NetInf: The Network of Information

Dirk Kutscher
NEC Laboratories Europe
**NetInf Work**

- SAIL NetInf work package
  - NEC Laboratories Europe
  - DOCOMO Eurolabs
  - Ericsson
  - France Telecom
  - Nokia Siemens Networks
  - Swedish Institute of Computer Science
  - Telecom Italia
  - Trinity College Dublin
  - Universität Paderborn

Presentation based on contributions from many individuals

Special thanks to
Bengt Ahlgren, Börje Ohlman, Elwyn Davies, Holger Karl
Christian Dannewitz, Luca Muscariello,
Matteo D’Ambrosio
for providing material
Information-Centric Networking
More than CDN++

Canonical ICN motivation

- Imminent traffic volume explosion
  - Video distribution as (literally) a killer application
  - Resource management issues unsolved today

- ICN needed as a generalized CDN
  - Leveraging ubiquitous in-network storage
  - Extending caching and access to named data to other applications than just web

Cisco Forecasts 3.6 Exabytes per Month of Mobile Data Traffic by 2014
Problems with CDNs today

- Current CDN models good for offloading content distribution to CDN providers
- Caches mostly located at exchange points
  - Colocation
- No benefit to access network operators
  - No traffic reduction
  - No revenue sharing
**Operator CDN?**

- Operators are looking to move into operator CDN
  - Running their own CDN
  - Cooperating with incumbent CDN operators

- Does not solve all problems
  - Traditional CDN gear not designed for large-scale operation: operational cost can be prohibitive
  - Wireless communication will still be inefficient
Access Link Dilemma

- Cannot leverage broadcast mechanisms of broadcast-enabled link layers
  - Connection-oriented model requires repeated transmission of identical content bits
  - Tunneling in current mobile network architectures does not help either
Transport Protocol Performance

- Different sources for delay and packet loss
  - Wireless scheduling
  - Queuing in gateways

- Today: one TCP control loop
  - Single packet loss (wireless) will affect congestion window for the whole connection
  - Delay variations (scheduling) can affect RTT estimation
Host-centric networking

Connect to Server X and get object B

Trusted Server

Secure Connection

Server X
Information-Centric Networking

Trustable copy of object B

Get object B

Untrusted connection

Untrusted host
ICN-based Information Retrieval 101

Requestor 1

Requestor 2

Original Content “XY1”

Content Repository

Owner “Joe”
ICN Core Properties

• Accessing named data as a first-order network principle
  – Transmission of self-contained units

• Name-content-binding validation and other security services based on object/naming security
  – Not based on connection security

• Ability to leverage ubiquitous in-network memory
  – Rate adaptation
  – Repair (efficient re-transmissions)
  – Sharing (Re-use)
NetInf Thin Hour Glass Waist

a) Schema

Applications
Transport
NetInf
Under-\text{lay 1} \quad \text{CL 1} \quad \text{CL 2} \quad \text{Under-\text{lay 2}}
Physical

b) Example

InFox \quad \text{InBird} \quad \text{InTube}
Request Scheduling
NetInf
Under-\text{lay 2} \quad \text{CL 1} \quad \text{CL 2}
Physical

- Ethernet
- TCP/IP
- WLAN
NetInf Naming

• **Problem Statement**
  – Need unique object identification, request/routing keys
  – Need ways to validate name-content binding („have I received what I asked for?“)

• **Requirements**
  – Uncoordinated name creation – NetInf nodes should be able to create NDOs and names independently
  – No PKI dependency – difficult to deploy, especially in less-well connected networks
  – Algorithm agility: today’s hash/encryption algorithm may be considered insecure in the future – have to be able to adopt new algorithms without changing protocols and all deployed NetInf nodes
ICN Naming and Security

• In ICN, availability can be threatened by content-level denial of service attacks
  – Flooding the network with objects that don’t have correct name-content binding or provenance

• This is directly related to naming

• Two types of objects
  – Immutable: can use hash-based names to assure name-content binding
  – mutable/dynamic: requires other forms of name bindings: Public-Key Cryptography...

[Ghodsi et. al; Naming in Content-Oriented Architectures; ACM SIGCOMM ICN Workshop 2011]
Named Data Object Structure

Object Name

Object in Message

- multipart/mixed
  - Cryptographic info
  - Signatures etc.

- multipart/mixed
  - Named data object
    - application/steam-meta+xml
    - Application-specific meta data
    - application/binary
    - Actual object bits

Signature coverage
Naming Bindings

• **Real-World Identity (RWI)**
  - “cnn.com”
  - Provenance
  - Principal for the data (person or organization)

• **Name**
  - Name used by network to identify and provide access to the object
  - Created by principal

• **Public Key**
  - Principal is associated with public-private key pair
  - Public key can be used by receiver to verify that RWI signed the object
Intrinsic Bindings

- **Human-readable names**
  - com/cnn/headline/2012/09/11
  - Human readable – Name-to-RWI binding intrinsic
  - Name-to-public-key binding validation requires external mechanisms – PKI

[Ghodsi et. al; Naming in Content-Oriented Architectures; ACM SIGCOMM ICN Workshop 2011]
Intrinsic Bindings

- Self-certifying names

| Type | A = Hash(PK_{IO}) | L = \{identifier attr.\} |

- Hash of public key part of name – Name-to-public-key binding intrinsic
- Name-to-RWI binding must be established through additional means
  - Human-friendly names mapped to objects names, SEARCH etc.

[ Ghodsi et. al; Naming in Content-Oriented Architectures; ACM SIGCOMM ICN Workshop 2011 ]
NetInf Naming

• Quite many objects are immutable
  – Hash-based names work fine

• Dynamic objects
  – Prefer self-certifying names against PKI dependencies

• NetInf names: flat labels with minimal structure
  
  \[
  L = \{\text{identifier attr.}\}
  \]
  
  \[
  A = \text{Hash}(\text{PK}_{10})
  \]
  
  \[
  \text{Hash}(\text{NDO})
  \]
NetInf Names: Missing Elements

• Not human-friendly
  – Name-to-RWI binding not intrinsic

• Cannot (easily) be guessed, programmatically generated
  – Receivers need to know (valid) name when wanting to send a request
  – Catalog files, hypertext etc. can be used for that
    • Another level of indirection in naming
  – Or: SEARCHing for NDOs (and corresponding names)
Two main options:
1. Name resolution translating to a namespace designed for global routing (e.g., IP)
2. Routing directly on information object names

Scalability depends on namespace:
- Different for flat and hierarchical namespaces
- Aggregation of routing information is key to scalability
  - 1 routing entry for many objects

Hierarchically structured name: /com/cnn/headlines/today
Non-Hierarchically structured name: UyaQV-Ev4rdLoHyJJWCi11OHfrYl
How to use DHTs in ICN?

- **Option 1: Use the DHT to store the actual objects**
  - Needed: Extension to store same object on multiple nodes (exists)
  - Possible problem: Places where an object is stored are hard to control and not necessarily optimal to access
  - Issue is *topological embedding* of the DHT in the actual network: How to ensure that nodes with similar IDs are topologically close to each other?
    ! Questionable!

- **Option 2: DHT stores, for each object ID, a list of addresses** (from an underlying network) where this object can be found
  - List can again be stored at multiple places
  - Question: How to make sure that resolution latency is small, at least on average?
    - Depends on access pattern: Which objects are requested from where?
  - Question: How to pick the “best” address out of this list?
    ! Usually preferred
Name resolution in ICN – Optimization goals

• Option 1: Ask DHT for list of addresses, decide locally
  – Minimize time to resolve name into list of addresses
  – Then, minimize time to retrieve object
• Option 2: Ask DHT to tell providing node to send data
  – Jointly minimize time to resolve and retrieve object

• Performance highly depends on access patterns (which node requests which data how often?) and network structure

• One idea: adapt NRS structure to network structure
  – … without trying to do a conventional topological embedding on node IDs
Example NRS: Multi-level DHT (MDHT)

- Challenge: how to arrange name resolution and data copies to minimize latencies?
- Idea: exploit natural structure of networks and desired provider autonomy
  - Build local, hierarchically structured DHT systems at multiple levels
    - E.g., provider networks
  - At level x, register all objects stored in level x
  - NRS node participates in all levels above itself

D’Ambrosio, Dannewitz, Karl, Vercellone; MDHT: A Hierarchical Name Resolution Service for Information-centric Networks; ACM SIGCOMM 2011 ICN Workshop; August 2011
Alternative: Re-Use Routing Infrastructure

- MDHT and similar approaches require new DHT infrastructure
- How to build a NetInf DFZ?
- In the following: some thoughts how NetInf can play with existing inter-domain infrastructure
ICN routing issues

● Want to route globally on large object namespace
  - $O(10^{12})$ today, need to aim higher, perhaps $O(10^{15})$
  - Single $O(10^{12})$ database possible today

● So name resolution option should be feasible
  - And if aggregating on publisher prefixes (129M), it is easy
  - But that ties objects to location in an undesirable way

  - But BGP-like protocol can likely not scale to this level
Hybrid resolution/routing scheme

Name Resolution Service

ni://example.com/foo;YY
http://D
ftp://D2
gopher://D2x

Named object
ni://example.com/foo;YY

SCALABLE & ADAPTIVE INTERNET SOLUTIONS
Hybrid resolution/routing scheme

- NetInf experiments with name-based routing and name resolution
- Name-based routing can be used in small domains
- Resolution to *routing hints* for global scalability
  - A hint does NOT name the final end-point
  - More like an anycast address
- Use *explicit aggregation*
  - Several routing hints each at different aggregation levels, for instance, ISP, ISP on a particular continent, ISP site, server cluster, ...
  - Use the most specific hint that you have a routing entry for
Hybrid scheme example

- `ni://sics.se/foo;XYZ`
- Edge domains can route directly on this name
- To make global routing scalable, a set of routing hints are looked up, for example (using IP addresses):
  - `130.236.0.0, prio=1` – SUNET /14 network
  - `130.237.0.0, prio=2` – KTH /16 network
  - `130.237.211.0, prio=3` – KTH /24 subnetwork
  - `193.10.64.0, prio=4` – SICS /22 network
- Forward to next-hop of routing entry that matches hint with highest priority
- Global table only needs the lowest priority hint
Request Flow Control Protocol

Objectives

- **Reliability**
  - Reliable data transfers by re-sending requests messages in case of response message loss (timer-based)

- **Efficiency**
  - Minimize data completion times
  - AIMD for adapting request window size to keep to intended maximum rate

- **Fairness**
  - Fair bandwidth allocation among flows sharing a path
Protocol Description

- Intermediate nodes keep track of outstanding requests
  - For a limited amount of time

- Requested NDOs are delivered back on reverse path

- Nodes cache NDOs

- Optional multi-path operation (request striping)
  - Different latencies on different next-hop links

- Receiver window
  - Maximum number of outstanding requests a requestor is allowed to have

- RTT estimation for expiration timer adaptation
Current Work

- Finding good window size increase/decrease factors and request retransmission timer calculation functions
- Simulating behavior in different settings
  - Bottleneck configurations
  - Interest distribution
- Implementing protocol for UDP CL network

Figure 3.4: (a) Topology, (b) Delivery time and RTT$_k$ in case II with capacity (C)
C1=C3=100Mbps, C2=40Mbps with fixed $\tau$
Running Code

• NetInf
  – NI naming scheme, HTTP convergence layer, caching, routing
  – Different interoperable implementations: C, Clojure, Java, Python, Ruby

• OpenNetInf
  – Focus on multi-layer DHT name resolution
  – Java, with Browser plugins
  – [http://netinf.org](http://netinf.org)

• Plus proprietary developments at SAIL partners
IRTF ICN Research Group

• Objectives
  – Couple ongoing ICN research with solutions that are relevant for evolving the Internet at large
  – Produce a document that provides guidelines for experimental activities in the area of ICN
  – Initiate discussions about the desirable interfaces/APIs to an ICN (e.g., for application layer and network management)

• Short-term goals
  – Survey document
  – Problem statement
  – Reference baseline scenarios for performance comparisons

www.ietf.org/icnrg
Summary

• ICN a promising approach for making accessing named data work better
  – Whether it can/should be stretched beyond that is an interesting discussion
  – Building on previous work: caching, DTN, P2P
  – Lots of momentum: SIGCOMM ICN WSs, IRTF ICNRG, Dagstuhl seminars

• NetInf: ICN with non-hierarchical names
  – Independent object/name creation, no PKI dependency
  – Heterogenous networks – not just for well-connected ones
  – Thesis for NetInf: name resolution and routing can scale