Network of Information

Börje Ohlman
& 4WARD colleagues
Network of Information

Today’s Internet focuses on nodes

Future Information-centric Network focuses on information objects and real world objects

In today’s Internet, accessing information is the dominating use case!
Problems Resulting from a Host-centric View

- No common *persistent naming scheme* for information
  - Information is named relative to the box they are located in, URLs resolves to IP-addresses
    - Moving information = changing it's name („404 file not found“ errors)
- Mobility and multihoming for hosts and networks is problematic due to the semantic overload of IP-addresses
- No consistent *representation of information* (copy-independent)
  - No consistent way to keep track of *identical copies*
  - Different *encodings* (e.g., mp3, wav) worsen problem
- Security is host-centric
  - Mainly based on *securing channels* (encryption) and *trusting servers* (authentication)
  - Can't generally trust a copy received from an untrusted server

Problems can be solved in a consistent manner via an information-centric architecture
NetInf Scenarios

Content distribution
- VideoOnDemand, Live TV, Web pages
- Caching can be built-in from the beginning
- Information can be retrieved from the closest available source
- Common dissemination infrastructure for all applications, including network support
API for accessing any type of object, regardless of location

NetInf

getObject(objectID)
getObject(attr1, attr2, attr3)

API

Internet
Mobile networks
Broadcast TV/Radio
Real world
Organize Information – IO, DO and BO definitions

An Information Object is a set of attributes defining the semantics of a data object. An IO may refer to a piece of music, a film or a webpage. Can be static, dynamic or real-world objects, including streams and services.

Data Object (DO) Sub-class of IO holding attributes for bit-level objects and pointer(s) to the actual data.

Bit-level Object (BO) A specific sequence of bits, independent of any semantic meaning, also independent of where they exist, like in a file, on the wire, in the air or in a primary memory.
| Tag | P=Hash(PublicKey_{Owner}) | L={Hash(C) | String} |
|-----|--------------------------|---------------------|

**Tag**
- Defines the format
  - Hash algorithm used (SHA1, MD5, …)

**Principal (P)**
- Object ‘publisher’ (optional)
  - Owner
  - Creator
  - Anonymizing service

**Label (L)**
- Identifying individual object published by Principal
  - Hash of object or label created by principal

IDs have no hierarchical structure. Strong influence on name resolution!
NetInf Architecture Overview

NetInf node

Applications

Application programming interface
publish (…)
resolve (…)
join (…)

NetInf API

NetInf Storage API

NetInf App X API

Storage engine

App X engine

Resolution engine

Local storage engine

Local resolution engine

Transport control engine

Cache engine

TCP/UDP/IP

IP multicast

Generic Path(s)

INI

NetInf Additional Services

Storage protocol(s)
STORE(…)
NetInf App. X protocol

Name resolution protocol(s)

PUT (…)
GET (…)
...

NetInf Transport control protocol

12/11/2009