Information-Centric Networking
From Point-to-Point Communication To Content Distribution

Liang Wang
liang.wang@cl.cam.ac.uk
Content

● Motivation & Key Components
● Naming Schemes
● Routing & Mobility
● In-Network Caching
● Well-Known Designs
● Service-Centric Networking
The Big Picture of Today’s Internet

A very high-level abstraction of current Internet: ISPs are interconnected with each other, along with big service providers. End-users are attached to various ISP networks.
Why Content Networking Is Proposed?

- Content distribution is the primary task for today’s Internet. E.g., the estimated video traffic will reach 79% of the Internet traffic by 2018.
- Traditional paradigm of communication network is Point-to-Point.
- Point-to-Point paradigm has many drawbacks when dealing with large-scale content distribution - efficiency, security and privacy.

Content consumer only cares what it is instead of where it is from.
The Key Architectural Components

ICN is a clean-slate redesign of the current Internet infrastructure,

- Content is accessed by name.
- Caching is universal in the network.

ICN tries to solve the problems confronting the current Internet, e.g., content distribution efficiency, security, network congestion and etc.

Meanwhile, ICN also poses new challenges on cache management, content addressing, routing and etc.
Before We Continue, Remember

- ICN is not a silver bullet.
- There is no one-fits-all solution in system which gives you all the benefits (e.g. efficiency, simplicity, scalability, security, privacy, adaptability, so on and so on).
- We always need to balance different trade-off in engineering.
- System building is both art and science!
The Quandary Betw. Locator and Identifier

- We need two mappings from Identifier $\rightarrow$ Locator $\rightarrow$ Path.
- It’s all about “finding a path to what you want”, which, we have been doing for thousands of years in different forms …
  - Human society in old days: social knowledge, real map.
  - Internet: DNS, various routing algorithms.
How Do You Actually Name Content?

Three naming schemes in ICN, two dominate the literature.

- **Hierarchical naming:**
  - similar to nowadays DNS,
  - correlates to underlying network topologies.

- **Flat naming:**
  - usually done by hashing,
  - self-certified.

- **Attribute-based naming:**
  - more expressive, richer in semantic structures,
  - can combine with previous two naming schemes.
Which Is the Best Naming Scheme?

● Recall, “No silver bullet in system engineering!”

● Each scheme has its own pros and cons:
  ○ routing complexity
  ○ scalability
  ○ security
  ○ expressiveness
How A Request/Interest Is Routed?

- Recall, there are always two basic functionality:
  - Name resolution: identifier → locator
  - Routing: locator → path

- How routing is done depends on ICN architectures:
  - Source routing: PURSUIT
  - Hop-by-Hop routing: CCN
  - DHT-like routing: MDHT
How Mobility Is Handled in ICN?

- Receiver mobility is **trivial**. It is inherently handled by design.
- Publisher (or source) mobility is **non-trivial**.
- Simultaneous handoff makes life even more **complicated**!

<table>
<thead>
<tr>
<th></th>
<th>Avg. Latency</th>
<th>Handoff Delay</th>
<th>Simultaneous Handoff</th>
<th>Scalability</th>
<th>Single Point of Failure</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>MobiCCN</td>
<td>Medium</td>
<td>Low</td>
<td>Yes</td>
<td>High</td>
<td>No</td>
<td>Medium</td>
</tr>
<tr>
<td>Sender-Driven Msg</td>
<td>Low</td>
<td>High</td>
<td>No</td>
<td>High</td>
<td>No</td>
<td>Low</td>
</tr>
<tr>
<td>Rendezvous Point</td>
<td>Low</td>
<td>Medium</td>
<td>Yes</td>
<td>Medium</td>
<td>Yes</td>
<td>Low</td>
</tr>
<tr>
<td>Indirection Point</td>
<td>High</td>
<td>Medium</td>
<td>Yes</td>
<td>Low</td>
<td>Yes</td>
<td>High</td>
</tr>
<tr>
<td>Interest Forwarding</td>
<td>Medium</td>
<td>Low</td>
<td>Yes</td>
<td>Medium</td>
<td>No</td>
<td>High</td>
</tr>
</tbody>
</table>

**TABLE III: Comparison of different mobility schemes**
Why In-Network Caching Is Different?

The fundamental difference between a single cache and a cache network:

The topological structure becomes a system parameter in ICN designs.

- Content caching ≠ Content addressing
- Effective capacity ≠ Aggregated cache size
- Local optimum ≠ Global optimum

The whole system should not be treated as a simple “entity”, we need examine the internal topological structures of a cache network.
Model of In-Network Caches

Given a group of networked caches, how to utilize them smartly and efficiently in order to push the system to its optimal state?
Model of In-Network Caches

Given a group of networked caches, how to utilize them smartly and efficiently in order to push the system to its optimal state?

Input
- Server - content holder
  \[ I = \{ f_1, f_2, f_3, \ldots, f_N \} \]
- ISP - networked caches
  \[ C = \{ C_1, C_2, C_3, \ldots, C_M \} \]
- Client - requester
  Zipf distribution

Caching Strategy

Objective
1. inter-ISP traffic
2. intra-ISP traffic

Output
Caching Decision
\[ X = \{ X_{11}, X_{12}, \ldots, X_{NM} \} \]
Collaborative In-Network Caching

What is purpose of collaboration?
- Discovering content;
- Reducing duplicates.

How expensive is the collaboration?
- For global optimal solution;
- For off-path collaboration.

How effective is the collaboration?
- Filtering effect.
Well-Known Designs - DONA

Well-Known Designs - CCN

Well-Known Designs - PSIRP

Well-Known Designs - NetInf

## Architectural Comparison

<table>
<thead>
<tr>
<th></th>
<th>DONA</th>
<th>CCN</th>
<th>PSIRP</th>
<th>NetInf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Namespace</td>
<td>Flat with structure</td>
<td>Hierarchical</td>
<td>Flat with structure</td>
<td>Flat with structure</td>
</tr>
<tr>
<td>Name-data integrity</td>
<td>Signature, PKI independent</td>
<td>Signature, external trust source</td>
<td>Signature, PKI independent</td>
<td>Signature or content hash, PKI indep.</td>
</tr>
<tr>
<td>Human-readable names</td>
<td>No</td>
<td>Possible</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Information abstraction model</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>NDO granularity</td>
<td>Objects</td>
<td>Packets</td>
<td>Objects</td>
<td>Objects</td>
</tr>
<tr>
<td>Routing aggregation</td>
<td>Publisher/explicit</td>
<td>Publisher</td>
<td>Scope / explicit</td>
<td>Publisher</td>
</tr>
<tr>
<td>Routing of NDO request</td>
<td>Name-based (via RHs)</td>
<td>Name-based</td>
<td>NRS (rendezvous)</td>
<td>Hybrid NRS and name-based</td>
</tr>
<tr>
<td>Routing of NDO</td>
<td>Reverse request path or direct IP connection</td>
<td>Reverse request path using router state</td>
<td>Source routing using Bloom filter</td>
<td>Reverse request path or direct IP connection</td>
</tr>
<tr>
<td>API</td>
<td>Synchronous get</td>
<td>Synchronous get</td>
<td>Publish/subscribe</td>
<td>Synchronous get</td>
</tr>
<tr>
<td>Transport</td>
<td>IP</td>
<td>Many including IP</td>
<td>IP/PSIRP</td>
<td>Many including IP</td>
</tr>
</tbody>
</table>

From Static Content to Dynamic Service

- Information should **not** only refer to static content.
- Recursive definition: Information = $f(\text{Information})$.
- $f$ is a service which filters, edits, combines existing information to provide new information.
What Are the Benefits of Service Caching?

- Better localised communication: latency, bandwidth, availability …
- Better control on sharing conventional static content.
- Flexible policy configuration but with simpler architecture.
- Key services in emergency and disaster scenarios.
- Efficient access to popular Internet cloud-based services.
A Glimpse on Service-Centric Networking
Thank you. Questions?
Conclusion