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

Stefan Saroiu, P. Krishna Gummadi, Steven D. Gribble
University of Washington

Abstract: *Despite the existence of many peer-to-peer systems, some of their design choices and implications are not well understood. This paper compares several distributed and peer-to-peer systems by evaluating a key set of architectural decisions: naming, addressing, routing, topology, and name lookup. Using the WWW, Triad, and Chord/CFS as examples, we illustrate how different architectural choices impact availability, redundancy, security, and fault-tolerance.*

1 Introduction

Peer-to-peer systems are the latest addition to a family of distributed systems whose goal is to share resources across their participants. Previous members of this family include the WWW, distributed file systems, and even the telephony network. To compare these systems, one can decompose them along the following design axes, which are an extension of those proposed by Shoch [9] and Saltzer [7]:

Content name: A name describes *what* a user is looking for, such as a file name in a file system.

Host address: An address describes *where* a resource is, for example, an IP address describes where a host resides in the Internet.

Routing mechanism: A route describes *how* to get to a destination. A routing mechanism (such as BGP across Internet autonomous systems, or ASs) is used to discover or disseminate routes.

Network topology: Topology describes the set of physical or logical *links* between hosts.

Lookup: *Bindings* between names and addresses are registered in the system. Participants use a lookup mechanism that *resolves* a name into an address, based on the registered bindings.

These design axes represent one possible framework to reason about the architecture of a system. Although this framework is clear in the abstract, in practice real systems often blur the distinction between some of these axes. For example, NAT blurs routing and lookup by introducing a name translation mechanism so that non-routable IP addresses can be "bridged" to the routable Internet. Additionally, IP addresses are converted into MAC Ethernet addresses in a manner similar to name translation. However, we believe that our decomposition is useful both when designing and analyzing a system, and that, by mapping design choices along these axes, we can learn about the trade-offs made by each system.

In this paper, we compare the designs of three different distributed architectures: the WWW, TRIAD [5], and Chord/CFS [2] (as a representative of recently proposed peer-to-peer architectures [3, 6, 10]). We then derive several performance, security, and robustness implications that result from their design choices.

1.1 The World Wide Web

The WWW is perhaps the most ubiquitous, popular, and successful distributed system. The WWW enables clients to retrieve hyperlinked content.

Names: Web content names are drawn from an infinite space of globally unique Uniform Resource Locators (URLs), which are structured as a fully qualified domain name (FQDN) combined with a locally unique relative URL [1, 4]. The right to bind an FQDN to an IP address is controlled by hierarchical delegation, and the right to bind relative URLs is controlled by local policy.

Addresses: WWW addresses are globally unique, hierarchically organized IP addresses of Internet hosts (servers, clients, caches, or intermediate routers). There is a finite but large number of IP addresses; addresses are allocated in ranges from a centralized authority, and address assignment rights are delegated locally within these ranges.

Routing: Routing in the WWW is a combination of Internet routing protocols, including BGP, IS-IS, and OSPF. Routing decisions are driven by business policy and performance. The ability to route to an IP address is the result of advertising that address on a routing protocol. There is little control over the right to advertise, as there is typically a lack of authentication and access control in routing protocols.

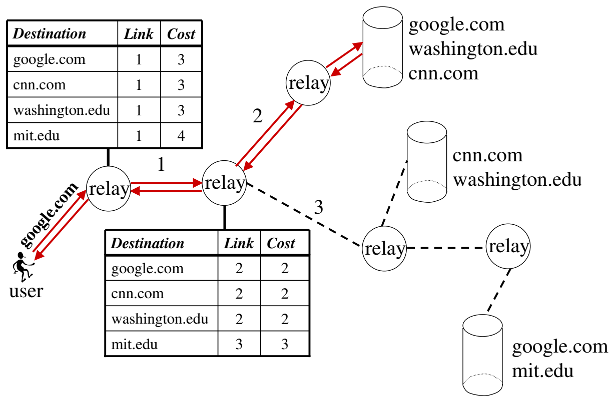

Figure 1. The TRIAD architecture. A name request (google.com) is forwarded by intermediate relays toward the "best" content replica.

Topology: The WWW topology is based on the physical topology of Internet hosts. This topology is roughly hierarchical, consisting of interconnected autonomous systems and subnetworks within them.

Lookup: FQDNs within URLs are resolved to IP addresses through the domain name system (DNS); relative URLs are resolved and bound locally by servers. DNS itself is another distributed system; however, the WWW could replace the DNS lookup mechanism with no semantic loss, as is proposed by TRIAD.

1.2 TRIAD

TRIAD defines a content layer that replaces the Web's address-based routing with a *name-based* routing protocol. An individual piece of content is advertised by each server replica, so that lookup requests are directed from clients along intermediate routers (relay nodes) to servers, and back along the same path. Each relay node maintains a set of name-to-next-hop mappings, just as an IP router maps address prefixes to next hops. When a request for a content name arrives, a relay looks up the name and forwards the request toward the "best" server replica. Once the request reaches a relay responsible for a server replica, that relay sends back a response containing the server's address (Figure 1).

Names: Similar to the Web, TRIAD resources are objects spread across servers. Although TRIAD's content namespace does not require a specific structure, content names are routing table entries, and therefore need to aggregate. Because URLs are hierarchical, TRIAD suggests using the Web's URL namespace for content naming.

Addresses: TRIAD's addresses are a composition of two namespaces: globally unique IP addresses of AS, and locally unique IP addresses within each AS. This results in a finite but very large address space. While the inter-AS address space is controlled by a centralized authority, each intra-AS address space is managed locally by its AS.

Routing: TRIAD uses a name-based, BGP-like routing protocol called NBRP, which distributes name suffix reachability messages. The ability to route to a name is the result of advertising that name across the NBRP protocol.

Topology: TRIAD's topology can be arbitrary, consisting of logical links between relay nodes over which NBRP messages flow. For performance reasons, it is suggested that TRIAD's topology should reflect the physical Internet topology.

Lookup: TRIAD unifies lookup and routing: resolving a name into an address is achieved by routing the name to its destination. Once the destination address is found, the lookup reply is routed back to the source on the same path.

1.3 Chord/CFS

All hosts in Chord/CFS-style peer-to-peer systems serve three roles; they act as servers, clients, and intermediate routers. This symmetry of roles has several design implications.

Names: In Chord, each piece of content is named with a Chord identifier, obtained by hashing the content into 160 bits. The content namespace is flat, large, and uniformly populated.

Addresses: Chord's address namespace is structurally identical to the content namespace. 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. Because addresses have 160 bits, they are probabilistically globally unique in the system.

Routing: Since the content and address namespaces are equivalent, routing can be thought of both as address-based, like the Web, or name-based, like TRIAD. Unlike the Web or TRIAD, any participating Chord host acts as an intermediate router. Routing in Chord is simple: each host directs queries to the neighbor whose address is closest to the name according to a pre-determined lexicographic order.

Topology: Chord's overlay network topology is a deterministic function of participating peers' addresses. Each peer has a successor and predecessor based on a total ordering of addresses, and each

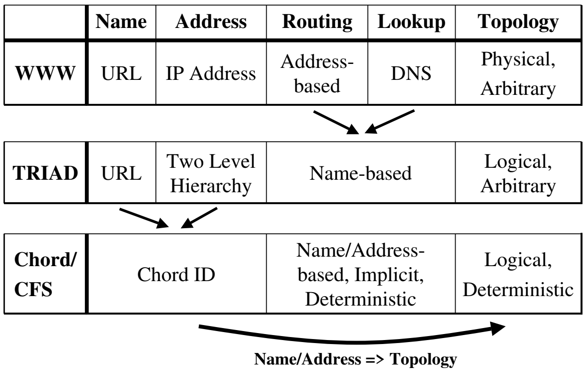

Figure 2. Routing and lookup are unified in TRIAD and Chord. Chord's name and address spaces are identical, and its topology is a deterministic function of names/addresses.

peer maintains logarithmically-sized "finger table" of connections to other peers. Although stale finger table entries are tolerable, they act as shortcut routes and their freshness ensures efficient routing.

Lookup: Like TRIAD, Chord unifies lookup and routing. Since the content and address namespaces are equivalent, the identity function is sufficient to bind names to addresses: a name is bound to the address which *is* the content name. Name resolution is done by routing the name through the network.

1.4 Summary

A summary of the systems' designs is given in Figure 2. This decomposition has served to illustrate important differences between these systems. For instance, in Chord, binding of names to addresses is done by the identity function and the topology is a deterministic function of host addresses. In both Chord and TRIAD, lookup is unified with routing. We now discuss the implications of these design differences.

2 Names and Addresses

In this section, we explore how content names and host addresses are created and bound to each other in our three systems.

2.1 The World Wide Web

Binding names to IP addresses is controlled by the use of DNS as a lookup mechanism. The hierarchical nature of DNS imposes structure on content names, but it also serves to delegate binding rights to authorities that own subtrees of the namespace. While any Web server can create an infinite number of URLs, binding them to IP addresses is restricted by the ability to register FQDNs within a particular DNS subtree. As a result, a malicious Web server cannot pollute the global namespace by registering a large number of dummy or otherwise harmful names, nor can it attack another Web server's content by duplicating its URLs.

There is a similar hierarchical delegation of IP address assignment rights within the Web. An individual may be able to control the assignment of a large but finite number of IP addresses; for instance, MIT owns a class A subnet, allowing it to assign 2^{24} addresses, but all within a fixed range. However, it is difficult for a malicious host to hijack an IP address outside of its allocated range, as this involves injecting false routing advertisements into the network. Although they both are hierarchical, the name and address namespaces in the Web are completely independent: control over one does not grant control over the other.

2.2 TRIAD

In TRIAD, lookup and routing are unified. As a result, the binding of a name to an address is accomplished by advertising a route across the TRIAD network, and lookup is performed by routing a name to its destination. Similar to the Web, 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 unresolved.

Addresses in TRIAD are the composition of globally routable IP addresses assigned to ASs, and locally routable IP addresses within ASs. The authority to assign routable addresses is therefore split across two levels. Similar to the Web, a centralized naming authority delegates globally visible IP ranges to ASs, and ASs enforce local policies for address assignment. Therefore, individual hosts in TRIAD typically cannot affect globally visible IP assignment.

Because name routing in TRIAD involves routing advertisements and routing table formation, the ability to aggregate names is important for scalability. As a result, the content namespace must be hierarchical (or otherwise aggregatable) in practice. For this reason, TRIAD content names are modeled after URLs.

2.3 Chord/CFS-style Peer-to-Peer Systems

Since the peer-to-peer content namespace is flat, the responsibility for managing content is randomly distributed across the address namespace. The insertion of a name-to-address binding (i.e. publishing content) into the system causes some host to accept the responsibility and incur the cost of managing that content. Thus, unless the right to insert a name-to-address binding is controlled, any host can cause unbounded amounts of effort and storage to be expended across the system. Furthermore, attacks on specific victims are also possible. For example, an attacker could overwhelm a targeted victim address with content, or even cause the targeted host to store undesirable or illicit content. In contrast, in the Web and TRIAD, binding a name to an address does not cause the host to store the content, making such attacks impossible.

The set of content names associated with an address is also deterministic. If hosts are allowed to select their own addresses, they can use this deterministic mapping to control access to specific content names. In Chord/CFS, the ability to create an address is restricted by limiting hosts to using hashes of their IP addresses concatenated with a small "virtual host" number. An attacker who has assignment rights over *O(number of Chord nodes / max virtual hosts)* IP addresses can control arbitrary content in the system.

3 Routing, Lookup and Topology

In this section, we discuss the consequences of unifying routing and lookup in TRIAD and Chord/CFS, in contrast to the Web. Furthermore, because Chord's topology is a function of its address space, several unexpected implications emerge affecting the system's redundancy, availability, fault-tolerance and security.

3.1 The World Wide Web

The structure of the WWW is mapped directly onto the Internet's physical topology: Web servers and clients are addressed by their IP addresses, and the routing of data between them is performed using IP routing protocols such as BGP, IS-IS, and OSPF. Infrastructure such as content-delivery networks and caching hierarchies extend the name-to-address lookup mechanism, but the result of a lookup is still an IP address of the host that will serve the data.

In the WWW, routing policy can be selected independent of both physical topology and content. This flexibility allows policy to be driven by efficiency, business rules, or even local physical characteristics. Routing policy can be altered without affecting naming, binding, or content placement. It is common for ISP operators to adjust policy to achieve a financial or traffic balancing goal; however these adjustments are functionally transparent to the rest of the system.

It is possible to engineer redundancy (and hence higher availability) at two levels in the Web. At the routing level, redundant physical routes provide alternate paths for data transport between clients and servers. At the name binding level, binding the same name to multiple addresses allows clients or middleware to fail over to an alternate address if one destination becomes unavailable.

The endpoints of a Web transfer (servers and clients) are, in general, physically distinct from routers. This physical separation of roles has several benefits. Different degrees of trust can be associated with different roles; for example, core Internet routers are more protected and trustworthy than Web clients or servers. Hosts can be provisioned and optimized for their specific roles; a high-speed router needs different hardware, OS, and software support than a Web server or client. Finally, side-effects of host failures are isolated with respect to the role they play. A Web server failure does not affect the routability of IP addresses, and a router failure doesn't affect content availability (unless the failure partitions the network).

3.2 TRIAD

In TRIAD, because routing *is* the content name-to-address lookup mechanism, routing policy can no longer be selected independently of content. If TRIAD's network topology mirrors the physical topology of the Internet, as suggested by the authors, then an efficient routing policy is enough to enable clients to route requests to their topologically "nearest" content replica. This only works because TRIAD can route on arbitrary topologies, unlike Chord/CFS-style peer-to-peer systems, as discussed below.

TRIAD also supports two levels of redundancy. Multiple name-to-address bindings are attainable by replicating content on additional routable destinations; if one destination fails, the content is available at the replicas' addresses. This replication technique is possible because TRIAD routing uses content names rather than host addresses. In addition, from a given source, there may be multiple routes to a destination name. Thus, the failure of a link in TRIAD does not necessarily cause content to become unavailable.

Because TRIAD supports routing over arbitrary topologies, it is possible to construct a topology in which content servers are never intermediate nodes in a route, and therefore servers do not need to participate in the routing of requests. Thus, content hosting and routing are still separable roles, enabling the same separations of trust, provisioning, and failure as the Web.

3.3 Chord/CFS-style Peer-to-Peer Systems

The topology of Chord/CFS-style peer-to-peer systems is a deterministic function of the set of participating addresses. As a side-effect, routing tables need not be advertised across the system, eliminating one cause of overhead. Routing tables, approximated by finger tables in Chord, are constructed by each peer upon its entry to the system, and lazily updated to reflect the departure of its neighbors. If finger tables are kept up-to-date, the carefully chosen topology bounds lookup route lengths by $\log(\text{\# peers})$.

The content name and address namespaces in Chord/CFS are unified, which allows binding to be the identity function: the content name *is* the address towards which a peer routes requests. When combined with Chord's deterministic topology, this implies that all peers are expected to serve both as routers and content destinations. These roles are inseparable: a peer cannot choose an address that will relieve it of routing responsibilities, and the topology cannot be engineered to relieve content destinations of routing responsibilities. However, roles no longer need to be explicitly assigned, and the topology need not be explicitly constructed; they are determined as peers join and leave the system, vastly simplifying and decentralizing the administration of the system.

Redundancy in Chord/CFS can occur at multiple levels. Because binding is the identity function, it is impossible to bind the same content name to multiple addresses. However, a naming convention can assign aliases to any given content name; unlike TRIAD or the WWW, redundancy at this level is not transparent to the user, since it is exposed in the content namespace. A second level of redundancy exists within the overlay itself. There are on the order of $\log(\text{\# peers})$ mutually disjoint routes between any two given addresses. As long as routes fail independently, this provides a high degree of availability to the system.

The Chord/CFS network is an overlay that maps down to a physical IP network. Redundancy can be added to the physical network, but since the overlay topology is a function of Chord addresses that involves a randomly distributed hash function, physical locality is diffused throughout the overlay. Accordingly, it is difficult to predict the effect of physical network redundancy on the overlay network. For example, the failure of a network link will manifest itself as multiple, randomly scattered link failures in the overlay.

The diffusion of physical links across the logical Chord network tends to amplify the bad properties of a system, but not its good properties. If any link within a lookup path has low bandwidth, high latency, or low availability, the entire path suffers. Conversely, all links within a path must share the same good property for the path to benefit from it. Thus, a single bad physical link can "infect" many routes. As was measured in [8], 20% of the hosts in popular file-sharing peer-to-peer systems connect to the Internet over modems. Since Chord overlay paths traverse essentially random physical links, a simple calculation reveals that for a network of 10,000 peers with similar characteristics to those in [8], there is a 79% probability that a lookup request encounters at least one modem.

As another example, it is possible for a single physical link failure in the Internet to cause a large network partition. Consider a worst-case failure that separates an AS from the rest of the Internet: as long as all Web content within that AS is replicated outside of it, all content is available to all non-partitioned clients. However, in Chord, the number of failed routes that this single link failure will cause is proportional to the number of Chord addresses hosted within the partitioned AS, and these failed routes will be randomly distributed across both peers and content.

A final implication of the deterministic nature of routes in Chord/CFS-style systems is that it is possible for an attacker to construct a set of addresses that, if inserted into the system, will intercept all lookup requests coming from a particular member of the system. Even though mechanisms exist to prevent a peer from selecting arbitrary addresses, if a peer can insert enough addresses, it can (probabilistically) surround or at least become a neighbor of any other peer.

4 Summary

We presented a design decomposition of the WWW, TRIAD [5] and Chord/CFS [2] (as representative of recent peer-to-peer architectures [3, 6, 10]). This decomposition allowed us to describe fundamental system design differences: (1) in Chord/CFS, the content and address namespaces are equivalent, as opposed to WWW and TRIAD; (2) Chord's net-

<table>
<tr><th></th><th>WWW</th><th>TRIAD</th><th>Chord/CFS</th></tr>
<tr><td rowspan="2">Access Control</td><td>Localized bindings, hierarchical space</td><td colspan="2">Global bindings
Single host can force others to do work</td></tr>
<tr><td>Namespace and address space are decoupled</td><td colspan="2">Namespace control equivalent to address-space control</td></tr>
<tr><td>Content Replication</td><td colspan="2">Achieved through multiple, user-transparent bindings of same name</td><td>Achieved through multiple, user-aware bindings of different names</td></tr>
<tr><td rowspan="2">Path Redundancy</td><td colspan="2">Some alternate network paths</td><td>Many alternate network paths</td></tr>
<tr><td colspan="2">Can provision network for targeted content</td><td>Can’t provision, locality is diffused</td></tr>
<tr><td>Security</td><td colspan="2">Different levels of trust for different roles</td><td>Servers are routers; routers are servers
Single role, single level of trust</td></tr>
<tr><td rowspan="2">Failures</td><td colspan="2">Router failure doesn’t affect content availability
Server failure doesn’t affect routing</td><td>Server failure = = Router failure</td></tr>
<tr><td colspan="2">Local failures have local effects</td><td>Link failures diffuse throughout overlay</td></tr>
</table>

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

work topology is a deterministic function of its content and address namespace and (3) unlike the WWW, in both TRIAD and Chord, lookup and routing are unified. These differences have unexpected consequences, some of which can have serious implications to these systems' availability, security, redundancy and fault-tolerance (Figure 3).

References

[1] T. Berners-Lee, L. Masinter, and M. McCahill. RFC 1738 - Uniform Resource Locators (URL), December 1994.

[2] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Wide-area cooperative storage with CFS. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP 2001)*, Lake Louise, AB, Canada, October 2001.

[3] P. Druschel and A. Rowstron. Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP 2001)*, Lake Louise, AB, Canada, October 2001.

[4] R. Fielding. RFC 1808 - Relative Uniform Resource Locators, June 1995.

[5] M. Gritter and D. R. Cheriton. An Architecture for Content Routing Support in the Internet. In *Proceedings of the 3rd Usenix Symposium on Internet Technologies and Systems (USITS)*, San Francisco, CA, USA, March 2001.

[6] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content-Addressable Network. In *Proceedings of the ACM SIGCOMM 2001 Technical Conference*, San Diego, CA, USA, August 2001.

[7] J. Saltzer. RFC 1498 - On the Naming and Binding of Network Destinations, August 1993.

[8] S. Saroiu, P. K. Gummadi, and S. D. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. In *Proceedings of the Multimedia Computing and Networking Conference (MMCN)*, San Jose, CA, USA, January 2002.

[9] J. F. Shoch. Inter-Network Naming, Addressing, and Routing. In *Proceedings of IEEE COMPCON*, pages 72–79, Washington, DC, USA, December 1978. Also in K. Thurber (ed.), Tutorial: Distributed Processor Communication Architecture, IEEE Publ. #EHO 152-9, 1979, pp. 280–287.

[10] B. Zhao, K. Kubiatowicz, and A. Joseph. Tapestry: An Infrastructure for Fault-Resilient Wide-Area Location and Routing. Technical Report UCB//CSD-01-1141, University of California at Berkeley Technical Report, April 2001.