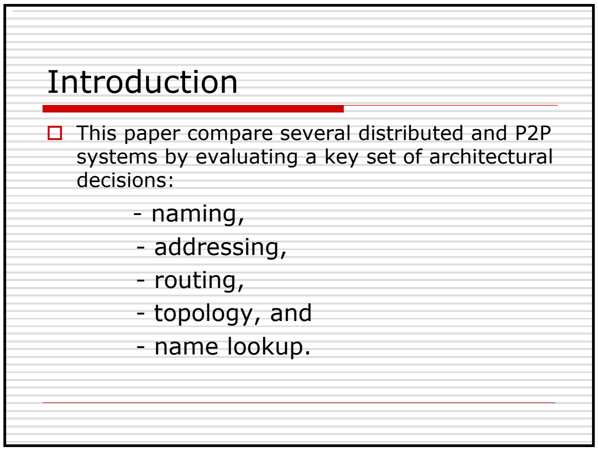
Conclusions

- □ The limited budget requirement is the most important motivator, relevance comes second since can be compensated for. Problems that lack these 2 requirements are not appropriate for a P2P solution.

- □ Trust between nodes greatly eases P2P deployment.

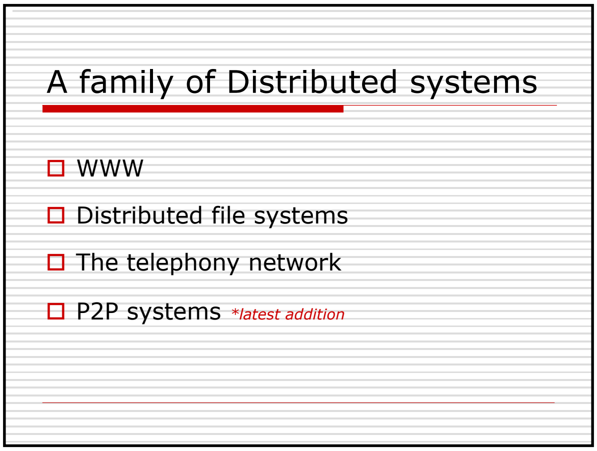
Second Paper

Exploring the Design Space of Distributed and Peer-to-Peer Systems:
Comparing the Web, TRIAD, and Chord/CFS



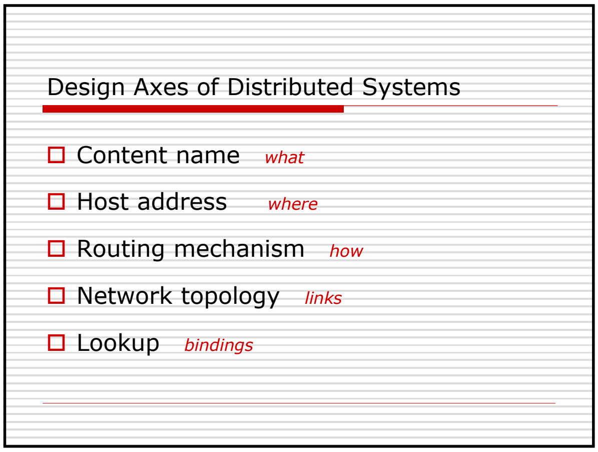
Introduction

- This paper compare several distributed and P2P systems by evaluating a key set of architectural decisions:
  - naming,
  - addressing,
  - routing,
  - topology, and
  - name lookup.



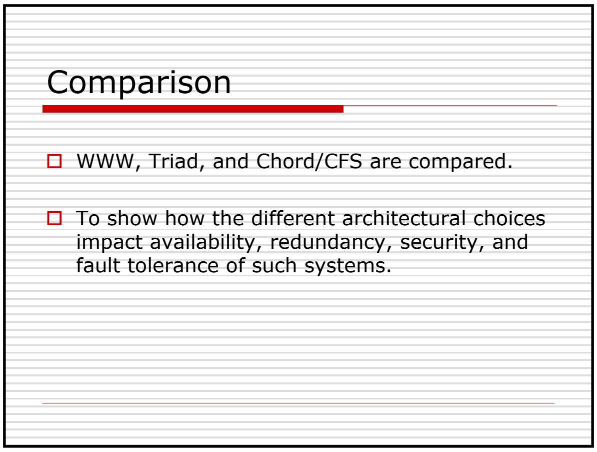
A family of Distributed systems

- WWW
- Distributed file systems
- The telephony network
- P2P systems **latest addition*



Design Axes of Distributed Systems

- Content name *what*
- Host address *where*
- Routing mechanism *how*
- Network topology *links*
- Lookup *bindings*



Comparison

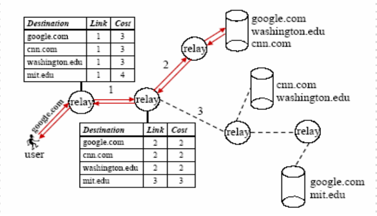
- WWW, Triad, and Chord/CFS are compared.
- To show how the different architectural choices impact availability, redundancy, security, and fault tolerance of such systems.

The World Wide Web

- Perhaps the most ubiquitous, popular, and successful distributed system.
- Enables clients to retrieve hyperlinked content.
- Names: infinite space.
- Addresses: globally unique IP addresses
- Routing: a combination of Internet routing protocols
- Topology: hierarchical, consisting of interconnected autonomous systems and sub-networks within them
- Lookup: URLs are resolved to IP addresses through the domain name server (DNS)

TRIAD

- Defines a content layer that replaces the Web's address-based routing with a *name-based* routing protocol.



Destination	Link	Cost
google.com	1	3
cnn.com	1	3
washington.edu	1	3
mit.edu	1	4

Destination	Link	Cost
google.com	2	2
cnn.com	2	2
washington.edu	2	2
mit.edu	3	3

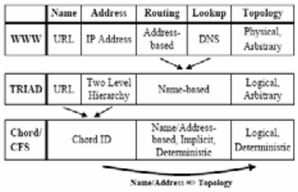
TRIAD

- Names: uses the Web's URL namespace for content naming
- Addresses: a composition of two namespaces: globally unique IP addresses of AS, and locally unique IP addresses within each AS.
- Routing: uses name-based, BGP-like routing protocol
- Topology: can be arbitrary consisting of logical links between relay nodes
- Lookup: unifies lookup and routing

Chord/CFS

- Hosts serve as servers, clients, and intermediate routers.
- Names: a Chord identifier is obtained by hashing
- Addresses: Addresses are obtained by concatenating a host's IP address with a small virtual host number, and hashing the result into a 160 bit address.
- Routing: can be thought of address-based or name-based.
- Topology: a deterministic function of participating peers' addresses
- Lookup: unifies lookup and routing

Summary



	Name	Address	Routing	Lookup	Topology
WWW	URL	IP Address	Address-based	DNS	Physical, Arbitrary
TRIAD	URL	Two Level Hierarchy	Name-based		Logical, Arbitrary
Chord/ CFS	Chord ID		Name/Address-based, Implicit, Deterministic		Logical, Deterministic

Figure 2. Routing and lookup are unified in TRIAD and Chord. Chord's name and address spaces are identical, and its topology is a deterministic function of names/addresses.

How do these different architectural choices impact the performance of a distributed system?

Names and Addresses

1) WWW:

- □ **Hierarchical DNS**

  a malicious web server cannot:
  - register web names randomly
  - cannot attack another web server's content by duplicating its URL

- □ **Hierarchical IP**

  makes it difficult for a malicious host to hijack an IP address outside of its allocated range

*** although both hierarchal, name space and address space are completely independent in the web*

Names and Addresses

2) TRIAD:

- □ Content names are modeled after URLs, this is important for scalability.
- □ The ability to create a name in TRIAD is unrestricted. Restrictions on binding rights must be enforced by the routing infrastructure; to date, this issue remains unsolved.
- □ Similar to web, individual hosts cannot affect IP assignments.
- □ <u>Lookup and routing are unified</u>

Names and Addresses

3) Chord/CFS:

- □ Content name space is flat:
  - unless the right to insert a name-to-address binding is controlled, any host can cause unbound amounts of effort and storage to be expended across the system.
  - attacks on a specific victim are possible
- □ The set of content names associated with an address is deterministic (possibility for attacks)
- □ <u>Lookup and routing are unified</u>

Routing, Lookup, and Topology

1) WWW:

- □ Routing policy is selected independent of both physical topology and content
- □ Possible to engineer redundancy (higher availability) at two levels in the web
- □ Endpoints of a web transfer (servers and clients) are physically distinct from routers
- □ A web server failure does not affect the routability of IP addresses, and a router failure doesn't affect content availability

Routing, Lookup, and Topology

2) TRIAD

- □ Routing policy cannot be selected independently of content
- □ Two levels of redundancy is supported
- □ Possible to construct a topology in which content servers are never intermediate nodes in a route
- □ The failure of a link doesn't cause contents to be unavailable

Routing, Lookup, and Topology

3) Chord/CFS

- □ Topology is a deterministic function of the set of participating addresses:
  - routing tables need not be advertised
- □ Redundancy occur at multiple levels
- □ Content name and address namespaces are unified: the content name is the address towards which a peer routes requests
- □ All peers serve as both routers and content distributors
- □ The deterministic nature of routes lead to many problems

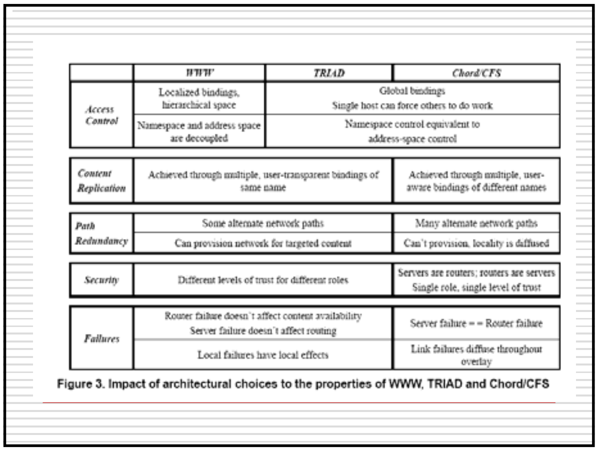
Conclusion

- □ Three fundamental design differences:

1) In Chord/CFS, the content and address namespaces are equivalent

2) Chord's network topology is a deterministic function of its content and address namespace

3) In both TRIAD and Chord, lookup and routing are unified.



	WWW	*TRIAD*	*Chord/CFS*
Access Control	Localized bindings, hierarchical space	Global bindings Single host can force others to do work	
	Namespace and address space are decoupled	Namespace control equivalent to address-space control	
Content Replication	Achieved through multiple, user-transparent bindings of same name		Achieved through multiple, user-aware bindings of different names
Path Redundancy	Some alternate network paths		Many alternate network paths
	Can provision network for targeted content		Can't provision, locality is diffused
Security	Different levels of trust for different roles		Servers are routers; routers are servers Single role, single level of trust
Failures	Router failure doesn't affect content availability Server failure doesn't affect routing		Server failure = = Router failure
	Local failures have local effects		Link failures diffuse throughout overlay

Figure 3. Impact of architectural choices to the properties of WWW, TRIAD and Chord/CFS