MAP-Me
Managing Anchor-less Producer Mobility in Content-Centric Networks

Jordan Augé, Software Engineer, Cisco Systems (France)
joint work with G. Carofiglio, G. Grassi, L. Muscariello, M. Papalini, G. Pau, X. Zeng
ICNRG interim meeting, March 18, 2018
Mobility challenges

Mobility challenges beyond ICN

- Core overloaded with signalization and user traffic
- Opportunity to remove anchors, tunnels... simplify & unify
- Much interest in IETF (DMM/ILA/...) and 3GPP

Interesting ICN features from a mobility standpoint

Much progress in producer mobility...

... and some original solutions wrt IP
Consumer vs. Producer mobility

- Consumer mobility natively supported in virtue of the pull-based model
- Producer Mobility = still an open challenge

We are only looking at L3 here, transport need to support / leverage ICN specifics, and help with seamless handovers

Towards native mobility

- Preserve features of ICN (caching, multipath, content-based security) : no tunnelling, no name rewriting
- L2-agnostic
- Decentralized : remove the need for anchors
- Feasible and scalable : lightweight in terms of network / CPU usage
- High bandwidth / low latency requirements
- Preserve performance of user flows in progress
Illustration of related approaches in ICN

- **RDV-BASED**
  - [DNS]
  - [SCOM, SNAMP, ...]

- **ANCHOR-BASED**
  - [Mobile IP]
  - [MobiCCN, ...]

- **TRACE-BASED**
  - [LBMA, KITE, ...]

- **ANCHOR-LESS**
  - [Interest Forwarding, ...]
MAP-Me in a nutshell

- Use of **data plane protocols** for sending forwarding updates to the network
- Producer **sends mobility updates to itself**
- Modifies the forwarding graph by patching a small set of routers’ FIB entries (next hops)

- → **Update protocol** with good path stretch properties for fast global reconnnectivity: routing can optimize paths later...

+ **Notification protocol**: Optimizations for low-latency traffic
MAP-Me implementation overview

1. Mobility detection & prefix management

2. Packet formats

3. Processing

4. Temporary state
1. Mobility detection & prefix management

**Mobility detection**

A **FaceManager** module listens to **L2/L3 events** and dynamically create/delete faces.

Example on Linux:

![Diagram]

**Prefix management**

Hook the FIB to learn about **locally served prefixes**
2. Packet formats

Interest (I.)

**Name** | **I.** | **Seq**
--- | --- | ---

I. Update

Prefix | IU | Seq | Sec. T.

I. Update Ack

Prefix | IU_{ack} | Seq | Sec. T.

I. Notification

Prefix | IN | Seq | Sec. T.

I. Notification Ack

Prefix | IN_{ack} | Seq | Sec. T.
3. Processing: Update protocol

Modify FIB entries at forwarding timescale to min. disconnectivity assumption

- existence of a routing protocol populating FIBs
- multipath support (not shown here for clarity)

key ideas

- interest-based signaling triggered by producer after movement
- reuse stale FIB entries to forward signalization
- flip edges in existing forwarding tree to forward to new location
Update propagation

Hop-by-hop retransmissions for reliability
Concurrent updates
3. Processing : Notification protocol
4. Transient FIB (TFIB)

TFIB

```
<table>
<thead>
<tr>
<th>/prefix</th>
<th>[ F_m ]</th>
<th>sequence number</th>
<th>retx timers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td></td>
<td>{ F_k → τ_m }</td>
</tr>
</tbody>
</table>
```

TFIB

nexthop faces sequence number retx timers
Implementation summary

1. Face Manager FIB inspection

2. IU / IN and ack
   - IU
   - IU Ack
   - IN
   - IN Ack

3. IU/IN processing
   - Scoped discovery

4. TFIB
   - Sequence number
   - Reliability (ReTX)
MAP-Me timescales

FAST MOBILITY
- Notifications: Near-zero disconnectivity
  - Update 1 node

e.g. 5s

ROAMING
- Updates: Fast forwarding plane restoration
  - Update 1 path

e.g. 1m

MOBILE RELOCATION
- Routing: Recomputation of optimal paths
  - Update whole graph

improves?
Interaction between MAP-Me and routing

FIB update overwritten by routing protocol?

*Initial proposal*: independent protocols
- a routing instance running on producer
- seq → (Rseq, seq)
- delay MAP-Me IU/IN until local routing convergence
- reissue IU/IN after routing update has been received

Detailed example with link state in paper

Towards a joint protocol?
MAP-Me properties [see paper]

FOR UPDATES

• Optimal reconnection algorithm; **bounded stretch properties**
• Update Algorithm **correctness and convergence** have been proved
• Preserves the **loop-freeness** of the forwarding tree
• Works in presence of multipath

FOR NOTIFICATIONS

• Near-zero disconnectivity
• Ability to use various heuristics for instance to bound stretch / load of neighbouring links, etc.
Security

- A lightweight **prefix attestation scheme** based on **hash-chains**
- Appropriate for Trace-based and Anchor-less mobility approaches
- Preserves MAP-Me benefits in terms of user / network performance
- Lightweight in term of CPU
- Additional security wrt typical HMAC based approaches in a distributed context

Evaluation
# Evaluation methodology

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Simple analytical model under RWP: MapMe &gt; AB wrt. <em>overhead</em> and <em>offloading</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph simulations</td>
<td>Synthetic / Rocketfuel topologies : <em>stretch</em></td>
</tr>
<tr>
<td>Simulation</td>
<td>Synthetic &amp; Trace-based - <em>user &amp; network performance</em></td>
</tr>
<tr>
<td>Emulation</td>
<td>Demo &amp; testbed evaluations (Grid5k → Cisco)</td>
</tr>
</tbody>
</table>
ndnSIM simulation setup

- Realistic 802.11n radio access (handover, rate adaptation, fading, etc.)
- N permanent CBR pairs
- Random waypoint mobility (uniformly dist. and varying const. speed) & vehicular trace
- Baseline scenario:
  - C = 10Mb/s – 5ms delay
  - N = 1..5..20
  - speed: up to 50km/h
Simulations

**User performance**
- Average packet loss and delay
  *(flow. performance)*
- Layer-3 handover delay
  *(e2e availability)*
- Average hop count
  *(path stretch)*

**Network performance**
- Average number of signalization packets per handover
  *(signaling overhead)*
- Average number of processed messages per router per handover
  *(distributedness)*
- Per-class average link utilization
  *(offloading)*
Simulations

User performance
- Average packet loss and delay (flow performance)
- Layer-3 handover delay (e2e availability)
- Average hop count (path stretch)

Network performance
- Average number of signalization packets per handover (signaling overhead)
- Average number of processed messages per router per handover (distributedness)
- Per-class average link utilization (offloading)
Behaviour of mobility protocols

Always connected

Some path have longer stretch...

Time to reach Anchor

Hop update

Impact of routing!
Avg. stretch and latency on synthetic topologies

[Bar graph showing stretch and latency for different topologies and algorithms, with labels for each topology and algorithm represented by different colors.]

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Confidential
Conclusion

**MapMe**: effective anchorless producer mobility solution for ICN

- improvements over state-of-the-art solutions
- simple, easily deployable

Future work includes:

- WiFi / LTE handover - Interdomain
- Multi-source / Multihoming
- More experimentation (see next presentation)

MAP-Me implementation & deployment
CICN, NDN, Hybrid ICN (hICN)

Jordan Augé, Software Engineer, Cisco Systems (France)
joint work with G. Carofiglio, L. Muscariello, M. Papalini, M. Sardara
ICN RG interim meeting, March 18, 2018
NDN prototype

- Original C++ / boost implementation used in paper
  - Compatible both with NFD and NDNSIM
- Link to paper and NFD source code: [http://mapme-tnsm.github.io/](http://mapme-tnsm.github.io/)
- Not maintained

- Bonus: implementations of alternative solutions:
  - Anchor-based (generic)
  - Trace-based (inspired from KITE)
« Making WiFi mobile » demo @ MWC’17

DEMONSTRATED SCENARIOS
1. CONSUMER MOBILITY
2. ASYNC. MULTICAST
3. NATIVE MULTIHOMING
4. PRODUCER MOBILITY & LIVE VIDEO STREAMING
CICN (hICN)

• Continuation of CCNx’s Metis forwarder
  • C / libparc (data structures: hashmaps, etc.)
• Most codebase identical between CICN & hICN
  • Differences at lower layers (packet formats, etc)

• Very fresh, will be released in fd.io shortly
  • No notifications yet
  • No security
Upon attachment to $P_1$, producer sens an IU that should update forwarding state in both hICN and IP routers on the way to $P_0$.

hICN principles of reusing RFC compliant packets.

In hICN, we can use an non-reactive ICMP(v6) redirect packet to $P_0$’s address using the link local address as default gateway.

Sequence number can be encoded in payload, and security can be ensured as usual by AH header, SEND, etc.
ICMP redirect processing

In an hICN router

- ICMP-RD packets recognized and used to update FIB entries
- Seq. check, multipath possible

In an IP router

- IPv6 routers can be configured to use ND proxy that will update local route cache (with higher administrative distance), rewrite the gateway with own LL address, and forward it.
- no multipath
- no local check on seqno., deferred to closest hICN router
Experimentation with vICN

Conclusion

• Two ICN implementations
  • ICN (opensource) and hICN
• We have just finished a draft for MAP-Me
  • working on hICN version
  • We welcome community discussion and feedback
• MAP-Me ~ a « mobility control plane » for (Hybrid) ICN
  • Interesting anchorless properties are generally applicable
• Interest beyond the ICN community?

Link to paper and NFD source code: [http://mapme-tnsm.github.io/](http://mapme-tnsm.github.io/)