

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

Information-Centric Networks

Section # 6.1: Evolved Naming & Resolution

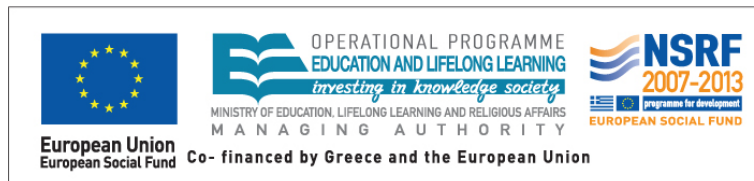
Instructor: George Xylomenos

Department: Informatics



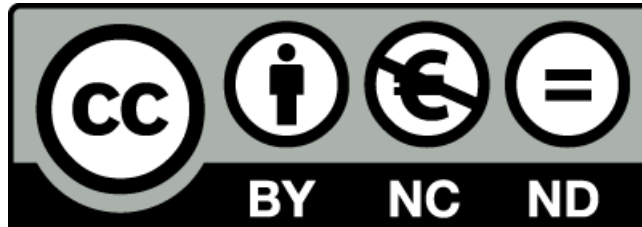
Funding

- These educational materials have been developed as part of the instructors educational tasks.
- The **“Athens University of Economics and Business Open Courses”** project only funded the reformatting of these educational materials.
- The project is being implemented as part of the Operational Program “Instruction and Lifelong Learning” and is co-financed by the European Union (European Social Fund) and national funds.



Licencing

- These educational materials are subject to a Creative Commons License.



Week 6 / Paper 1

- Untangling the Web from DNS
 - Michael Walfish, Hari Balakrishnan and Scott Shenker
 - Networked Systems Design and Implementation (NSDI) 2004
- Main point
 - The Web uses DNS to resolve hostnames to IP addresses
 - DNS limitations influence the Web
 - Web load hurts DS performance
 - Proposal to use semantic free references on the Web
 - Resolution infrastructure based on DHTs

Introduction

- DNS translates machine names to addresses
 - The commercialization of the Internet makes this complex
 - DNS has become a branding mechanism!
 - The Web simply requires a Reference Resolution Service (RRS)
 - Map references (links) to actual locations
 - DNS is not a very good RRS
 - References are tied to hosts, that is, specific machines
 - Makes replication and migration hard
- What should a good RRS offer?
 - Persistent object references: not tied to administrative domains
 - Contention-free references: no legal disputes on ownership
 - Leave the human name to reference translation to others
 - General-purpose infrastructure: not only for the web

Semantic free referencing

- Two key design principles
 - Semantic-free namespace: no explicit semantics
 - No administrative domain or other ties
 - Minimal RRS interface and factored functionality
 - Nothing else beyond reference resolution
 - Human friendly names are dealt with by someone else
- Semantic-free references (SFRs)
 - Today the web is based on human readable URLs
 - With SFR we have a two step process
 - Translate human readable names to SFRs
 - Resolve SFRs to IP addresses or other information
 - Allows migration without changing links
 - Simplifies object replication

SFR challenges

- Scalable resolution
 - Can be based on DHTs, modified to be faster than $O(\log n)$
- Security and integrity
 - Need to avoid collisions in a distributed manner
- Fate sharing
 - Disconnected domains should still function
- Trust and financing
 - Incentives for external nodes to serve my SFRs
- Canonical names (left to higher layers)
 - Need names that users can remember
- Confidence (left to higher layers)
 - Users expect to understand the URLs they use

SFR design

- SFR essentials
 - Uses a DHT to map SFRtags to o-records
 - SFRtags: 160 bit strings identifying objects
 - O-records: metadata associated to SFRtags
 - Location: (ip,port), (DNS,port), SFRtag
 - Oinfo: application specific metadata
 - TTL: time to live, a caching hint
 - Location can contain multiple fields
 - Either IP addresses, DNS names or other SFRtags
 - Allows multiple degrees of indirection
 - Oinfo: transparent to the SFR, only for applications
 - For the web could be protocol (HTTP/HTTPS) or a pathname

SFR model and components

- DNS relies on local servers for local names
 - Not possible to do with SFRs
- SFR requires a common trusted infrastructure
 - Could be provided by NSF, EU
 - Eventually could become commercial and competitive
- The infrastructure is critical
 - Should be well managed and well connected
 - It is not a simple P2P system!
- SFR components
 - A collection of managed nodes, the SFR servers (DHT)
 - SFR relays connect to SFR servers similarly to resolvers
 - SFR client software uses SFR relays to resolve SFRs

Security and integrity

- Creation of SFRs in a distributed manner
 - SFRtag = hash (public key, salt)
 - Send to SFR infrastructure the SFRtag with additional info
 - The o-record with the metadata
 - Public key, salt and version
 - Signature(o-record, salt, version)
 - SFR infrastructure verifies hash and signature
- Lookup of SFRtags
 - Returns all the above information
 - Client can verify that the o-record is valid
 - Does not need a PKI: the public key is tied to the tag and record
- Modifications to keys or o-records
 - Next version number, SFR keeps all previous versions

Latency, fate sharing, scoping

- Three levels of TTL-based caching
 - Clients cache entire o-records
 - SFR servers cache IP addresses of other SFR servers
 - SFR servers cache o-records for load balancing
- Fate sharing and scoping
 - A domain's SFRtags can be stored anywhere
 - A local org-store is used to additionally store local SFRtags
 - All modifications first go to the org-store and then to the DHT
 - If the DHT does not reply, then the change will be sent later
 - Lookups ask the org-store and the DHT in parallel
 - If the DHT does not reply, rely on the org-store
 - If the DHT replies, update the org-store if needed
 - Scoping requires SFRtags to be stored only in the org-store

Web over SFR

- Links have the form `sfr://f012120.../optional_path`
 - The bit string resolves to an o-record
 - The o-record includes location (IP or DNS) and oinfo
 - Oinfo includes the protocol and (maybe) an additional path
 - The optional-path is added at the end of the o-record path
- The web SFRtag can work in different ways
 - The oinfo path may be a full path: entire link in the SFR
 - The oinfo path may be a directory: a set of pages in the SFR
 - The oinfo path may be empty: only the server in the SFR
 - The remaining path is the optional_path in the link
- Replication: simply return many location tuples
 - Can work at any level (page, directory, host), unlike DNS

Human usability challenges

- Canonical names & user confidence
 - Humans are used to human friendly URLs
 - But increasingly we do not type the URLs
 - They come from web pages, e-mails or search engines
 - SFR requires canonicalization services
 - Translation of human friendly names to SFRtags
 - SFR can support different granularities though
 - Host, page, directory
 - Bootstrap by embedding SFRtags to search engines in browsers
 - Confidence can be enhanced by third parties signing SFRtags
- Pragmatics
 - Relative references: include hints for local SFRtags
 - Avoid going through the SFR infrastructure for relative references

Implementation

- SFR servers use DHash over Chord
 - Applications use an SFR client to get and put o-records
- Web proxy to translate URLs to SFRtags
 - The web client asks the proxy for a URL
 - The proxy translates it to an SFRtag
 - The salt is a hash of the URL
 - The proxy uses the SFR client to ask for the o-record
 - If the o-record does not exist, it is created and entered to the SFR
 - This requires a DNS lookup to discover the information
 - Essentially parts of the DNS are entered into SFR

Evaluation

- Experiments on PlanetLab
 - SFR consists of 130 physical nodes, 390 virtual nodes
 - Web proxies at 3 nodes
 - Aggressive caching makes the DHT very fast (2-3 hops)
 - Latency similar to DNS
- Simulations based on NLANR traces
 - Simulated a 1000 node SFR
 - 2-3 hops per request (similar to DNS)
 - Small changes with a reasonable churn rate
 - The SFR is a managed infrastructure, not a P2P application
 - Churn is expected to be low

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS**

End of Section # 6.1

Course: Information-Centric Networks, **Section # 6.1:** Evolved Naming & Resolution

Instructor: George Xylomenos, **Department:** Informatics

