Decoupling Information and Connectivity via Information-Centric Transport

Hila Ben Abraham, Jyoti Parwatikar, John DeHart, Adam Drescher, and Patrick Crowley

Computer Science and Engineering (CSE)
Washington University in St. Louis
Outline

- The ICN abstraction and its promise
- Forwarding strategies
- Information-Centric Transport (ICT)
- Sync
- Push notifications

- Topics may seem unrelated, but they are important components in the story of: Decoupling information and connectivity
Outline

- The ICN abstraction and its promise – Motivation
- Forwarding strategies – The source of the problem
- Information-Centric Transport (ICT) – Proposed solution
- Sync – An ICT example
- Push notifications – An ICT example

- Topics may seem unrelated, but they are important components in the story of: Decoupling information and connectivity
Motivation - The Promise of ICN

Information-Centric Networking (ICN) is based on the "request for named data" abstraction.

What is the content for this name?

App

Interest
Motivation - The Promise of ICN

Information-Centric Networking (ICN) is based on the “request for named data” abstraction.

What is the content for this name?

Network’s job to find and retrieve the named data

App

Interest

Data
Motivation - The Promise of ICN

Information-Centric Networking (ICN) is based on the “request for named data” abstraction.

What is the content for this name?

Network’s job to find and retrieve the named data

It doesn’t matter how
Motivation - The Promise of ICN

Information-Centric Networking (ICN) is based on the “request for named data” abstraction.

ICNs promise that applications can operate in the Information Plane, dealing only with namespaces, and trust relationship. Applications can be decoupled from connectivity.
The Problem

• Forwarding strategies binds applications to the details of connectivity in an unsustainable way.

• Two questions that demonstrate this coupling:
  1. What is the role of the forwarding strategy component?
     – It is underspecified. Strategies are used by both applications and the network.
  2. Who chooses a forwarding strategy?
     – Neither application developers nor the network operators are qualified!

• An application can break when the strategy selection or implementation changes.
The problem lies in ICNs, such as NDN and CCN, having one architectural component, the forwarding strategy, that both reconciles application and network considerations, and manages the interests of both application developers and network operators.
Step 1: Specify Strategy's Role

What is the content for this name?

App

Interest

Data

Network’s job to find and retrieve the named data
Step 1: Specify Strategy's Role

- In practice, there is always an actual, real-world connectivity present. E.g. a collection of one or more connectivity options.
Step 1: Specify Strategy's Role

• In practice, there is always an actual, real-world connectivity present. E.g. a collection of one or more connectivity options.

• Existing connectivity characteristics must be relied upon to answer the questions above.

What is the content for this name?

- Who might have the data?
- What is the most efficient way to retrieve it?
Step 1: Specify Strategy's Role

- In practice, there is always an actual, real-world connectivity present. E.g. a collection of one or more connectivity options.
- Existing connectivity characteristics must be relied upon to answer the questions above.
- Applications operate in the information plane, but the network is required to satisfy the “request for named data” while operating in the connectivity plane.
We specify the forwarding strategy as the architectural component that bridges the information and connectivity planes in ICN.
Step 1: Specify Strategy's Role (2)

We specify the forwarding strategy as the architectural component that bridges the information and connectivity planes in ICN.

- Therefore, a forwarding strategy
  - should implement mechanisms with respect to local network characteristics.
  - should not implement information-oriented mechanisms.
  - can be safely chosen and deployed by network operators, according to the connectivity they manage.
Is This Enough?

- Removing name-based strategy means:
  - Applications should not rely on strategy mechanisms.
  - Applications should only use end-point libraries.
- This is not enough!
  - Same in-network processing for all application.
  - Applications must respond to connectivity events.
    - What if we have intermittent or lossy links?
    - What if there is never a synchronous end-to-end path?
- ICN is channel-less, but applications are still bound to the end-to-end communication model.
  - But what if we could deploy information-oriented process in the network?
Step 2: Information-Centric Transport (ICT)

- ICT is an abstraction, and a distributed service, that can implement information-oriented mechanisms in the network.
- An ICT consists of two components:
  - an API for applications at the end hosts,
  - and an intermediate process that runs in the network.
- An ICT must be designed to support a specific, but broadly applicable set of application requirements.
  - and for scalability reasons, those needs must be shared among different types of applications.

![Diagram](Diagram.png)
How can In-Network Transport be Scalable?

• According to the end-to-end principle, application-specific features should never reside in the network.

• Therefore, an ICT must be implemented to capture abstractly a specific set of application-level needs, and for scalability reasons, those needs must be shared among different types of applications.
  – This abstract application needs can be purely semantic to performance and reliability.

• While Forwarding Strategies implement connectivity-oriented mechanisms, an intermediate ICT implement information-oriented mechanisms.
Demonstrating the Concept of ICT

• We do not have a clear understanding of the true needs and requirements of ICN applications.

• We demonstrate the ICT concept using two ICTs designed to support specific application semantics
  – ICT-Sync: retrieves all the names under a specific application prefix.
  – ICT-Notify: pushes the latest content under a namespace.

• In both of the implemented ICTs, we use the ICT namespace to express the application semantics.
Why a Sync ICT?

• What is Sync?
  – In ICN, sync is a service that establishes data consistency over time among multiple participants sharing the same prefix.
  – Sync notifies others when a participant adds content under the shared prefix.
Why a Sync ICT?

- **What is Sync?**
  - In ICN, sync is a service that establishes data consistency over time among multiple participants sharing the same prefix.
  - Sync notifies others when a participant adds content under the shared prefix

Bob adds a message:
Name: ChatApp/Room-wustl/BobMsg/1
Content: “Hi everyone”

Sync will let Alice and Ted know that ChatApp/Room-wustl/BobMsg/1 Was added to the shared dataset
Why a Sync ICT?

• What is Sync?
  – In ICN, sync is a service that establishes data consistency over time among multiple participants sharing the same prefix.
  – Sync notifies others when a participant adds content under the shared prefix.

• Sync is a primitive shared by different types of applications (Dropbox-style file sharing, chat applications and more).
  – Because names can represent different types of data
ICT-Sync

- ICT-Sync is a proof of concept implementation and an enhancement of ChronoSync.
  - Uses the same data structure and namespace design.
- The intermediate ICT-Sync
  - responds to sync updates by automatically fetching the relevant content.
  - validates and saves the full Data packet, including the original signature.
  - serves as a provider of the fetched content.
  - can be configured to use either persistent storage or the ICN CS
Why a notification ICT?

- ICT-Notify was designed to push the latest known (usually small) content of an application.

- Can become a primitive for different types of ICN applications:
  - IoT devices, asking to push their latest measurements.
  - GPS-based tracking applications, asking to push their current location.
  - Event-driven applications, asking to notify dynamic events.
  - Traditional applications with event-driven properties.

- Sync can do it, but it is heavyweight.
ICT-Notify

- Push notification is a hard problem in ICN.
- We used long-lived interests with a namespace design that allows us to push small pieces of content within one-way delay latency.
  - We don’t focus on the specifics of the mechanism, but on its role as an ICT.
- The intermediate ICT-Notify service, does not require a repository, but only needs to ”remember” the last pushed data
  - By only looking at its name.
Deploying an Intermediate ICT

- An intermediate ICT should be deployed where connectivity might impact applications.
- When deployed in the network, an intermediate ICT can make the ICT-based applications agnostic to connectivity, even with intermittent links or inconsistent end-to-end path.
- The ICT name gives the context.
  - Without revealing the content!!

The intermediate ICT is deployed by the operator.

The parties don't know about ICT.
Tested Topologies in Open Network Lab (ONL)

- Linux Host
- Linux Software Router (SWR)
### ICT-Sync Demo – File Transfer

<table>
<thead>
<tr>
<th></th>
<th>0% loss</th>
<th>10% loss</th>
<th>20% loss</th>
<th>30% loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 0% loss
- 10% loss
- 20% loss
- 30% loss
ICT-Notify Demo – Location Tracking App

Reliable links

Never a synchronous end-to-end path
Takeaways

• The goal of this paper is to propose architectural modifications to ICN that allow effective decoupling of information and connectivity mechanisms.

• We do this by specifying that:
  – Forwarding strategies should implement only connectivity-oriented mechanism, and should be chosen by network operators.
  – Information-oriented mechanisms should be implemented by Information-Centric Transport (ICT).

• We implemented ICT-Sync and ICT-Notify to address specific name-based semantics.
  – Retrieve all the names of a shared prefix
  – Push the latest content of the shared prefix
Future Work

• How can an ICT be designed to support performance and reliability requirements while keeping the application decoupled from connectivity?
  – mechanisms for performance properties intrinsically couple information-level and connectivity-level considerations.
• Name-based ICT might be too simplistic
  – Explore other discovery mechanisms.
• Where to deploy intermediate services?
  – At every hop? Selected routers? Should all ICTs be deployed at the same places? Or is it related to their mechanisms? We need to quantify the tradeoffs.
• Is the a finite set of ICTs?
• What are the Security implications?
Questions?

• Thank you!