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 meering, 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

e socket library + forwarder mechanisms as in:

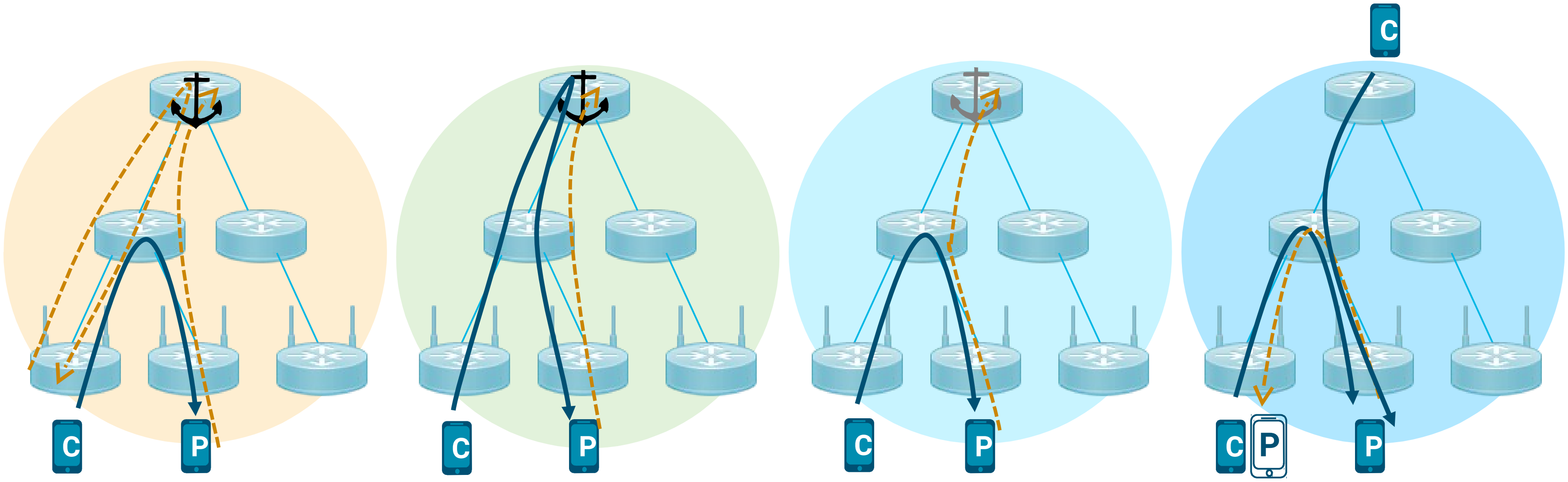
G. Carofiglio, L. Muscariello, M. Papalini, N. Rozhnova, X. Zeng, "**Leveraging ICN In-network Control for Loss Detection and Recovery in Wireless Mobile networks**", *In. Proc. ICN'16, Sep. 26-28, 2016, Kyoto (JP)*

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, …]

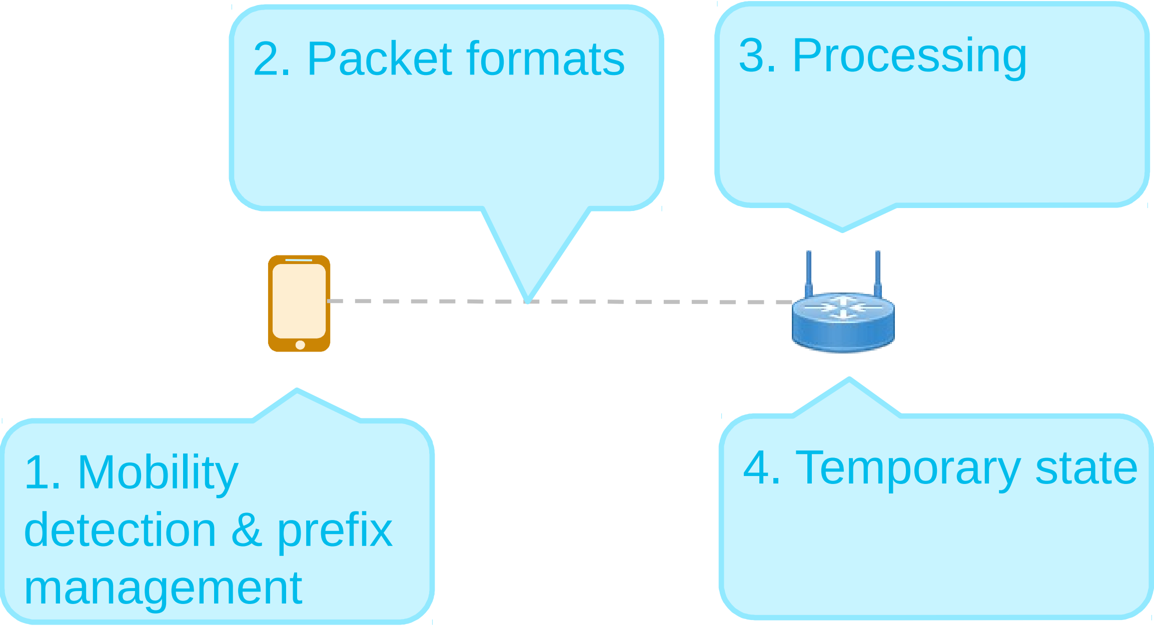
MAP-Me in a nutshell

- 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 reconnectivity : routing can optimize paths later…

  + Notification protocol : Optimizations for low-latency traffic

MAP-Me implementation overview

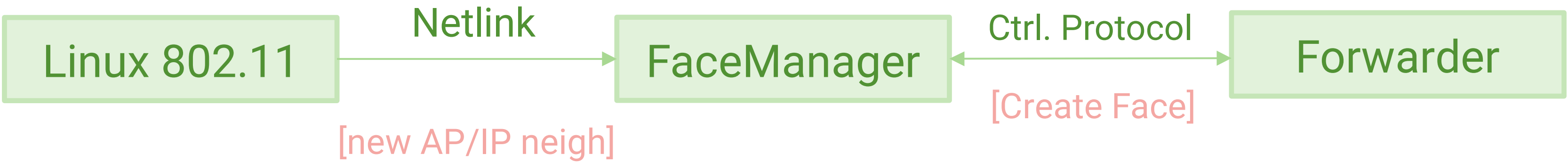


1. Mobility detection & prefix management

Mobility detection

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

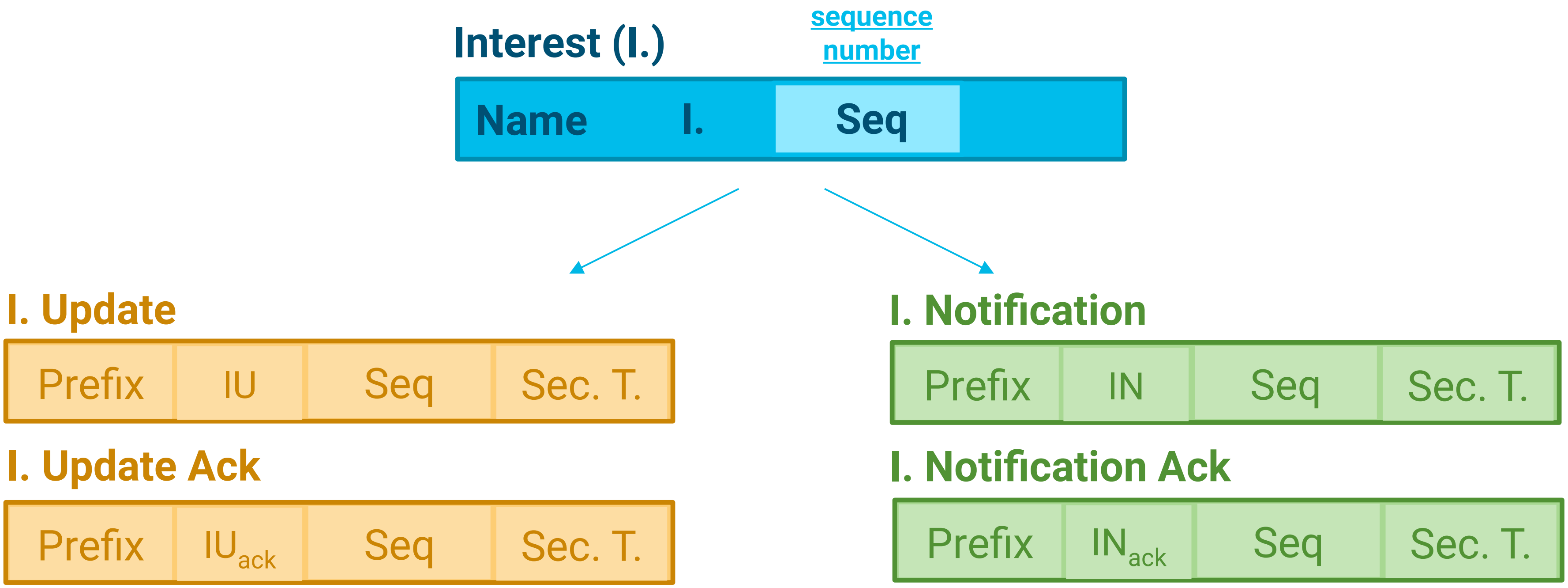
Example on Linux:



Prefix management

Hook the FIB to learn about **locally served prefixes**

2. Packet formats



3. Processing: Update protocol

Modify FIB entries at forwarding timescale to min. disconnectivity

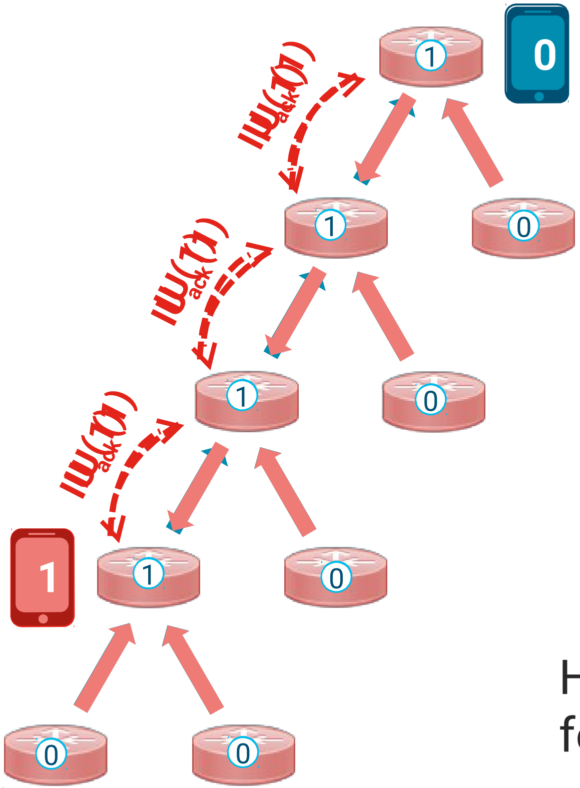
assumption

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

key ideas

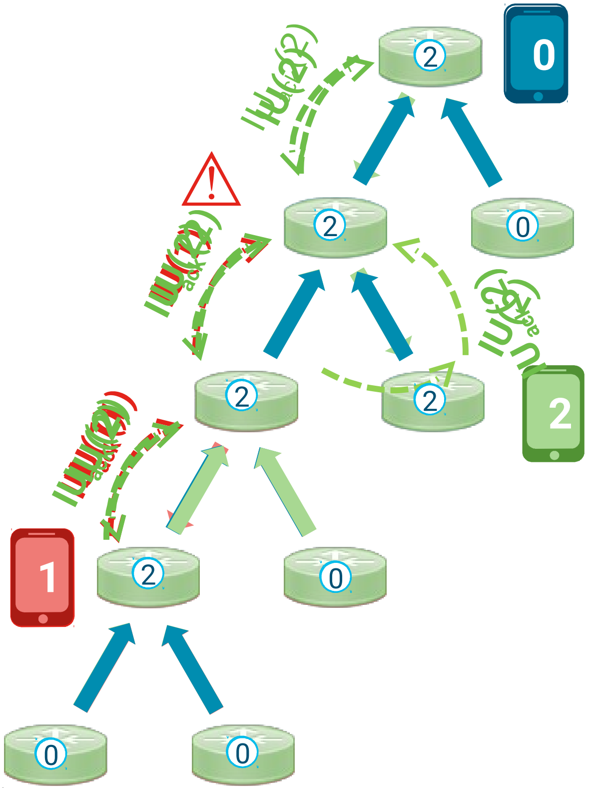
- interest-based signaling **triggered by producer** after movement
- reuse **stale FIB entries** to forward signalization
- **flip edges** in existing forwarding tree to forward to new location

Update propagation

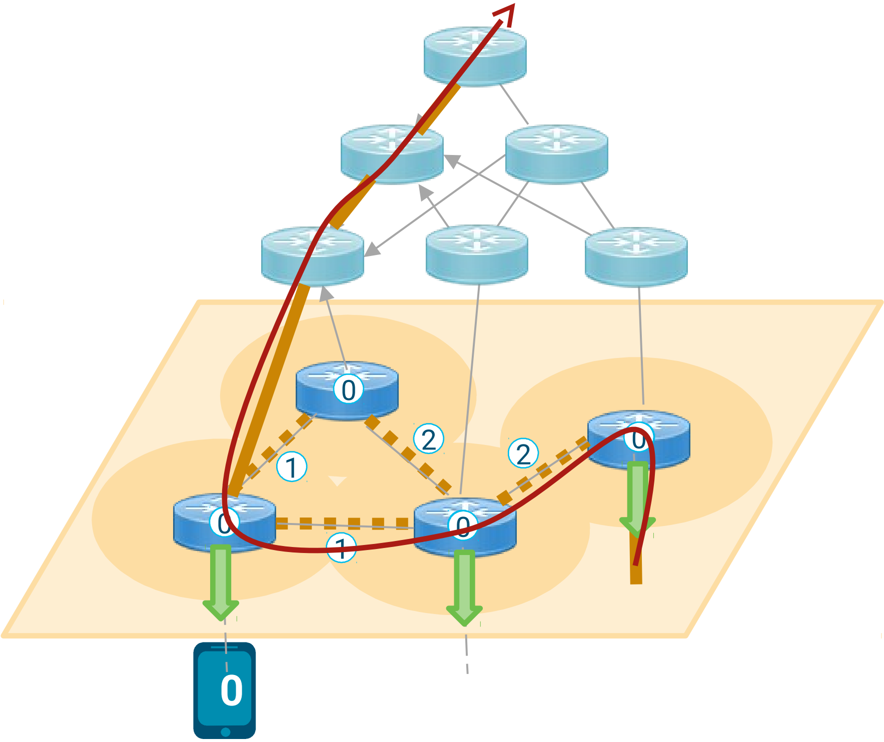


Hop-by-hop retransmissions for reliability

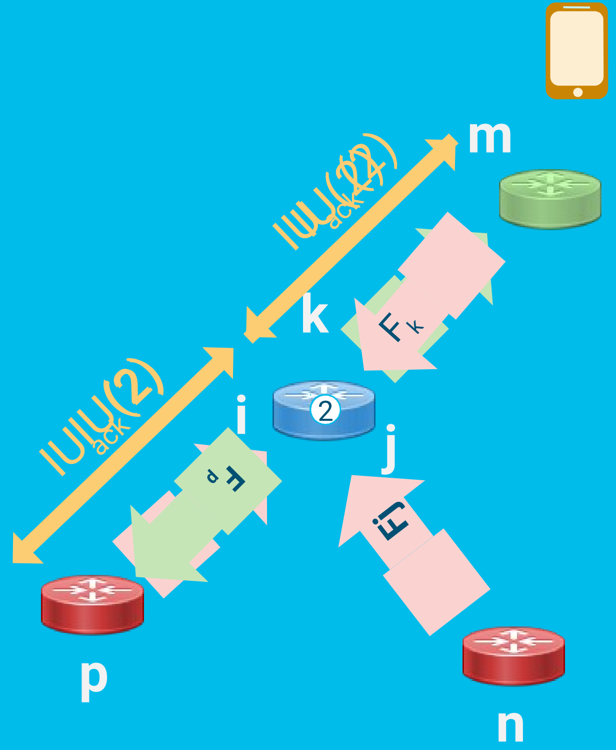
Concurrent updates



3. Processing : Notification protocol



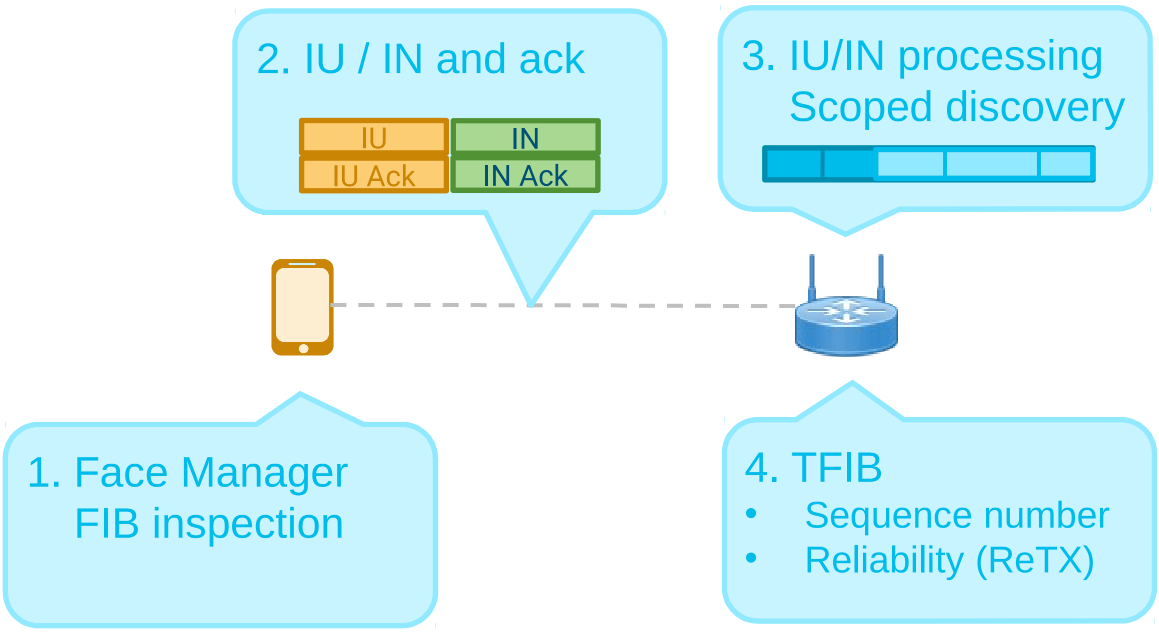
4. Transient FIB (TFIB)



	nexthop faces	sequence number	retx timers
/prefix	[F_m]	2	{ $F_k \rightarrow \tau_m$ }

TFIB

Implementation summary



MAP-Me timescales

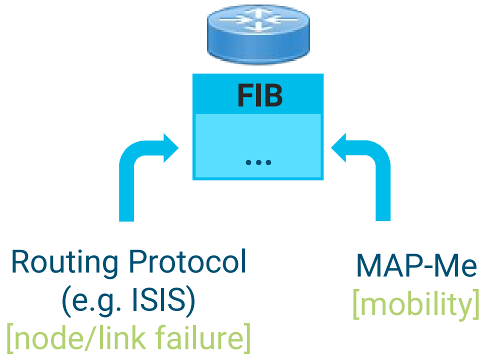
FAST MOBILITY | e.g. 5s | ROAMING | e.g. 1m | MOBILE RELOCATION

Notifications — T_U — Updates — T_R — Routing

improves ?

Near-zero disconnectivity	Fast forwarding plane restoration	Recomputation of optimal paths
Update **1 node**	Update **1 path**	Update **whole graph**

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 reconnectivity 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**

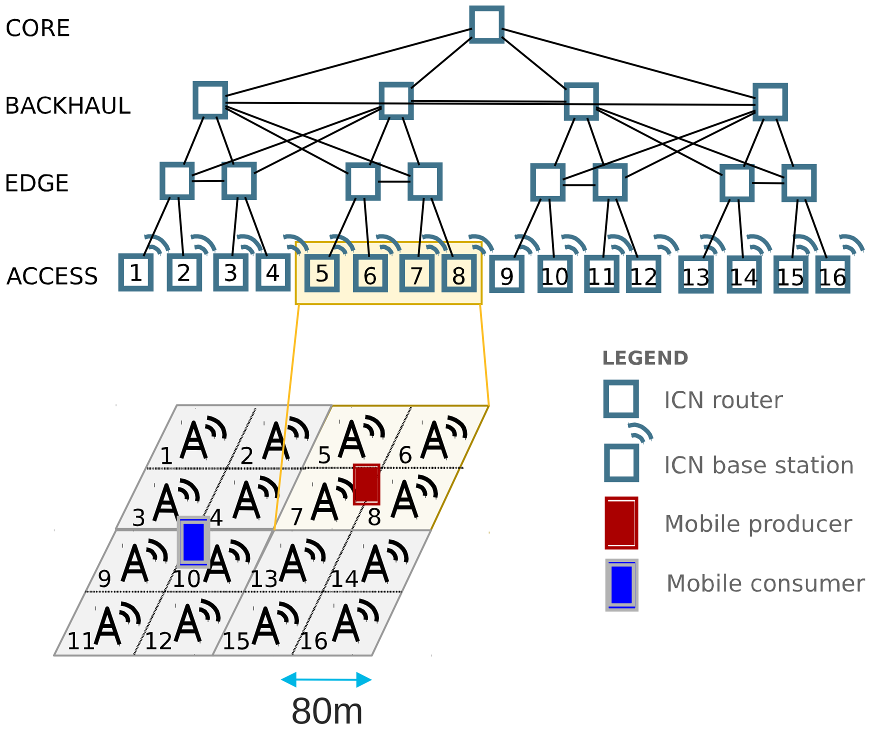
A. Compagno, X. Zeng, L. Muscariello, G. Carofiglio, J. Augé, "**Secure Producer Mobility in Information Centric Networking**", *in. Proc. ICN'17, Sep. 26-28, 2017, Berlin (DE)*

Evaluation

Evaluation methodology

Analysis	Simple analytical model under RWP: MapMe > AB wrt. *overhead* and *offloading*
Graph simulations	Synthetic / Rocketfuel topologies : *stretch*
Simulation	Synthetic & Trace-based - *user & network performance*
Emulation	Demo & testbed evaluations (Grid5k → Cisco)

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 **50**km/h

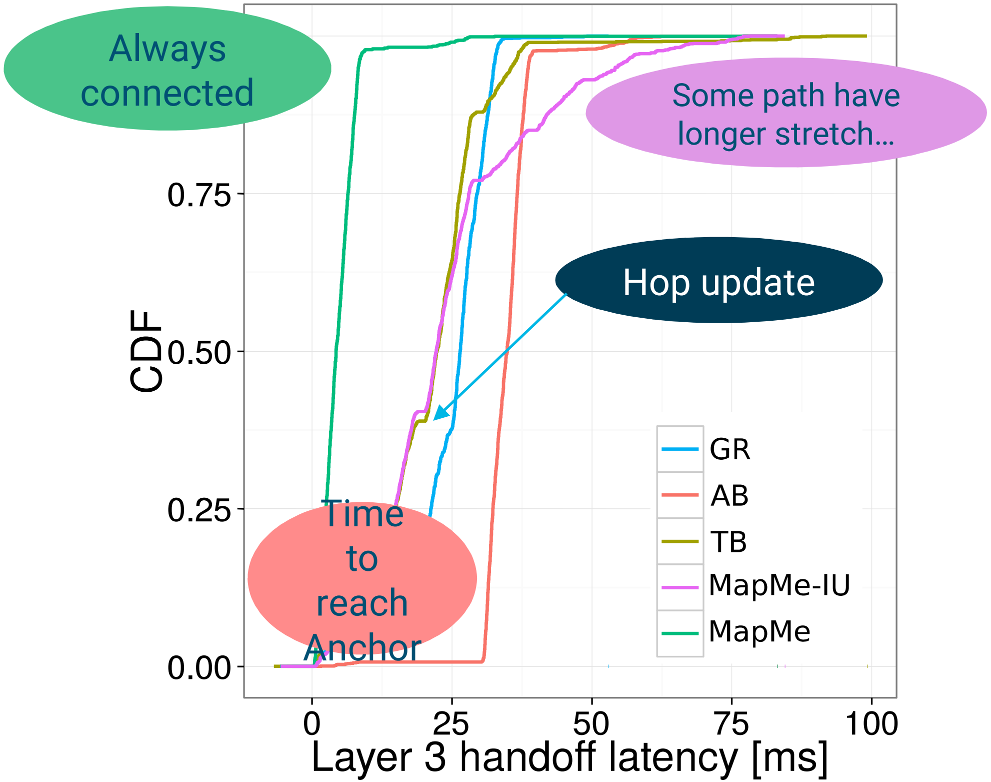
Simulations

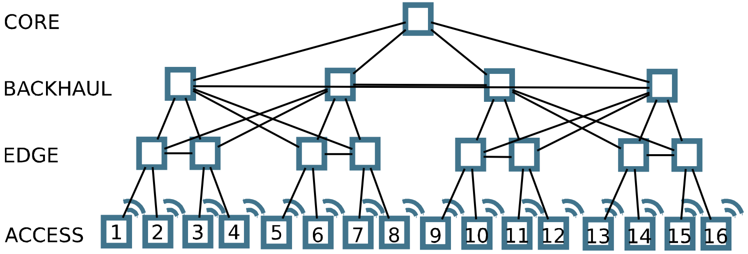
User performance	Average packet loss and delay *(flow. performance)*	Layer-3 handover delay *(e2e availability)*	Average hop count *(path stretch)*
Network performance	Average number of signalization packets per handover *(signaling overhead)*	Average number of processed messages per router per handover *(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)*

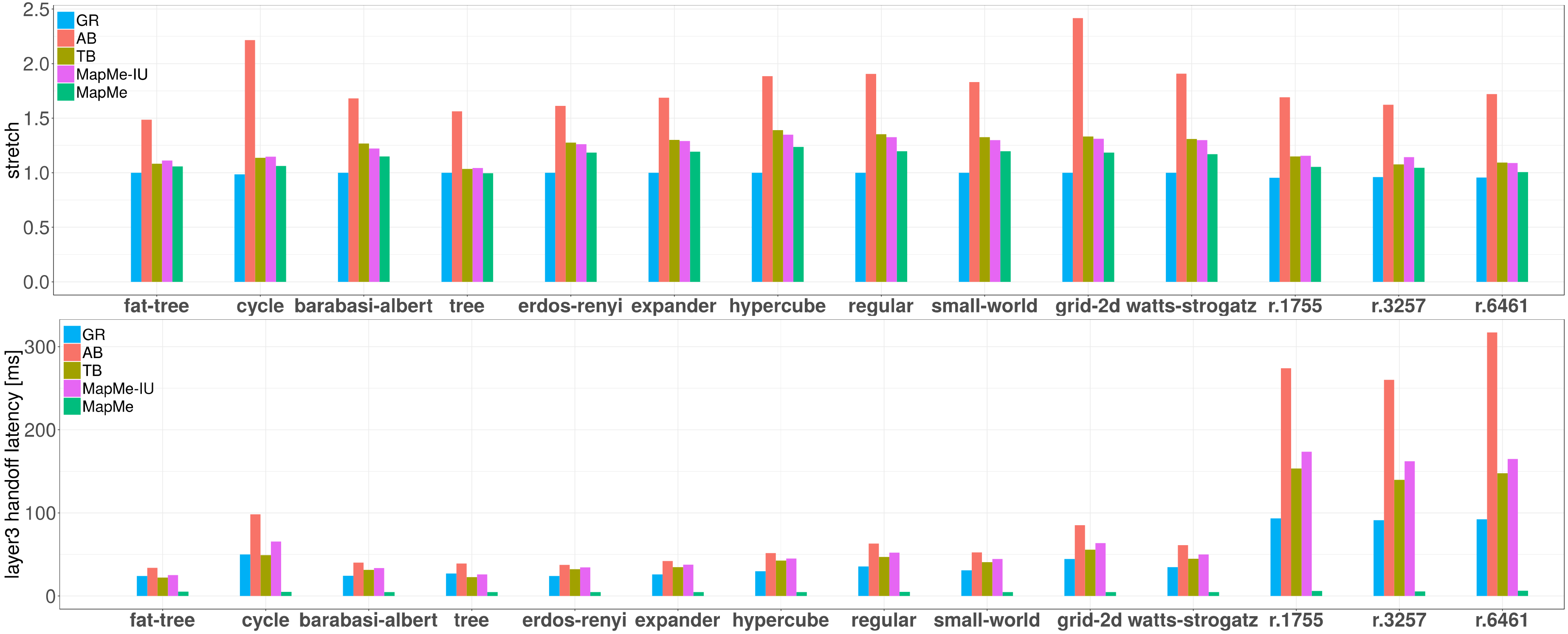
Behaviour of mobility protocols





Impact of routing !

Avg. stretch and latency on synthetic topologies



Conclusion


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

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)

J.Augé, G.Carofiglio, G.Grassi, L.Muscariello, G.Pau and X.Zeng, **MAP-Me: Managing Anchor-less Producer Mobility in Content-Centric Networks**, in IEEE TNSM, vol. PP, no. 99.

MAP-Me implementation & deployment

CICN, NDN, Hybrid ICN (hICN)

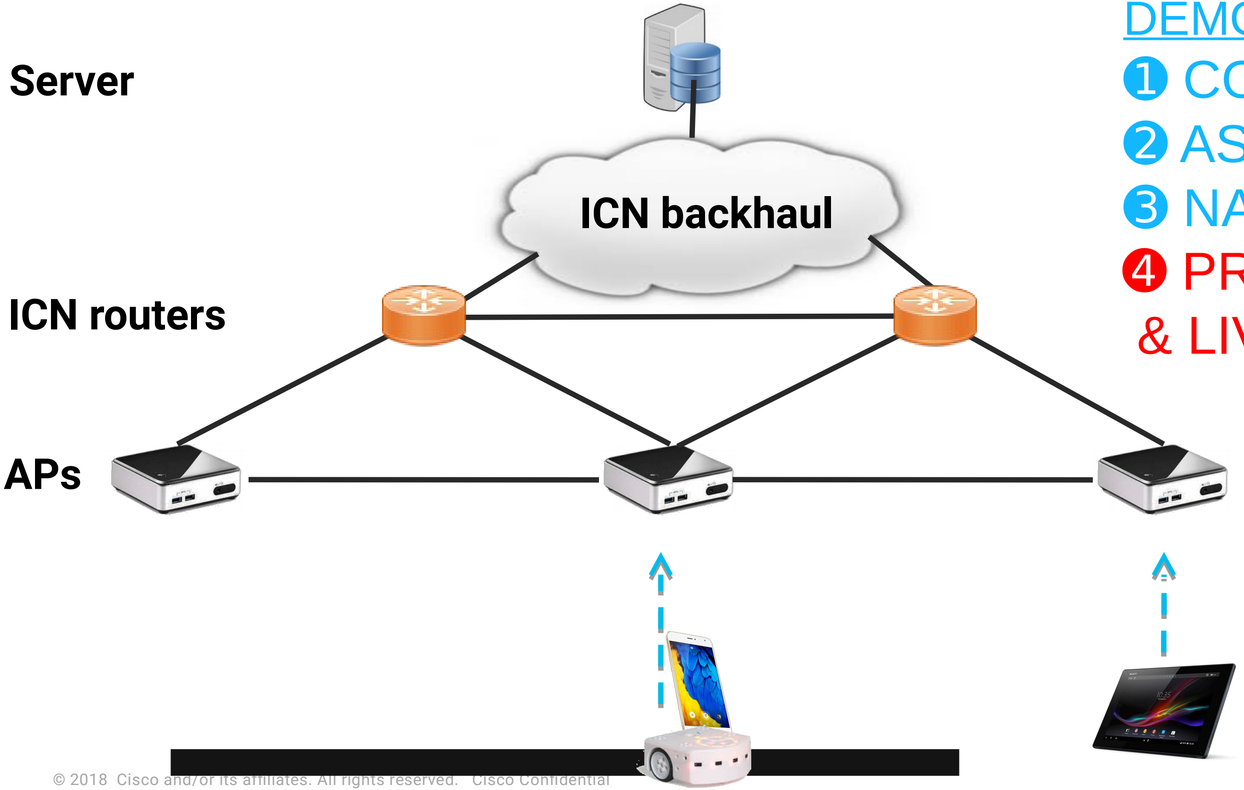
Jordan Augé, Software Engineer, Cisco Systems (France)
joint work with G. Carofiglio, L. Muscariello, M. Papalini, M. Sardara
ICNRG interim meering, March 18, 2018

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/
- Not maintained

- Bonus : implementations of alternative solutions:
  - Anchor-based (generic)
  - Trace-based (inspired from KITE)

« Making WiFi mobile » demo @ MWC'17



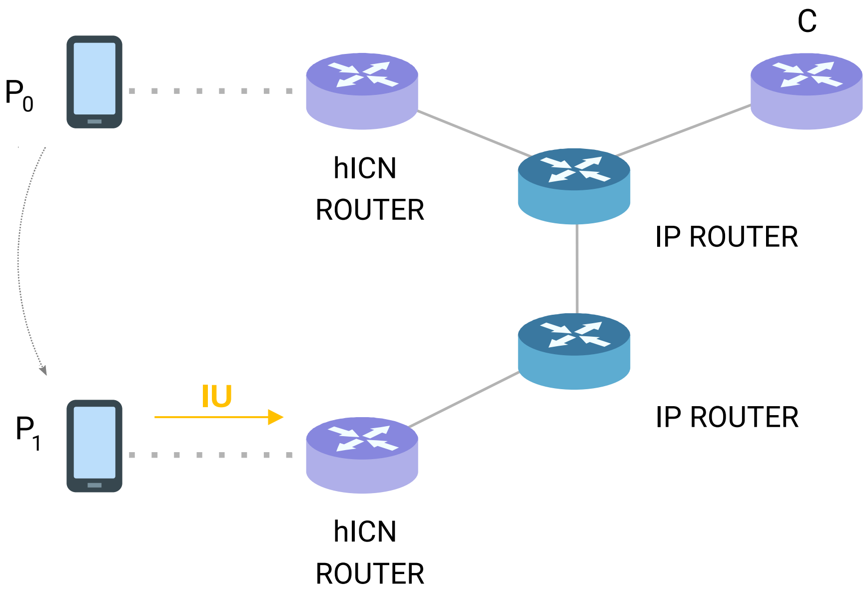
DEMONSTRATED SCENARII

❶ CONSUMER MOBILITY
❷ ASYNC. MULTICAST
❸ NATIVE MULTIHOMING
❹ 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

MAP-Me in Hybrid ICN



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

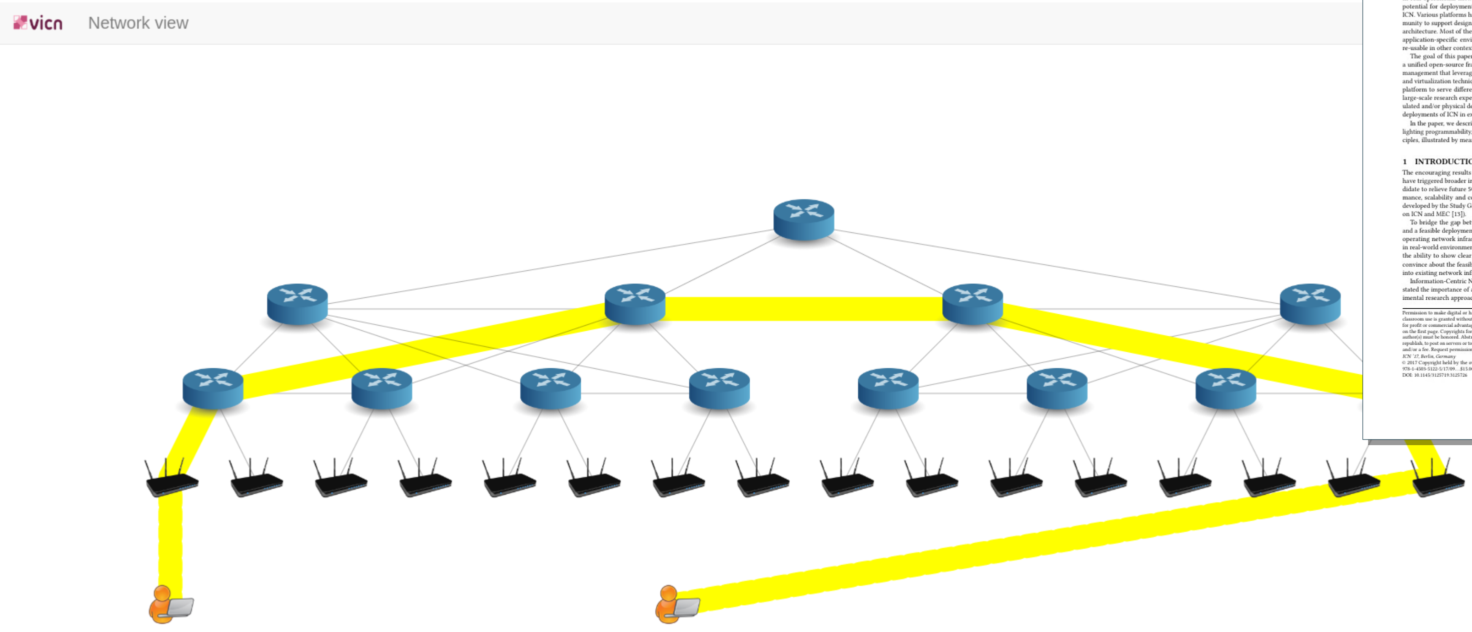
In an hICN router

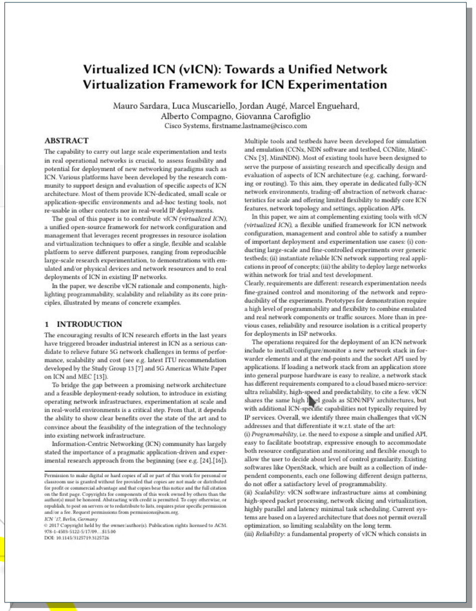
- ICMP-RD packets recognized and used to update FIB entries
- Seq. check, multipath possible

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





Virtualized ICN (vICN): Towards a Unified Network Virtualization Framework for ICN Experimentation

Mauro Sardara, Luca Muscariello, Jordan Augé, Marcel Enguehard, Alberto Compagno, Giovanna Carofiglio
Cisco Systems, firstname.lastname@cisco.com

ABSTRACT

The capability to carry out large scale experimentation and tests in real operational networks is crucial, to assess feasibility and potential for deployment of new networking paradigms such as ICN. Various platforms have been developed by the research community to support design and evaluation of specific aspects of ICN architecture. Most of them provide ICN-dedicated, small scale or application-specific environments and ad-hoc testing tools, not re-usable in other contexts nor in real-world IP deployments.

The goal of this paper is to contribute *vICN (virtualized ICN)*, a unified open-source framework for network configuration and management that leverages recent progresses in resource isolation and virtualization techniques to offer a single, flexible and scalable platform to serve different purposes, ranging from reproducible large-scale research experimentation, to demonstrations with emulated and/or physical devices and network resources and to real deployments of ICN in existing IP networks.

In the paper, we describe vICN rationale and components, highlighting programmability, scalability and reliability as its core principles, illustrated by means of concrete examples.

1 INTRODUCTION

The encouraging results of ICN research efforts in the last years have triggered broader industrial interest in ICN as a serious candidate to relieve future 5G network challenges in terms of performance, scalability and cost (see e.g. latest ITU recommendation developed by the Study Group 13 [7] and 5G Americas White Paper on ICN and MEC [13]).

To bridge the gap between a promising network architecture and a feasible deployment-ready solution, to introduce in existing operating network infrastructures, experimentation at scale and in real-world environments is a critical step. From that, it depends the ability to show clear benefits over the state of the art and to convince about the feasibility of the integration of the technology into existing network infrastructure.

Information-Centric Networking (ICN) community has largely stated the importance of a pragmatic application-driven and experimental research approach from the beginning (see e.g. [24],[16]).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ICN '17, Berlin, Germany
© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM.
978-1-4503-5122-5/17/09...$15.00
DOI: 10.1145/3125719.3125726

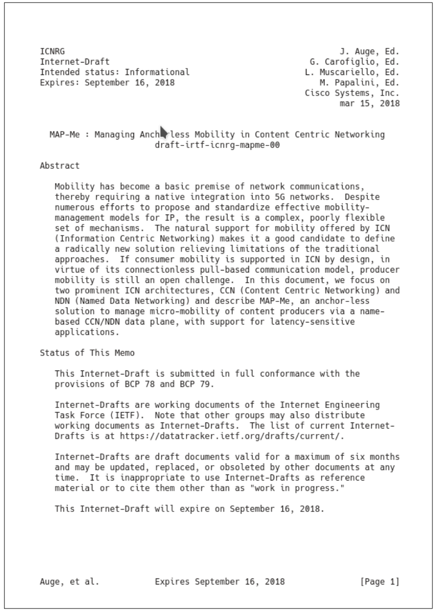
Multiple tools and testbeds have been developed for simulation and emulation (CCNx, NDN software and testbed, CCNlite, MiniCCNx [3], MiniNDN). Most of existing tools have been designed to serve the purpose of assisting research and specifically design and evaluation of aspects of ICN architecture (e.g. caching, forwarding or routing). To this aim, they operate in dedicated fully-ICN network environments, trading-off abstraction of network characteristics for scale and offering limited flexibility to modify core ICN features, network topology and settings, application APIs.

In this paper, we aim at complementing existing tools with *vICN (virtualized ICN)*, a flexible unified framework for ICN network configuration, management and control able to satisfy a number of important deployment and experimentation use cases: (i) conducting large-scale and fine-controlled experiments over generic testbeds; (ii) instantiate reliable ICN network supporting real applications in proof of concepts; (iii) the ability to deploy large networks within network for trial and test development.
Clearly, requirements are different: research experimentation needs fine-grained control and monitoring of the network and reproducibility of the experiments. Prototypes for demonstration require a high level of programmability and flexibility to combine emulated and real network components or traffic sources. More than in previous cases, reliability and resource isolation is a critical property for deployments in ISP networks.

The operations required for the deployment of an ICN network include to install/configure/monitor a new network stack in forwarder elements and at the end-points and the socket API used by applications. If loading a network stack from an application store into general purpose hardware is easy to realize, a network stack has different requirements compared to a cloud based micro-service: ultra reliability, high-speed and predictability, to cite a few. vICN shares the same high level goals as SDN/NFV architectures, but with additional ICN-specific capabilities not typically required by IP services. Overall, we identify three main challenges that vICN addresses and that differentiate it w.r.t. state of the art:
(i) *Programmability*, i.e. the need to expose a simple and unified API, easy to facilitate bootstrap, expressive enough to accommodate both resource configuration and monitoring and flexible enough to allow the user to decide about level of control granularity. Existing softwares like OpenStack, which are built as a collection of independent components, each one following different design patterns, do not offer a satisfactory level of programmability.
(ii) *Scalability*: vICN software infrastructure aims at combining high-speed packet processing, network slicing and virtualization, highly parallel and latency minimal task scheduling. Current systems are based on a layered architecture that does not permit overall optimization, so limiting scalability on the long term.
(iii) *Reliability*: a fundamental property of vICN which consists in

J.Augé, G.Carofiglio, G.Grassi, L.Muscariello, G.Pau and X.Zeng, **MAP-Me: Managing Anchor-less Producer Mobility in Content-Centric Networks**, in IEEE TNSM, vol. PP, no. 99, pp. 1-1.

Conclusion


ICNRG J. Auge, Ed.
Internet-Draft G. Carofiglio, Ed.
Intended status: Informational L. Muscariello, Ed.
Expires: September 16, 2018 M. Papalini, Ed.
Cisco Systems, Inc.
mar 15, 2018

MAP-Me : Managing Anchorless Mobility in Content Centric Networking
draft-irtf-icnrg-mapme-00

Abstract

Mobility has become a basic premise of network communications, thereby requiring a native integration into 5G networks. Despite numerous efforts to propose and standardize effective mobility-management models for IP, the result is a complex, poorly flexible set of mechanisms. The natural support for mobility offered by ICN (Information Centric Networking) makes it a good candidate to define a radically new solution relieving limitations of the traditional approaches. If consumer mobility is supported in ICN by design, in virtue of its connectionless pull-based communication model, producer mobility is still an open challenge. In this document, we focus on two prominent ICN architectures, CCN (Content Centric Networking) and NDN (Named Data Networking) and describe MAP-Me, an anchor-less solution to manage micro-mobility of content producers via a name-based CCN/NDN data plane, with support for latency-sensitive applications.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 16, 2018.

Auge, et al. Expires September 16, 2018 [Page 1]

- 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 ?

CICN project and CICN source code: http://wiki.fd.io/view/Cicn
Link to paper and NFD source code : http://mapme-tnsm.github.io/