Scalable Point-to-multipoint Communication Information-centric Networking

June, 2015

Based on the 2015 IEEE Consumer Communications & Networking Conference (CCNC) paper Scalable Point-to-multipoint Communication for Cloud Networking using Information-centric Networking

Börje Ohlman
Ericsson Research

Anders Eriksson, Karl-Åke Persson, Adeel Mohammad Malik, Marcus Ihlar and Linus Sunde
Outline

› ICN and NetInf background
› CCN vs. NetInf
› Motivating SPIN
› SPIN details
› Evaluation results
› Conclusion
The SAIL Project (Scalable & Adaptive Internet Solutions) 2010-2013

- EU Call FP7-ICT-2009-5
  - 25 partners
  - 30 months duration
  - 12.4 M€ EU funding in 2.5 years (total ~20M€)

- SAIL’s main objective
  - Design concepts and technologies for the networks of the future
  - Develop techniques to move from today’s to future networks
An information-centric Waist

Applications
application-specific names

Name Layer

ICN name layer for naming Information Objects

API

domain-specific schemes

Transport

Forwarding
Basic generic ICN Router functionality

(A day in the life of an Information Object (IO))

A request for an IO is received

Check if IO is in cache

Check if request for IO already been sent

Decide on which interface(s) to forward request

The IO request is forwarded

<table>
<thead>
<tr>
<th>Cache</th>
<th>Request Aggregator</th>
<th>Request Routing</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Bits</td>
<td>TT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GET I/F in
I/F out
GET I/F in
I/F out
GET I/F in
I/F out
GET I/F in
I/F out
GET I/F in
I/F out
GET I/F in
I/F out
GET I/F in
I/F out
GET I/F in
I/F out
GET I/F in
I/F out
GET I/F in
I/F out
GET I/F in
I/F out
Content Centric Networking (CCN)
CCN

I want /videos/CES
Prefix routing

`/videos/CES`
Caching at each node
(optional)
CCN

Automatic CDN!

"I want that same video"
Mobility!

"I moved. Re-send request"
ni naming: RFC6920

**Authority part**

```
ni://example.org/sha-256;UyaQV-Ev4rdLoHy...
```

**Hash value or label**

```
ni:///sha-256;B_K97zTtFuOhug27fke4_Z...
```

**NDO Structure**

- **Object Name**: `ni://sha-256;B_K97zTtFuOhug27fke4_Z...`
- **Object in Message**:
  - multipart/mixed
    - application/json
      - Object management data
    - multipart/mixed
      - application/steam-meta+xml
        - Application-specific meta data
    - application/binary
      - Actual object bits
  - SHA-256 hash coverage
NetInf Publisher mobility

NetInf components making publisher mobility possible:

- Uses flat ni names according to RFC6920
- Has a Name Resolution Service (NRS) that provides one level of indirection form mapping between identifiers and locators
NetInf components making publisher mobility possible:

› Uses flat ni names according to RFC6920

› Has a Name Resolution Service (NRS) that provides one level of indirection form mapping between identifiers and locators
**Name Resolution Service (NRS) - Alternatives**

**Name based routing**

**Broadcast Resolution**

**Source**
- Application
- NetInf
- Cache

**Router**
- Application
- NetInf
- NBR
- Cache

**Requester**
- Application
- NetInf
- Cache

**NetInf**
- NRS
- Cache

**NRS Server**

**Transport**

**SCALABLE & ADAPTIVE INTERNET SOLUTIONS**
CCN vs. NetInf

- CCN/NDN (Content Centric Networking/Named Data Networking)
  - Pros
    - Major approach
  - Cons
    - Lack support for publisher mobility
    - Can not find off path cashes
    - IPR
    - Divergence between CCN and NDN communities
    - New routing infrastructure needed
    - Security solution requires trusted third party

- NetInf (Networking of Information)
  - Pros
    - Designed for mobility support
    - Pub/Sub support
    - Migration friendly http convergence layer specified which can run as overlay on legacy IP network (no new routing needed)
    - No known IPR issues
  - Cons
    - Minor approach
Why ICN when we have CDN?

A recent example that today’s IP and CDN technology is broken…
Why SPIN?

Scalable Point-to-Multipoint Mechanism for Information-centric Networks (SPIN)

› Most existing ICN approaches require a new global ICN routing mechanism to route based on the names of information objects.
  – A new global name-based routing system raises concerns regarding migration and scalability.

› In contrast SPIN:
  – avoids name-based routing or IP multicast routing by making use of legacy IP unicast routing combined with DNS
  – supports two types of service models
    › publish-subscribe
    › request aggregation
  – support client and server mobility
    › DNS interacts with an additional Name Resolution System (NRS) which resolves the name of an information object to a current locator of a representation of the information object.
Point-to-multipoint as a Network Service

WHAT
› Scalable point-to-multipoint distribution of pre-recorded content or real-time data for e.g. IoT or live video
› Application-independent
› Scales to very large numbers of clients

HOW
› Based on legacy IP routing
› New name resolution system as a complement to DNS
› Small modifications of CoAP, or a new NetInf protocol
› Introduction of packet interception, pub-sub, and cache functions in edge routers

WHY
› Allows operators to add value, since the service is integrated in edge routers and cannot be built as an OTT service
Publish-Subscribe

Subscription State

Subscriber Client

Request Subscribe

Response Notification 1

Response Notification N

Cancel

Publisher Server
Pub-Sub & Event Notification Service

- Point-to-multipoint distribution of notifications to subscribers in near real-time
- Application-independent
- Can be used for video distribution or IoT
- Scales to millions of subscribers
- Allows operators to add value, since this is not an OTT service
- Can be built with small modifications of existing protocols (HTTP and Coap) and with legacy IP routing

Diagram:
- Publish binding: FQDN / label -> IP address
- Resolve: FQDN / label -> IP address
- Name format: FQDN / label
- Event notification
- Subscribe request
- Publication
- Subscribe request acknowledgement

NERS (NetInf Edge Routers)

DNS
NRS
Signalling sequence for set-up of a point-to-multi-point tree

1: Req \((A_{C1}, A_s, t_1)\)  
2: Req \((A_{21}, A_s, t_3)\)  
3: Req \((A_{11}, A_s, t_4)\)  
4: Resp \((A_s, A_{11}, t_4)\)  
5: Resp \((A_{12}, A_{21}, t_3)\)  
6a: Resp \((A_{22}, A_{C1}, t_1)\)  
6b: Resp \((A_{23}, A_{C2}, t_2)\)  
7: Resp \((A_{22}, A_{C1}, t_1)\)  
8: Resp \((A_{23}, A_{C2}, t_2)\)

**Symbols:**  
\(A_n\) \(\rightarrow\) IP address  
\(C\) \(\rightarrow\) Client  
\(IO_x\) \(\rightarrow\) Information Object x  
\(R\) \(\rightarrow\) Information-centric Router  
\(S\) \(\rightarrow\) Server  
\(R1\) \(\rightarrow\) Request  
\(R2\) \(\rightarrow\) Response
Combining connectionless and Connection Oriented Signalling
Token Matching

1: \( \text{Req}(A_C, A_S, t_1) \)  
2: \( \text{Req}(A_1, A_S, t_2) \)  
5: \( \text{Resp}(A_2, A_1, t_2) \)  
6: \( \text{Resp}(A_1, A_C, t_1) \)

**Int-1**  
Address \( A_1 \)

- **Receive and store**  
  1: \( \text{Req}(A_C, A_S, t_1) \)

- **Construct, store, bind to 1, and send**  
  2: \( \text{Req}(A_1, A_S, t_2) \)

- **Receive**  
  5: \( \text{Resp}(A_2, A_1, t_2) \)

- **Match**  
  2: \( \text{Req}(A_1, A_S, t_2) \) with  
  1: \( \text{Req}(A_C, A_S, t_1) \)

- **Match**  
  5: \( \text{Resp}(A_2, A_1, t_2) \) with  
  2: \( \text{Req}(A_1, A_S, t_2) \)

- **Match**  
  2: \( \text{Req}(A_1, A_S, t_2) \) with  
  1: \( \text{Req}(A_C, A_S, t_1) \)

- **Construct and send**  
  6: \( \text{Resp}(A_1, A_C, t_1) \)
NetInf NODE Protocol Stack Using Legacy IP Routing

NetInf Node Functions

- pub/sub
- event notification
- request aggregation
- caching
- name-based routing

NetInf / COAP API

IP
UDP
NetInf/CoAP
NetInf / CoAP

intercept

IP
UDP

IP Network

Sub
Sub
Sub
Sub

A

Publisher

DNS
NRS

NetInf functions

NRS Name Resolution System
Example Use Case prototyped and Evaluated

› Live Video Streaming
  – We show that a NetInf implementation of SPIN in request aggregation mode can handle flash crowds very well, can scale to unlimited number of users
  – Request aggregation is best suited for use cases where each IO is (normally) only requested once and the client can (roughly) predict when it will become available

› Sensor networking
  – With the pub/sub model the sensor can publish new readings as they become available, no need to respond to periodic requests. It can even go offline in-between publications
  – Publish/Subscribe is best used when the publications are time series and/or published infrequently at unpredictable times (e.g. fire alarms) and/or when the publisher is only intermittently online
NetInf/SPIN Request Aggregation

Streaming node

NetInf router
Cache

NetInf router
Cache

NetInf router
Cache

NetInf Access network

NetInf network

Request

Response
CPU utilization as a function of the number of clients measured on routers R1, R2 and R3.
Cache hit limit
Traditional HTTP caching vs. request aggregation

- Lower cache hit limit with traditional HTTP proxy
- Lower cache hit limit with request aggregation proxy

Round trip time between R1 and Publisher (ms)

Measured values with traditional http proxy
Measured values with request aggregation proxy
Cache hit region with traditional http proxy
Cache hit region with request aggregation proxy
CPU load in ICN routers R1 and R2 as a function of the number of subscribers
Conclusions and Future Work

› Key take a ways:
  – SPIN support both request aggregation and publish/subscribe
  – SPIN can support flash crowds without network provisioning, something not possible in today’s IP networks
  – Use IP unicast routing both for forwarding requests and for retrieving objects

› Future Work
  – Access control using Attribute Based Encryption (ABE)
  – Cut-through forwarding