Less Pain, Most of the Gain: Incrementally Deployable ICN

Seyed K. Fayazbakhsh, Yin Lin, Amin Tootoonchian, Ali Ghodsi, Teemu Koponen, Bruce Maggs, KC Ng, Vyas Sekar, Scott Shenker
A high-level view of ICN

- Equip network with content caches
- Decouple “what” from “where”
- Bind content names to intent
- Route based on content names e.g., find nearest replica

Today: Fetch from server IP
Gains of deploying ICN

- Lower latency
- Reduced congestion
- Support for mobility
- Intrinsic security

e.g., CCN, DONA, NDN, 4WARD ....
Pains of deploying ICN

- Routers need to be upgraded
- Routing needs to be content based

e.g., CCN, DONA, NDN, 4WARD ....
Motivation for this work

Gains

• Lower latency
• Reduced congestion
• Support for mobility
• Intrinsic security

Can we get ICN gains without the pains?
e.g., existing technologies?
e.g., incrementally deployable?

Pains

• Routers need to be upgraded with caches
• Routing needs to be content based
Approach: Attribute gains to tenets

Quantitative
- Lower latency
- Reduced congestion

Qualitative
- Support for mobility
- Intrinsic security

- Decouple “what” from “where”
- Bind content names to intent
- Equip network with content caches
- Route based on content names
Key Takeaways

• To achieve quantitative benefits:
  ➔ Just cache at the “edge”
  ➔ With Zipf-like workloads, pervasive caching and nearest-replica routing don’t add much

• To achieve qualitative benefits:
  ➔ Build on HTTP

Basis for incrementally deployable ICN
Outline

• Background and Approach

• *Analyzing quantitative benefits*

• Qualitative benefits → Incrementally deployable ICN

• Discussion
Design space of caching

• Quantitative benefits are largely due to caching

• Two key dimensions to this design space:
  – Cache placement
    • E.g., everywhere? Edge?
  – Request routing
    • E.g., shortest path, nearest replica?
Representative points in design space

<table>
<thead>
<tr>
<th>Cache Placement</th>
<th>Request Routing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ICN-NR</strong></td>
<td>Everywhere</td>
</tr>
<tr>
<td><strong>Edge</strong></td>
<td>Only at edge nodes</td>
</tr>
</tbody>
</table>
Simulation setup

PoP-level topologies (Rocketfuel) augmented with access trees

Assume name-based routing, lookup incurs zero cost

Real CDN request logs

Cache provisioning ~ 5% of objects
Uniform or Proportional

LRU replacement

PoP

Access tree node

Access tree

Edge

1

2

3

4

[Diagram of network topologies with PoPs and access nodes]
Gap between architectures is small (< 10%)
Similar results for congestion + server load
Performance gain of ICN-NR over EDGE (%)

Even in best case, ICN-NR is only 17% better

Gap can be easily reduced
Implications of Edge Caching

• Incrementally deployable
  – Domains get benefits without relying on others

• Incentive deployable
  – Domains’ users get benefits if domain deploys caches
Outline

• Background and motivation

• Approach

• Quantitative benefits of ICN

• Qualitative benefits $\rightarrow$ Incrementally deployable ICN

• Discussion
Revisiting Qualitative Aspects

1. Decouple names from locations

   Build on HTTP
   – Can be viewed as providing “get-by-name” abstraction
   – Can reuse existing web protocols (e.g., proxy discovery)

2. Binding names to intents

   Use self-certifying names
   e.g., “Magnet” URI schemes

   Extend HTTP for “crypto” and other metadata
idICN: Content Registration

Name Resolution System

Register L.P.idicn.org

Reverse Proxy

Publish content

Origin Server

P = Hash of public key

L = content label

e.g., http://en.5671....fda627b.idicn.org/wiki/
idICN: Client Configuration

Name Resolution System

Proxy
Edge
Cache

Automatic Proxy Discovery
e.g., WPAD

Client

Reverse Proxy

Origin Server
idICN: Content Delivery

1. Rqst L.P.idicn.org

2. Name resolution

Proxy Edge Cache

3. Rqst by address

4. Fetch

Origin Server

5. Response + Metadata

Reverse Proxy

6. Response

Client

Name Resolution System

Try it out: www.idicn.org
Conclusions

• Motivation: Gains of ICN with less pain
  – Latency, congestion, security
  – Without changes to routers or routing!

• End-to-end argument applied to ICN design space

• Can get most quantitative benefits with “edge” solutions
  – Pervasive caching, nearest-replica routing not needed

• Can get qualitative benefits with existing techniques
  – With existing HTTP + HTTP-based extensions
  – Incrementally deployable + backwards compatible

• idICN design: one possible feasible realization
  – Open issues: economics, other benefits, future workloads ..