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

Jordan Auge, Giovanna Carofiglio, Giulio Grassi, Luca Muscariello, Giovanni Pau, Xuan Zeng

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

Jordan Augé, Giovanna Carofiglio, Giulio Grassi,
Luca Muscariello, Giovanni Pau, and Xuan Zeng

***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 work, we look at two prominent ICN architectures, CCN (Content Centric Networking) and NDN (Named Data Networking) and we propose *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 streaming applications. We analyze *MAP-Me* performance and provide guarantees of correctness, stability, and bounded stretch, which we verify on real ISP topologies. Finally, we set up a comprehensive simulation environment in NDNSim 2.1 for *MAP-Me* evaluation and comparison against the existing classes of solutions, including a realistic trace-driven car-mobility pattern under a 802.11n radio access. The results are encouraging and highlight the superiority of *MAP-Me* in terms of user performance and network cost metrics. All the code is available as open-source.

***Index Terms*—Information-Centric Networking(ICN), micro-mobility, producer mobility; anchor-less.**

I. Introduction

With the phenomenal spread of portable user devices, mobility has become a basic requirement for almost any communication network as well as a compelling feature to integrate in the next generation networks (5G). The need for a mobility-management paradigm to apply within IP networks has striven a lot of efforts in research and standardization bodies (IETF, 3GPP among others), all resulting in a complex access-dependent set of mechanisms implemented via a dedicated control infrastructure. The complexity and lack of flexibility of such approaches (e.g. Mobile IP) calls for a radically new solution dismantling traditional assumptions like tunneling and anchoring of all mobile communications into the network core. This is particularly important with the increase in rates and mobile nodes (IoT), a vast amount of which never moves.

The Information Centric Network (ICN) paradigm brings native support for mobility, security, and storage within the network architecture, hence emerging as a promising 5G technology candidate. Specifically on mobility management, ICN has the potential to relieve limitations of the existing approaches by leveraging its primary feature, the redefinition of packet forwarding based on *names* rather than on *network addresses*. We believe that removing the dependence on location identifiers is a first step in the direction of removing the need for any anchoring of communications into fixed network nodes, which may considerably simplify and improve mobility management. Within the ICN paradigm, several architectures have been proposed, as reported in [40], [4]. Among those different approaches, large attention from the research community has been focused on CCN [16] and on one of its evolution, NDN [42], which we consider in this work.

As a direct result of CCN/NDN design principles, consumer mobility is natively supported: a change in physical location for the consumer does not translate into a change in the data plane like for IP. The retransmission of requests for data not yet received by the consumers takes place without involving any signaling to the network. Producer mobility and realtime group communications present more challenges, depending on the frequency of movements, latency requirements, and content lifetime. The topology does not reflect the naming structure, and we have to preserve key functionalities such as multipath, caching, etc. In all cases, beyond providing connectivity guarantees, additional transport-level mechanisms might be required to protect the flow performance, which are beyond the scope of this paper (see [6] for instance).

Tackling such problems, in a simple and effective way by exploiting CCN/NDN key characteristics is at the core of this paper. Previous attempts have been made in CCN/NDN (and ICN in general) literature to go beyond the traditional IP approaches, by using the existing CCN/NDN request/data packet structures to trace producer movements and to dynamically build a reverse-forwarding path (see [43] for a survey). They still rely on a stable home address to inform about producer movements (e.g. [44]) or on buffering of incoming requests at the producer's previous point of attachment – PoA – (e.g. [18]), which prevents support for latency-sensitive streaming applications. We focus on this class of applications (e.g. live streaming or videoconferencing) as they have the most stringent performance requirements: negligible per-packet loss-rate and delays. In addition, they typically originate from a single producer and don't allow for the use of caching. Good performance for other classes such as adaptive or elastic flows is simpler to guarantee as they only require a high-enough average throughput [30], [32].

In this paper, we aim to take one step forward in the defini-

J. Augé, G. Carofiglio and L. Muscariello are with Cisco Systems
G. Grassi is with LIP6/UPMC
G. Pau is with LIP6/UPMC and UCLA
X. Zeng is with IRT SystemX and LIP6/UPMC

tion of a name-based mechanism operating in the forwarding plane and completely removing any anchoring, while aiming at latency minimization.

The main contribution of this work is a proposal for an anchor-less mobility-management mechanism, named *MAP-Me*, with the following characteristics:

- *MAP-Me* addresses micro (e.g. intra Autonomous Systems) producer mobility. Addressing macro-mobility is a non-goal of this paper, left for future work. We are focusing here on complementary mechanisms able to provide a fast and lightweight handover, preserving the performance of flows in progress.
- *MAP-Me* does not rely on global routing updates, which would be too slow and too costly, but rather works at a faster timescale propagating forwarding updates and leveraging real-time notifications left as breadcrumbs by the producer to enable live tracking of its position[1]. The objective being the support of high-speed mobility and real-time group applications like Periscope [28]. *MAP-Me* leverages core CCN/NDN features like stateful forwarding, dynamic and distributed Interest load balancing to update the forwarding state at routers, and relaying former and current producer locations.
- *MAP-Me* is designed to be access-agnostic, to cope with highly heterogeneous wireless access and multi-homed/mobile users.
- Finally, low overhead in terms of signaling, additional state at routers, and computational complexity are also targeted in the design to provide a solution able to scale to large and dynamic mobile networks.

To evaluate this proposal, we first contribute an analysis of protocol correctness and guarantees; then, we provide a comprehensive and realistic simulation environment in NDNSim 2.1 [25], where we compare it against an ideal Global Routing, which can instantly and optimally update all FIBs, anchor-based and tracing-based solutions over a set of random and trace-driven mobility patterns representing V2I scenarios based on 802.11 radio access. Results show that *MAP-Me* satisfies its objectives while equalling or outperforming existing alternatives both in terms of user performance (e.g. loss, delays) and network cost (e.g. signaling overhead, link utilization) metrics. We also show wide applicability of results across different topologies radio and mobility patterns.

The rest of the paper is organized as follows: Sec. II reviews the different classes of mobility-management approaches for CCN/NDN, and discusses their pros and cons. We introduce the design principles of *MAP-Me* in Sec. III, and detail its operations in Sec. IV, before analyzing its correctness and path-stretch guarantees in Sec. V. A comprehensive evaluation of the benefits of our anchor-less proposal is then performed in Sec. VI. Finally, Sec. VII investigates the interaction and possible cooperation between *MAP-Me* and an existing routing protocol, before concluding the paper in Sec. VIII.

We have released all the source code developed for this paper as opensource [1]. Additional results, as well as more details about the implementation of proposals is available in a technical report [5].

II. Related Work

Many efforts have been made to define mobility-management models for IP networks in the last two decades, resulting in a variety of complex, often not implemented, proposals. A good survey of these approaches is RFC 6301 [45]. Likewise, within the ICN family, different approaches to mobility–management have been presented [35]. In DONA [20] mobile publishers unregister and re–register their information at each handoff to the hierarchy of resolution handlers. Such an update process, however, may incur in a non-negligible messaging overhead to eliminate stale registration across the network [40]. Similarly, NetInf [3] and JUNO [33] report network mobility events to a resolution service, which may incur in network load in case of high mobility [34]. PURSUIT [10] instead uses a rendezvous system to handle network mobility, which requires notification to the topology manager at each handoff and, in some cases, the re-computation of the forwarding identifier used to compute the path to the producer, affecting the handoff delay [40], [34]. Finally MobilityFirst [31] uses a global name resolution service (GNRS), which is updated when a node changes point of attachment. When facing high-frequency mobility, those so-called Resolution-Based (RB) approaches present a similar trade-off: for every packet the consumer has to resolve the producer's location or use stale information and run the risk to reach an old position, incurring in timeout, or Nack, etc.

Specifically for the CCN/NDN solutions, several surveys of mobility-management approaches can be found [43], [9]. In [43] for instance, the authors distinguish three categories of solutions – routing, mapping, and tracing-based – depending on the type of indirection point (also called Rendez-Vous, RV). We build on such classification and extend it to distinguish a fifth class of approaches not relying upon the existence of any anchor point as the RV (Anchor-less approaches):

a) **Routing-based (RT)** solutions rely on intra-domain routing, and require updating all routing in the AS after a mobile's movement. Scalability of these solutions is widely recognized as a concern which explains why they are usually ruled out, in particular for CCN/NDN where the name space is even larger than IP. In the rest of the paper, we will use an idealized routing plane instantly updating all routers as a reference, called Global Routing (**GR**).

b) **Resolution-based (RB)** solutions rely on dedicated RV nodes (similar to DNS) which map content names into routable location identifiers. To maintain this mapping updated, the producer signals every movement to the RV node [13], [17], [24], [18], [2], [19]. Once the resolution is performed, packets can be correctly routed from the consumer along the shortest path, with unitary path stretch (defined as the ratio between the realized path length over the shortest path one). Requiring explicit resolution, together with a strict separation of names and locators, RB solutions involve a scalable CCN/NDN routing infrastructure able to leverage forwarding hints [13], [17]; however, scalability is achieved at the cost of a large hand-off delay as evaluated e.g. in [18], [9] due to RV update and name resolution. To summarize, RB solutions show good

[1] For simplicity, we use the word *producer* in place of the more correct expression *producer name prefixes*

scalability properties and low stretch in terms of consumer to producer routing path, but result to be unsuitable for frequent mobility and for reactive rerouting of latency-sensitive traffic, which are key objective of *MAP-Me*.

c) **Anchor-based (AB)** proposals are inspired by Mobile IP, and maintain a mapping at network-layer by using a stable home address advertised by a RV node, or anchor. This acts as a relay, forwarding through tunneling both interests to the producer, and data packets coming back. For instance in [22] the producer changes its prefix after each movement and then sends an update message to its anchor to notify it of the change. In such context, anchor's placement is critical for the performance of the approach. MobiCCN [38] uses distributed anchors and selects the closest in a hyperbolic space.

Advantages of this approach are that the consumer does not need to be aware of producer mobility and that it has low signaling overhead because only the anchor has to be updated. It however inherits the drawbacks of Mobile IP – e.g. triangular routing and single point of failure – and others more specific to the CCN/NDN context: potential degradation of caching efficiency, bad integrity verification due to the renaming of content during movement. It also hinders multipath capabilities and limits the robustness to failure and congestion initially offered by the architecture. In contrast, *MAP-Me* maintains names intact and avoid single point–of–passage of the traffic.

d) **Tracing-based (TB)** solutions allow the mobile node to create a hop-by-hop forwarding reverse path from its RV back to itself by propagating and keeping alive traces stored by all involved routers. Forwarding to the new location is enabled without tunneling. Like AB though, this approach assumes that the data is published under a stable RV prefix. Kite [44] introduced this approach and proposed storing traces in the PIT to build a breadcrumb trail which could be followed by crossing consumer interests and thus provide a shortcut towards the producer. While it exploits CCN/NDN data plane features without requiring a separate control infrastructure, Kite involves a large signaling due to keep-alive messages to maintain active traces stored in PITs. The idea of creating a reverse path to a stable home router is also expressed in [12], where the authors propose a similar tracing-based approach, leveraging updates in FIB, rather than in PIT, and sending updates to both RV and previous PoA.

e) **Anchor-less (AL)** approaches allow the mobile nodes to advertise their mobility to the network without requiring any specific node to act as a RV. They are less common and introduced in CCN/NDN to enhance the reactivity with respect to AB solutions by leveraging CCN/NDN name-based routing. [29] exploits multicast and directs the same Interest to the nearby PoAs of the producer. In [41] and in the *Interest Forwarding* scheme proposed in [18], the mobile producer sends a notification to its current PoA before moving. The PoA starts buffering incoming Interests for the mobile producer until a forwarding update is completed and a new route is built to reach the current location of the producer. Enhancement of such solutions considers handover prediction. Besides the potentially improved delay performance w.r.t. other categories of approaches, some drawbacks can be recognized: buffering of Interests may lead to timeouts for latency-sensitive applications and handover prediction is hard to perform in many cases. In contrast *MAP-Me* reacts after the handoff, without requiring handover prediction, and avoids Interests buffering but introduces network notification and discovery mechanism to reduce the handoff latency. [27] instead introduces proxy nodes at the edge of 3G/4G architectures and uses tunnels to forward Interest from the former PoA to the current edge. The solution, however, is specific to cellular network.

In addition, there is a class of cache-based orthogonal mechanism that can be combined with the aforementioned solutions to enhance performance for both consumer and producer mobility: the "proactive-caching" class. On consumer side, [36], [39] propose to pre-fetch contents at selected nodes before handover occurs to reduce handover delays, while on producer side the data provider pro-actively pushes predicted contents to the network when handover is imminent [8], [23]. Contents are then served by caches when producer is disconnected. Such approaches leverage ICN's in-network caching to keep high content availability regardless of mobility and can be integrated with *MAP-Me* or other solutions to reduce delays. However, such mechanisms can be insufficient for certain realtime applications (e.g. video call) where contents are generated online and not available in advance for pushing.

Finally, in-network caching and name-based routing techniques also enable a routing-to-replica approach abstracting consumers from producer movements (referred to as data depot in [43]). However, such approach is not suitable for realtime applications or targeted to unpopular content, which may be replaced in cache due to memory limitations. A study of the advantages for popular items can be found in [17].

III. Design

In this section, we introduce *MAP-Me*, a micro-mobility management architecture for CCN/NDN networks. Based on the classification and discussion made in the previous section, we detail here the design principles inspiring *MAP-Me*.

MAP-Me is an **anchor-less, name-based, layer-2 agnostic approach operating at forwarding plane** with the following additional characteristics:

- ***Transparent:*** *MAP-Me* does not involve any name nor modifications to basic request/reply operations to be compatible with standard CCN/NDN design and to avoid issues caused by name modifications like triangular routing, caching degradation, or security vulnerabilities.
- ***Distributed:*** *MAP-Me* is designed to be fully distributed, to enhance robustness w.r.t. centralized mobility management proposals subject to single point-of-passage problem.
- ***Localised:*** *MAP-Me* updates affect the minimum number of routers at the edge of the network to restore connectivity. The goal is to realize effective traffic off-load close to the end-users.
- ***Lightweight:*** *MAP-Me* mobility updates are issued at prefix granularity, rather than content or chunk/packet granularity, to minimize signaling overhead and temporary state kept by in-network nodes.
- ***Reactive:*** *MAP-Me* works at forwarding layer to enable updates in FIBs at network latency, i.e. round-trip time scale. Specific mechanisms are defined, referred to as network notifications and discovery, to maximise reactivity in mobility

management in case of real-time producer tracking and of latency-sensitive communications.

- ***Robust*** to network conditions (e.g. routing failure, wireless or congestion losses, and delays), by leveraging hop-by-hop retransmissions of mobility updates.

A. *MAP-Me* description

As a data plane protocol, *MAP-Me* handles producer mobility events by means of dynamic FIB updates with the objective of minimizing unreachability of the producer. It relies on the existence of a routing protocol responsible for creating/updating the FIB of all routers, possibly with multipath routes, and for managing network failures (eg. [37], [14]). *MAP-Me* is composed of:

- an Update protocol (*MAP-Me*-IU) (Sec.III-B), which is the central component of our proposal;

- a Notification/Discovery protocol (Sec.III-C), to be coupled with the Update protocol (the full approach is referred to as *MAP-Me*) to enhance reactivity in mobility management for realtime/latency-sensitive application.

In this section, we describe the protocols independently of the routing protocol, and dedicate Sec. VII at their integration. Routing will be in charge of handing the long-term impact of mobility (node relocation), and reoptimizing paths. We assume a single producer of content, and no prefix reaggregation at intermediary ISPs.

B. *MAP-Me* Update protocol

1) Rationale: The rationale behind *MAP-Me*-IU is that the producer announces its movements to the network by sending a special Interest Packet, named *Interest Update* (IU) to "itself" after it reattaches to the network. Such a message looks like a regular Interest packet named with the prefix advertised by the producer. As such, it is forwarded according to the information stored in the FIBs of traversed routers towards previous locations of the producer known by router FIBs. A special flag carried in the header of the IU enables all routers on the path to identify the Interest as a mobility update and to process it accordingly to update their FIBs (a detailed description of the IU processing is provided in Sec.IV-B).

The key aspect of the proposal is that it removes the need for a stable home address (present in Tracing-Based approaches for instance) by directly leveraging name-based forwarding state created by CCN/NDN routing protocols or left by previous mobility updates. FIB updates are triggered by the reception of mobility updates in a fully distributed way and allow a modification on-the-fly to point to the latest known location of the producer.

2) Updates propagation: *MAP-Me*-IU aims at quickly restoring global reachability of mobile prefixes with low signaling overhead, while introducing a bounded maximum path stretch (i.e. ratio between the selected and the shortest path in terms of hops). Let us illustrate its behavior through the example in Fig. 1, where a single producer serving prefix p moves from position P_0 to P_1 and so on. **Fig. 1(a)** shows the tree formed by the forwarding paths to the name prefix p where IU initiated by the producer propagates.

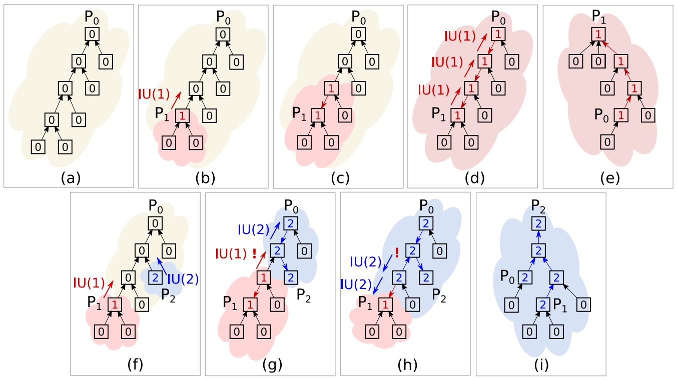


Fig. 1: *MAP-Me*-IU illustration.

Network FIBs are assumed to be populated with routes toward P_0 by a name-based routing protocol (yellow cloud) After the relocation of the producer from P_0 to P_1, once the layer-2 attachment is completed, the producer issues an IU carrying the prefix p and this is forwarded by the network toward P_0 (in general, toward one of its previous locations according to the FIB state of the traversed routers).

Fig. 1(b) shows the propagation of the IU. As the IU progresses, FIBs at intermediate hops are updated with the ingress face of the IU (Fig 1(c, d)). IU propagation stops when the IU reaches P_0 and there is no next hop to forward it. The result is that the original tree rooted in P_0 becomes re-rooted in P_1 (Fig. 1(e)). Looking at the different connected regions, we see that IU propagation and consequent FIB updates have the effect of extending the newly connected subtree (represented as a red cloud) : at every step, an additional router and its predecessors are included in the connected subtree. We will analyze the properties of the update propagation process in terms of bounded lenght and stretch in Sec.V.

3) Concurrent updates: Frequent mobility of the producer may lead to the propagation of concurrent updates. To prevent inconsistencies in FIB updates, *MAP-Me*-IU maintains a sequence number at the producer end that increases at each handover and identifies every IU packet. Network routers also keep track of such sequence number in FIB to verify IU freshness. Without detailing the specific operations in *MAP-Me* to guarantee update consistency (whose description is provided in Sec.IV-B), we can say that modification of FIB entries is only triggered when the received IU carries a higher sequence number than the one locally stored, while the reception of a less recent update determines a propagation of a more recent update through the not-yet-updated path. An example of reconciliation of concurrent updates is illustrated in Figure 1(f), when the producer has moved successively to P_1 and then to P_2 before the first update is completed.

Both updates propagate concurrently until the update with sequence number 1 (IU_1) crosses a router that has been updated with fresher information – that has received IU with higher sequence number (IU_2) as in Fig. 1(g). In this case, the router stops the propagation of IU_1 and sends back along its path a new IU with an updated sequence number (Fig. 1(h)). The update proceeds until ultimately the whole network has converged towards P_2 (Fig. 1(i)).

MAP-Me-IU protocol reacts at a faster timescale than routing – allowing more frequent and numerous mobility events

– and over a localized portion of the network edge between current and previous producer locations. We thus expect *MAP-Me*-IU respectively to minimize disconnectivity time and to reduce the link load, which are the main factors affecting user flow performance, as later shown by our evaluation in Section VI.

C. Map-Me Notification/Discovery protocol

IU propagation in the data plane accelerates forwarding state re-convergence w.r.t. global routing (GR) or resolution-based (RB) approaches operating at control plane, and w.r.t. anchor-based (AB) approaches requiring traffic tunneling through the anchor. Still, network latency makes IU completion not instantaneous and before an update completes, it may happen that a portion of the traffic is forwarded to the previous PoA and dropped because of the absence of a valid output face leading to the producer.

Previous work in the Anchor-Less category has suggested the buffering of Interests at previous producer location [18] to prevent such losses by increasing network reactivity. However, such a solution is not suitable for applications with stringent latency requirements (e.g. real-time) and may be incompatible with IU completion times. Moreover, the negative effects on latency performance might be further exacerbated by IU losses and consequent retransmissions in case of wireless medium. To alleviate such issues, we introduce two separate enhancements to *MAP-Me*-IU protocol, namely *(i)* an **Interest Notification** mechanism for frequent, yet lightweight, signaling of producer movements to the network and *(ii)* a scoped **Producer Discovery** mechanism for consumer requests to proactively search for the producer's recently visited locations.

1) Interest notification: An Interest Notification (IN) is a breadcrumb left by producers at every encountered PoA. It looks like a normal Interest packet carrying a special identification flag and a sequence number, like IUs. Both IU and IN share the same sequence number (producers indistinctly increase it for every sent message) and follow the same FIB lookup and update processes. However, unlike IU packets, the trace left by INs at the first hop router does not propagate further. It is rather used by the discovery process to route consumer requests to the producer even before an update process is completed.

It is worth observing that updates and notifications serve the same purpose of informing the network of a producer movement. The IU process restores connectivity and as such has higher latency/signaling cost than the IN process, due to message propagation. The IN process provides information to track producer movements before update completion when coupled with a scoped discovery. The combination of both IU and IN allows to control the trade-off between protocol reactivity and stability of forwarding re-convergence.

2) Discovery: The extension of *MAP-Me* with notifications relies on a local discovery phase: when a consumer Interest reaches a PoA with no valid output face in the corresponding entry, the Interest is tagged with a "discovery" flag and labeled with the latest sequence number stored in FIB (to avoid loops). From that point on, it is broadcasted with hop limit equal to one to all neighbors and discarded unless it finds the breadcrumbs left by the producer to track him (notifications). The notifications can either allow to forward consumer Interests directly to the producer or give rise to a repeated broadcast in case of no valid output face. The latter is the case of a breadcrumb left by the producer with no associated forwarding information because the producer has already left that PoA as well. A detailed description of the process is reported in Sec.IV-B.

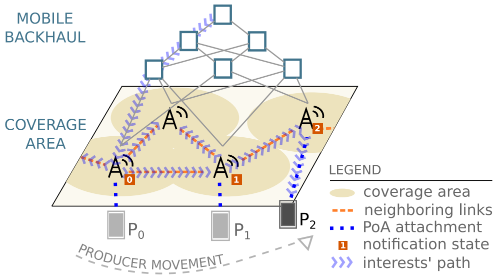


Fig. 2: Notifications/Discovery process example.

As further shown in Sec. VI, the notification/discovery mechanism is important to preserve the performance of flows in progress, especially when latency-sensitive.

D. Full MAP-Me approach

In the rest of the paper, we evaluate a combined update and notification/discovery approach consisting of sending a IN immediately after an attachment and a IU at most every T_U seconds, referred to as *MAP-Me*, to reduce signaling overhead especially in case of high mobility. The update-only proposal, denoted as *MAP-Me-IU*, is also evaluated separately.

Figure 2 illustrates the combined use of notifications and discovery in a mobile access network where the different PoAs form the leaves of a fat tree.

The producer, initially in position P_0 moves to P_1 and later to P_2, sending each time an Interest Notification (respectively IN 1 and IN 2. Consumer interests are forwarded using FIB information synchronized with the initial state of the producer and thus reach the initial PoA, P_0. Once the producer moves and the face is destroyed, no valid next-hop face information can be found into the FIB and consumer Interests reaching P_0 enter in discovery mode: they are tagged with the sequence number 0 found in the FIB, and broadcasted to one-hop neighbors, which may either forward them directly to the producer (this is the case for the current PoA of the producer) or broadcast them one hop further if they have been notified of producer attachment by means of INs, but there is no valid forwarding information. Other network nodes reached by a Discovery Interest just discard the packet since they have no fresher information about the position of the producer. The discovery process is iterated until the producer is reached.

IV. Implementation

A. MAP-Me introduction in a CCN/NDN network

In this section we describe the changes to a regular CCN/NDN architecture required to implement *MAP-Me* and detail the above-described algorithms. This requires to specify a special Interest message, additional temporary information associated to the FIB entry and additional operations to update such entry.

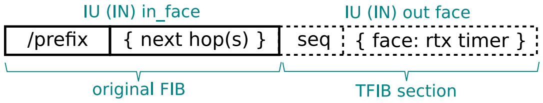


Fig. 3: *MAP-Me* FIB/TFIB description.

1) MAP-Me Messages: Two new optional fields are introduced in a CCN/NDN Interest header:

- a special *Interest Type* (T) to specify four types of messages: Interest Updates (IU), Interest Notifications (IN), as well as their associated acknowledgment (Ack) messages (IU_{ack} and IN_{ack}). Those flags are recognized by the forwarding pipeline to trigger special treatment.
- a *sequence number* to handle concurrent updates and prevent forwarding loops during signaling, and to control discovery interests' propagation;

2) MAP-Me additional Network Information: FIB entries are enriched with a sequence number, initialized to 0 by routing protocol and updated by *MAP-Me* upon reception of IU/IN messages. The Data about not-yet-acknowledged messages are temporarily stored in what we denote as **Temporary FIB buffer, TFIB,** to ensure reliability of the process, and removed upon reception of the corresponding acknowledgement. As sketched in Fig.3, each TFIB entry is composed of an associative array ($F \rightarrow T$) mapping a face F on which IU has been sent with the associated retransmission timer T (possibly null, denoted $\perp$). Note that the update mechanism is a constant delay operation at each router and is performed at line rate.

B. Algorithm description

1) IU/IN transmission at producer: MAP-Me operations are triggered by producer mobility/handover events. At the producer end, a mobility event is followed by a layer-2 attachment and, at network layer, a change in the FIB. More precisely, a new face is created and activated upon attachment to a new PoA. This signal triggers the increase of *MAP-Me* sequence number and the transmission of an IU or IN for every served prefix carrying the updated sequence number.

To ensure reliable delivery of IUs, a timer is setup in the temporary section of the FIB entry (TFIB). If an acknowledgement of the IU/IN reception is not received within τ seconds since the packet transmission, IU is retransmitted.

We define the `SendReliably(`F`, `$type$`, `ϵ`)` function for sending Special Interests of type $type$ on faces F based on FIB entry ϵ. It schedules their retransmission through a timer T stored in TFIB: $\epsilon.\text{TFIB} = \epsilon.\text{TFIB} \cup (F \rightarrow T)$, and removed on Ack.

2) IU/IN processing at network routers: At the reception of IU/IN packets, each router performs a name-based Longest Prefix Match lookup in FIB to compare sequence number from IU/IN and from FIB. According to that comparison:
- if the IU/IN packet carries a higher sequence number, the existing next hops associated to the lower sequence number in FIB are used to forward further the IU (INs are not propagated) and temporarily copied into TFIB to avoid loss of such information before completion of the IU/IN acknowledgement process (in case of IN, such entries in TFIB are set with a $\perp$ timer to maintain a trace of the producer recent attachment). Also, the originating face of the IU/IN is added to FIB to route consumer requests to the latest known location of the producer.
- If the IU/IN packet carries the same sequence number as in the FIB, the originating face of the IU/IN is added to the existing ones in FIB without additional packet processing or propagation. This may occur in presence of multiple forwarding paths.
- If the IU/IN packet carries a lower sequence number than the one in the FIB, FIB entry is not updated as it already stores "fresher information". To advertise the latest update through the path followed by the IU/IN packet, this one is re-sent through the originating face after having updated its sequence number with the value stored in FIB.

3) Hop-by-hop IU/IN acknowledgement: The operations in the forwarding pipeline for IU/IN processing are reported in Algorithm 1.

Algorithm 1: ForwardSpecialInterest(SpecialInterest SI, Ingress face F)

```
CheckValidity()
▹Retrieve the FIB entry associated to the prefix
$\epsilon, T \leftarrow$ FIB.LongestPrefixMatch($SI$.name)
if $SI.seq \geq \epsilon.seq$ then
    ▹Acknowledge reception
    $s \leftarrow \epsilon$.seq
    $\epsilon$.seq $\leftarrow SI$.seq
    SendReliably($F$, $SI$.type + Ack, $\epsilon$)
    ▹Process special interest
    if $F \in \epsilon.TFIB$ then
        ▹Remove outdated TFIB entry (eventually cancelling timer)
        $\epsilon$.TFIB = $\epsilon$.TFIB $\setminus F$
    if $SI.seq > s$ then
        if $SI.type = IU$ then
            ▹Forward the IU following FIB entry
            SendReliably($\epsilon$.NextHops, SI.type, $\epsilon$)
        else
            ▹Create breadcrumb and preserve forwarding structure
            $\epsilon$.TFIB = $\epsilon$.TFIB $\cup \{(f \rightarrow \perp) : \forall f \in \epsilon$.NextHops$\}$
        $\epsilon$.NextHops = $\emptyset$
    $\epsilon$.NextHops = $\epsilon$.NextHops $\cup F$
else
    ▹Send updated IU backwards
    $SI$.seq = $\epsilon$.seq
    SendReliably($F$, $SI$.type, $\epsilon$)
```

4) Face removal at producer/network nodes: Upon producer departures from a PoA, the corresponding face is destroyed. If this leads to the removal of the last next hop, then faces in TFIB with $\perp$ timer (entries generated by notifications) are restored in FIB to preserve the original forwarding tree and thus global connectivity.

5) Consumer request forwarding in case of producer discovery: The forwarding of regular Interests is mostly unaffected in *MAP-Me*, except in the case of discovery Interests that we detail in Algorithm 2. The function `SendToNeighbors(`I`)` is responsible for broadcasting the Interest I to all neighboring PoAs.

When an Interest arrives to a PoA which has no valid next hop for it (because the producer left and the face got destroyed), it enters a discovery phase where the Interest is flagged as a Discovery Interest and with the local sequence number, then broadcasted to neighboring PoAs. Upon reception of a Discovery Interest, the PoA forwards it direcly to the producer if still attached, otherwise it repeats the one-hop

Algorithm 2: InterestForward(Interest I, Origin face F)

```
▷Regular CS and PIT lookup
$\epsilon \leftarrow$ FIB.LongestPrefixMatch($I$.name)
if $\epsilon = \emptyset$ then
    return
if $I.seq = \emptyset$ then
    ▷Regular interest
    if hasValidFace($\epsilon$.NextHops) or DiscoveryDisabled then
        ForwardingStrategy.process($I$, $\epsilon$)
    else
        ▷Enter discovery mode
        $I$.seq $\leftarrow \epsilon$.seq
        SendToNeighbors($I$)
else
    ▷Discovery interest: forward if producer is connected...
    if hasProducerFace($\epsilon$.NextHops) then
        ForwardingStrategy.process($I$, $\epsilon$)
    ▷... otherwise iterate iif higher seq and breadcrumb
    else if $\epsilon.seq \geq I.seq \wedge \exists f |(f \rightarrow \bot) \in \epsilon.TFIB$ then
        $I$.seq $\leftarrow \epsilon$.seq
        SendToNeighbors($I$)
```

brodcast discovery to neighboring PoAs if it stores a recent notification of the producer presence, i.e. an entry in TFIB having higher sequence number than the one in the Discovery Interest. Otherwise, the Discovery Interest is discarded.

It is worth observing that the discovery process is initiated only in the case of no valid next hop, and not every time a notification is found in a router. This is important to guarantee that the notification/discovery process does not affect IU propagation and completion.

C. Security considerations

All mobility management protocols share the same critical need for securing their control messages which have a direct impact on the forwarding of users' traffic. [7] reviews standard approaches from the literature before developing a fast, lightweight and distributed approach that can be applied to *MAP-Me* and fits its design principles.

V. Analysis

In this section, we investigate *MAP-Me* guarantees of forwarding update correctness and path stretch stability and we support them by numerical evaluation over known ISP network topologies. For the sake of clarity, the reported proofs are for single-path routing; extension to multipath is straightforward by replacing trees by DAGs.

A. Correctness and stability of IU mechanism

We consider m consecutive movements of the producer in network positions $\{P_0, P_1, ..., P_m\}$ and focus on forwarding state variations determined by *MAP-Me* at the time instants corresponding to either producer movements or Interest Update processing. At any such instant, as in Fig.1, the network is partitioned into a set of islands, whose number varies in $[1, m + 1]$ as a function of producer movements and hence of the number of ongoing update processes. We assume that at the beginning, global routing builds a spanning tree rooted at first location P_0. The tree can be a minimum SP or a shortest-path tree depending on the routing. About the completion of the update process after a movement k, we can state that:

Proposition 1. *MAP-Me update mechanism guarantees finite completion time of update k, $\forall k \in [1, m]$ in a bounded number of hops equal to $2\left(\max_{0 \leq j<k}(|P_k - P_j| - 1)\right)$;*

Proof. Assuming that IU losses are handled by the retransmission mechanism described in Sec.III, the hop-by-hop propagation of an IU has two possible outcomes: either (i) the next router has a sequence number, which is inferior to the IU carried sequence number; in this case, IU continues its propagation towards the root of the latest routed tree, decreased by 1 hop; or (ii) the router has a higher sequence number, hence the IU is sent back with the encountered higher sequence number towards the originating routed position of the producer. Since the maximum sequence number is bounded by m, the maximum number of hops traversed by IU with sequence number k is finite.

More precisely, the maximum number of hops traversed by IU with sequence number k, IU_k is bounded by twice the maximum distance between the originating router P_k and the farthest previous location P_j, $j < k$ minus one, i.e. $2\left(\max_{0 \leq j<k}(|P_k - P_j| - 1)\right)$. Indeed, the worst case occurs when IU_k encounters a more recent update $k' > k$ at the hop before reaching the latest routed previous location, which can also coincide with the farthest one in terms of distance. In such a case, IU_k propagates back to P_k carrying k' sequence number before stopping. □

After IU_k propagation, the router P_k and all its predecessors traversed by IU_k to reach the last routed location are connected to the island of highest encountered sequence number, and thus the number of distinct islands is reduced by one unit. By iterating the same process on all IUs, it is straightforward to see that at IU_m completion $m+1$ islands associated to sequence number $0, 1, ..., m-1$ will merge into the island created by IU_m. Regarding the properties of an island, we can state that:.

Proposition 2. *Given a sequence of m consecutive movements of producer position on the routing tree rooted in P_0, producer movement m induces a new tree rooted in P_m.*

Proof. The initial tree rooted in P_0 gives routes to producer from all network nodes. *MAP-Me* update mechanism after movement m flips all directed links from P_m to the latest routed position P_j, $j < m$, so that they point to P_m. In the presence of multiple concurrent updates, the most recent one, i.e. the one with the highest sequence number, also propagates back along the routes of the encountered previous updates. Thus, update completion will merge different rooted trees into the one of highest sequence number, m, rooted in P_m. □

Corollary 1. *MAP-Me is loop-free under loop-free global routing.*

Proof. Starting from the spanning tree given by global routing, Prop.2 states that *MAP-Me* induces a new tree, as it only flips all edges over the unique path from the original position to the new one. Indeed, given the unchanged number of links/nodes, the result is still a directed tree rooted in the new position. Hence, it is loop-free. □

Proposition 3. *MAP-Me path stretch for node i over the tree rooted in P_m, created after producer's m-th movement,* *is upper bounded by the ratio* $(|i - P_0|_{P_0} + |P_0 - P_m|_{P_0})/|i - P_m|_{P_m}$ *as* $m \to \infty$*, which corresponds to the path stretch of the anchor-based approach with anchor in* P_0.

Proof. We can distinguish two cases according to whether P_0 is on the path between i and P_m on the P_m-rooted tree or not. If it is, then the path between i and P_m may be split into the paths i to P_0 and P_0 to P_m. The second component is equal to the path length between P_m and P_0 on the initial tree (only directions have been flipped).

The first one corresponds to the same path on the initial tree even in terms of directions. Therefore, the path stretch in this case is exactly equal to $(|i - P_0|_{P_0} + |P_0 - P_m|_{P_0})/|i - P_m|_{P_m}$. Otherwise, if P_0 is not on the path between i and P_m, the path between i and P_m is, by definition of *MAP-Me* update process (that utilizes the shortest path routing for IUs), shorter than the one including the detour via P_0 on the initial P_0-rooted tree. The bound remains true as $m \to \infty$, because it is intrinsically related to the properties of the initial tree. □

B. Numerical evaluation of path stretch

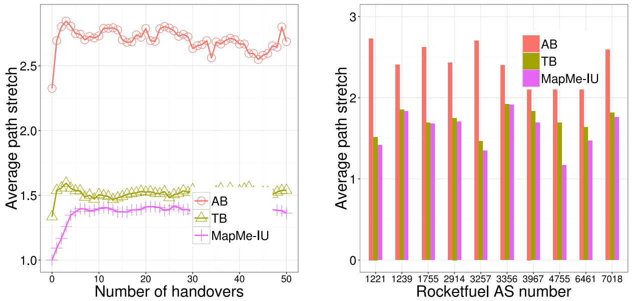
We compute now the average path stretch obtained by AB, TB, and *MAP-Me-IU* (*MAP-Me* requires geographical mobility and will be later considered in Section VI) on the rocketfuel topologies [2]. The initial position on the consumer, producer, and eventual anchor are chosen randomly, and two mobility models are implemented : (i) uniform – like in the related work – where the producer can jump towards any other node from the graph, and (ii) random waypoint (RWP), where the producer chooses a waypoint like in the previous approach, but advances hop-by-hop on the shortest path towards that waypoint, and then starts again. We average over 1000 runs to compute ensemble average for the path stretch after k movements of the producer with small confidence intervals.

Figure 4(a) represents the evolution of average *MAP-Me* path stretch over AS 1221 topology under RWP (other parameters all show similar trends). We observe that path stretch stabilizes after 10 movements, because *MAP-Me* preserves the initial structure of the forwarding tree (it only modifies links direction). Other Rocketfuel topologies show qualitatively the same results. A comparison of the different approaches over the all 10 Rocketfuel topologies is in Fig.4(b). Under both uniform and RWP mobility, *MAP-Me* outperforms AB, achieving up to 55% stretch reduction, and performs as well as TB or better.

VI. Evaluation

A. Simulation setup

The section gathers simulation assessment of *MAP-Me* over different mobility pattern, radio conditions and network topologies. We implemented both *MAP-Me* and *MAP-Me-IU*, anchor-based (AB), tracing-based (TB) – based on Kite ([44]) – and GlobalRouting (GR) approaches in NFD within the NDNSim 2.1 framework. We don't consider here resolution-based (RB) or other AL solutions as they are not appropriate for latency-sensitive applications (see discussion in Sec. II).



(a) AS 1221 stretch evolution

(b) Stretch comparison

Fig. 4: Path stretch evolution (a) and comparison (b) over Rocketfuel topologies.

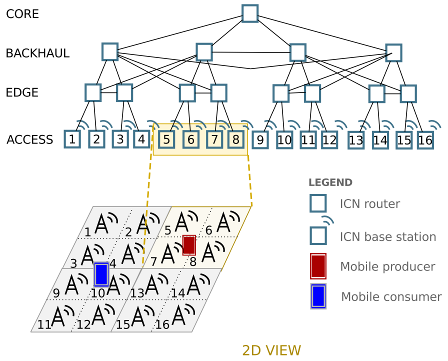


Fig. 5: Network with link capacity C=10Mb/s.

We first evaluate all mobility protocols in a baseline scenario, before varying parameters such as radio conditions, mobility model and network topology in order to gain insight into their sensitivity. All plot data is averaged over many runs, or a large number of handover(at least 250 per mobile node per run) depending on the context; although, for clarity, we chose not to display confidence intervals in the paper. The full set of results is available in [5].

B. Baseline scenario description

Topology: In the baseline scenario, we use 802.11n access network composed of a 4-by-4 grid of base stations (BS) with square-shaped cell of side $s = 80m$. They are connected to a fat-tree backhaul network represented in Fig. 5. This choice is motivated by the similarity in terms of redundancy and meshing found in real ISP access network. Wired links have a capacity of C=10Mb/s and 5ms delay. We complement it by a wide range of well-known topology models or Rocketfuel topologies to cover all types of graph metrics in the variant of baseline of scenario in section VI-D.

Radio and Mobility: We use IEEE 802.11n WiFi on 5GHz frequencies, with Minstrel rate adaptation [26] and log distance propagation model plus Rayleigh-fading model for wireless channel. Mobile nodes move in the 4x4 cells under full radio coverage. We choose random waypoint mobility model for user mobility. We also vary mobile's moving speed from 1m/s to 15m/s (i.e, pedestrian to vehicular speed). A range of other

[2] http://research.cs.washington.edu/networking/rocketfuel/
We extract the undirected graph corresponding to the largest connected component as in [38], [44]. We remark that the authors either did not use the same data, or processed the graph in an undocumented way, which prevents us to reproduce their exact results. Ours are still qualitatively similar.

radio propagation models and mobility models are also used in the variant scenario in section VI-D.

Application: We assume N disjoint pairs of mobile consumers and producers. In particular, we choose N=5 for baseline scenario and also its variants. To highlight *MAP-Me* benefits in the support of latency-sensitive traffic, we consider a streaming audio/video application, characterized by a CBR rate of 1Mbps with no retransmission in the baseline scenario, and further extend it with an adaptive protocol inspired by the Periscope streaming application in Sec. VI-F. While, the CBR application has the nice property of reflecting network performance, the adaptive one has a closed-loop behavior that is more realistic but might be affected by wireless and mobility losses. More in-depth study of these interactions is out of scope for this paper.

C. Results for baseline scenario: Fat-Tree + RWP + CBR

1) User performance: In Fig.6(a)-6(b), we show two performance indicators for latency-sensitive traffic, average packet loss and delay, both as a function of mobile speed (from 1m/s to 15m/s). We can distinguish two kinds of losses: due to the wireless medium, occurring irrespective of the mobility management approach, and those due to mobility. The fraction of mobility losses is consistently reduced by *MAP-Me*, especially in the presence of the notification/discovery mechanism, as a result of in-fly re-routing of Interests towards the new location of the producer, which prevents Interest timeouts. *MAP-Me-IU* like TB (or alternative AL solutions) enables re-routing of Interests only after the interval of time required for an update to complete. A longer time is required for a global routing update, but the resulting path is the shortest possible, which explains the equivalent performance w.r.t. *MAP-Me-IU*/TB. AB under performs because of worse update completion time and path stretch. The experienced average packet delay in Fig.6(b) is a consequence of the path stretch of different approaches: high for AB, medium for TB or *MAP-Me-IU*, low for GR. *MAP-Me* achieves better performance especially at high speed when the discovery/notification mechanism is mostly used in virtue of the shorter 1-hop forwarding between APs at the access that does not involve upper links in the topology (at the edge level). As explained, packet losses and delay result from the different average path lengths associated to each mobility update process, see Fig.6(c), and from the L3 hand-off latency, i.e. the time required for L3 reconnection after a handover, see Fig.6(d). The L3 hand-off latency illustrates the reactivity of the mobility-management protocol and highlights the significant improvement brought by *MAP-Me*, which significantly reduces handoff latency compared to other approaches. It is interesting to observe that AB shows a constant latency value of around 30ms due to update propagation up to the anchor, while for GR, TB, and *MAP-Me-IU*, such latency varies according to network distance between producer and routers to be updated, as a function of producer movement in the considered topology. Latency variations can be visualized at the inflection points in the corresponding CDFs in Fig.6(d).

2) Network cost: If user performance is critical to drive mobility-management choice, network cost analysis is equally important for the selection of a cost-effective solution. To this aim, we compare signaling overhead, meaning the total number of control messages triggered by a handover, in Fig.7(a), and the volume of signaling messages per handover to be processed by routers at different positions in the network, in Fig.7(b). More precisely, in the latter case, we visualize the distribution over the network of signaling load by distinguishing the average number of messages per handover received by different classes of routers, based on their position in the network: access, edge, backhaul, core as indicated in Fig.5. As expected, the overall number of signaling messages as a function of mobile speed is constant for AB, equal to the number of hops from mobile nodes to the anchor (4). Instead, it varies for *MAP-Me* and *MAP-Me-IU* according to the also varying average hop count (i.e. path stretch), as already observed in Fig.6(c). TB approaches involve a much higher signaling overhead due to "keep-alive" messages periodically sent to refresh update information. By reporting the way traffic is spread across the network and where signaling traffic goes, we can draw some key observations. Every mobility protocol relies on the control plane that enforces a routing state across the network (shortest-path routing in this paper), which corresponds to the initialization state for mobility. All protocols relying on an anchor have routing pointing to the anchor's location, whereas for AL mechanisms, it points to the producer's position at the routing update time. Thus, AL approaches are able to offload mobile backhaul and core networks from all local traffic, seamlessly (Fig.7(c)[3]). Finally, we report about *MAP-Me* sensitivity to parametrization, i.e. the impact of TU settings. In Fig.7(d), we observe that *MAP-Me* has robust parametrization as long as TU is not too small (signaling overhead and path stretch quickly converges to the best settings) or too high (load on access).

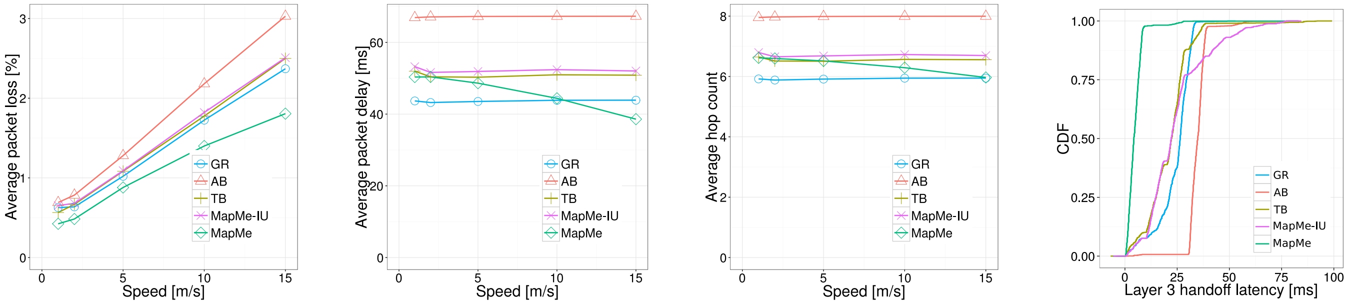
D. Impact of mobility pattern, radio conditions and topology

We have performed extensive simulations to evaluate the sensitivity of *MAP-Me* and other solutions, by varying several parameters in our baseline scenario [5]. We report here our most significant results and confirm the wide applicability of conclusions from the previous section.

1) Impact of mobility pattern and radio conditions: For mobility patterns, we have included the previous jump models across base-stations and classical models available in NS-3 (Random Direction 2D, Gauss-Markov and Random-Walk 2D models). For radio conditions, we have considered an ideal wireless channel (no loss nor interference at layer 2) by dynamically switching wired links up and down to emulate mobile handover across base stations, and the two radio models from ITU specifications [15], namely urban environment without line of sight (LoS), and suburban with LoS.

The impact of both radio and mobility patterns are negligible, and the plots show no significant difference between performance metrics. Comparative simulations with the ideal wireless channel (not represented here) show that the loss rate is not only due to the wireless channel, but also impacted by the mobility scheme in place, and more specifically the time

[3] For clarity, utilization of access link only represents traffic between base stations, excluding upstream from mobiles.



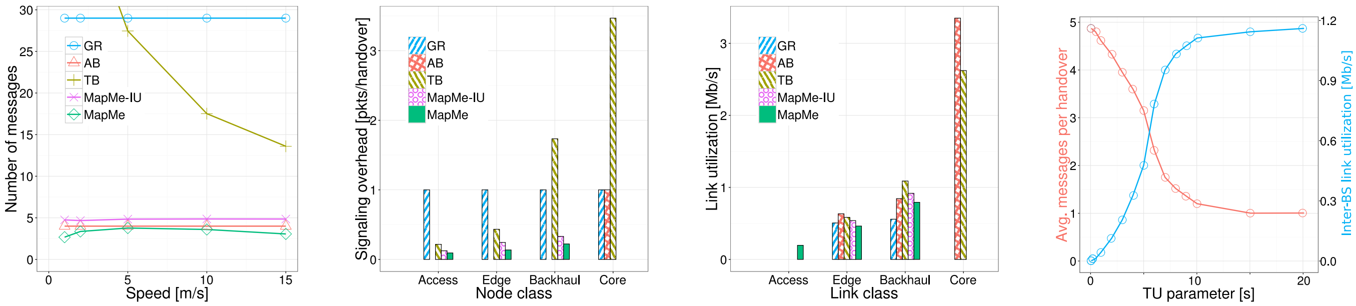
(a)

(b)

(c)

(d)

Fig. 6: User performance: packet loss (a), delay (b), and hop count (c); CDF L3 hand-off latency (d).



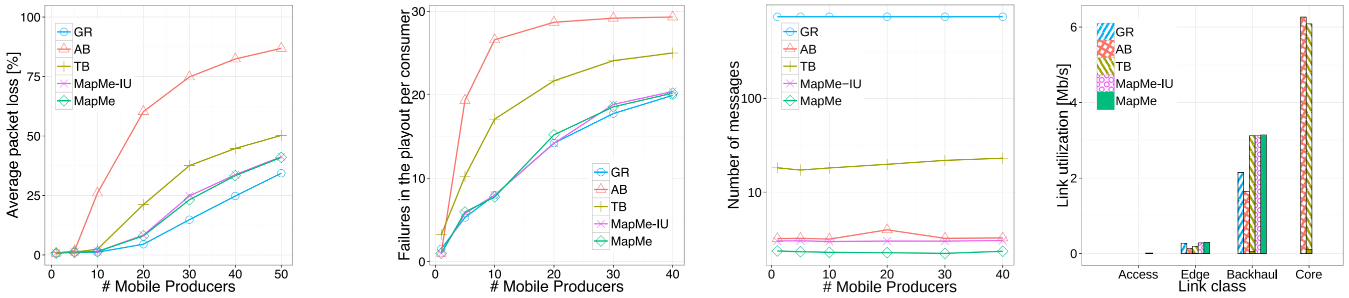
(a)

(b)

(c)

(d)

Fig. 7: Network cost: Signaling overhead vs mobile speed (a), overhead (b), and link utilization (c) per router class. Map-Me sensitivity analysis (d).



(a)

(b)

(c)

(d)

Fig. 8: User performance: CBR average packet loss (a), Periscope playout failures (b). Network Cost: CBR overhead (c) and Periscope link utilization (d).

to reestablish connectivity at L3 (denoted L3 handover time). Moreover, with ideal wireless channel where we can extract out the impact of only L3, the relative order of performance of protocols are the same as those in Fig 6(a), confirming *MAP-Me*'s superior performance in reducing mobility losses.

2) Impact of topology: We cover a wide range of network characteristics through the use of deterministic and stochastic graphs drawn from well-known models [5], as well as previously described Rocketfuel topologies. While not being representative of access networks, they provide insights into the performance of *MAP-Me-IU* in non-local mobility (e.g. jumps from WiFi to LTE networks). Edges nodes are randomly picked from graph nodes (or Rocketfuel leaves) to be connected to the previously described grid, while others form the backhaul.

As expected, topology is the most impacting parameter for absolute performance, being the direct consequence of the forwarding trees built atop. Figure 9 shows path stretch and L3 handoff latency. As hinted in Sec. V-B and shown in previous simulations, *MAP-Me-IU* and TB both offer lower stretch than AB – sometimes close to optimum (GR) – with a slight advantage to *MAP-Me-IU* in almost all cases. Those variations can be interpreted as the ability for the spanning tree (shortest path tree rooted in anchor for TB, and first producer location for *MAP-Me-IU*) to offer short paths between consumer and producer, and thus offload traffic at the edge.

Available path lengths are reflected in the CDF of Layer 3 handover latency, and we see that *MAP-Me-IU* is able to find shorter paths for close-by nodes (effectively offloading traffic), while those towards remote nodes are less optimal than if we were going through the anchor like in TB (hence the crossing of both curves). The reason is we had to enable

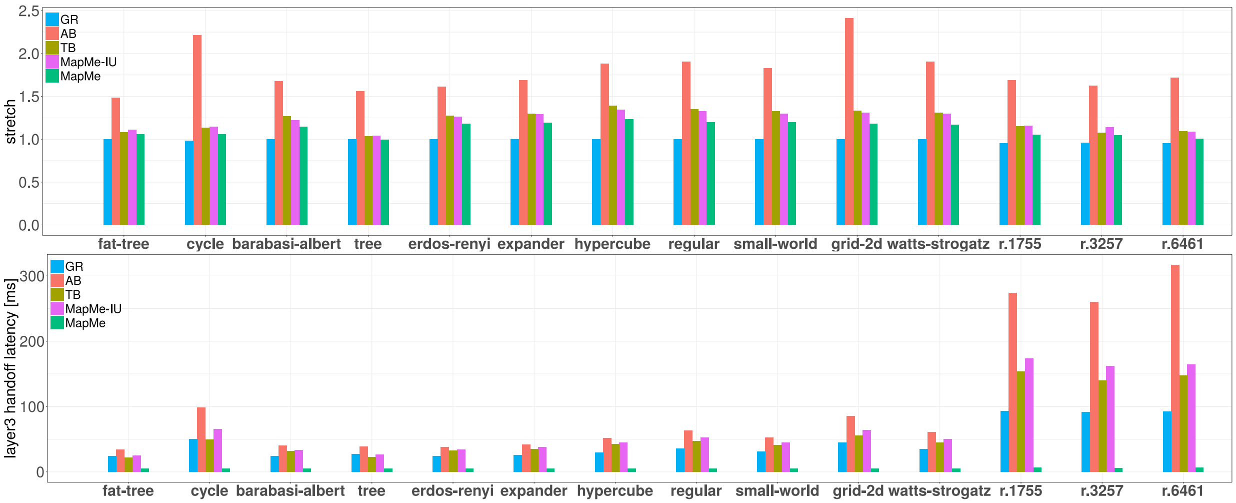


Fig. 9: Path stretch and handoff latency for simulated network topologies (r.1755, r.3257, r.6461 are Rocketfuel topologies).

all optional extensions in TB as in the default setting some situations could lead to unreachability of the producer. One consists in duplicating interests both on the trace and towards the anchor, which had the side effect on selecting the shortest of the two paths. This simple alternative is the reason of the much lower delays observed in TB.

In all cases, we see the extremely low handoff delay ensured by *MAP-Me*, which confirms the benefits of notification to reduce the time the producer is disconnected, and thus support latency-sensitive applications during mobility.

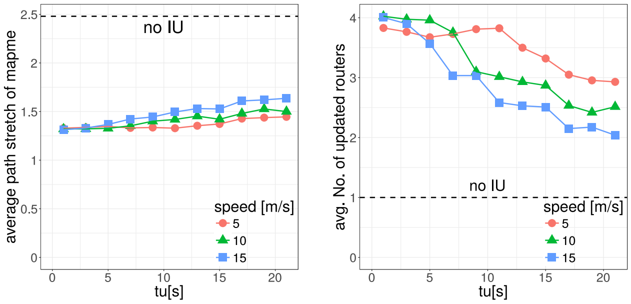
Beyond confirming our previous observations, these simulations open the way to further extensions of *MAP-Me* by considering how an alternative routing might lead to better performance – for instance using more efficient spanning trees (ST) such as minimum diameter ST (see Prop.3 in Sec. V-A) – and how more appropriate graph spanners and random strategies could allow the exploration of more than one path.

E. Impact of notifications on path stretch

As we have seen, the use of notifications improves performance during fast mobility by using inter-PoA links with the risk of increasing path stretch. We show here that the use of T_U as per the selected mechanism (Sec. III-D) changes the root of the IN's breadcrumb chain and thus limits its length. We thus evaluate the trade-off offered by *MAP-Me* through the adjustment of this T_U parameter by slightly modifying our baseline scenario. Instead of a grid, the PoA are arranged on a line. The producer now moves back and forth across them at a constant speed parameter, while the consumer is now static at the root of the fat tree.

Fig. 10(a) shows the average path stretch of *MAP-Me* as function of Tu. The dashed line indicates the path stretch limit reached when no IU is sent. In general path stretch slowly increases with T_U at any given speed and remains well below the no-IU threshold. At low speed, stretch remains constant up to higher T_U values (as an IU is sent for every handover).

If we now consider network overhead depicted in Fig. 10(b), we notice that a slight increase of path stretch allows for a significant reduction of network overhead (which peaks here at 50% for a speed of 15m/s). This confirms the interest of notifications in absorbing high-frequency mobility while preserving appropriate flow performance. The T_U threshold thus appears a useful setting to allow a network to cope with challenging mobile workloads.



(a)

(b)

Fig. 10: Effectiveness of Tu timer: a) Path stretch b) Network overhead (No. of updated routers per handover).

F. Trace-driven urban mobility

Topology: To evaluate our approach under more realistic mobility patterns, we consider an urban residential environment spanning a 2.1×2.1 km^2 area in Los Angeles, with a WiFi Hot Spot deployment similar to what Time Warner Cable [4] has in the area, i.e. we dislocate 729 WiFi APs, with the same wireless settings as in the previous (baseline) experiments, connected to the Internet through the fat-tree topology in Fig.5.

Mobility: We generate realistic vehicular mobility patterns using SUMO [21], with maximum car speed set according to road speed limits[5]. We place mobile producers in moving cars and analyze system dynamics on a given time interval (4 minutes, roughly corresponding to 33 handovers), so that all monitored cars are in the map at the same time. In such a scenario, we consider a group communication between one mobile producer and two non mobile consumers requesting different data. Consumers are connected to two APs that are picked at random, uniformly across the network coverage.

[4] http://coverage.twcwifi.com/

[5] In the selected area we have three different road categories characterized by different speed limits: 40, 70 and 55 km/h.

Applications: Two types of applications are considered: in the first set of simulations, the previous 1Mbit/s CBR application; in the second, a pseudo real-time video streaming application, reproducing the popular application Periscope is used. The mobile producer generates two different video streams, each one downloaded and played by one consumer, using a 5 s play-out delay buffer. If the video play-out stops because the consumer has no Data available, we consider this as a failure and momentarily stop the consumer: after a short period of time (few seconds), the consumer restarts downloading new data and to play-out the video. The video data rate is 1Mbit/sec, corresponding to a 480p video resolution.

Traffic is scaled up by increasing the number of groups, identified by the producer serving data.

1) User Performance: To quantify user experience, we analyze the following metrics: the average packet loss and user satisfaction, while varying the number of mobile producers in the area (from 1 to 50, each one serving two consumers).

Packet loss: We evaluate the distribution of packet losses per second for the CBR application. Fig. 8(a) shows the average packet loss, while increasing the number of mobile producers in the system. As expected, increasing the number of active users in the network has a negative effect on performance, because links are more congested and routers start to lose packets. However, as shown in Fig. 8(a), the performance of *MAP-Me* and *MAP-Me-IU* is close to the ideal GR, while TB leads to higher loss rate and with AB, we observe an even more rapid increase in packet loss. Indeed, the distributed nature of *MAP-Me* allows the proposed solution to better cope with an increasing number of mobile producers.

User Satisfaction: We evaluate user satisfaction by analyzing the number of failures in the play-out of the video stream for the pseudo real-time video streaming (Periscope-like). Fig. 8(b) shows the number of failures in the video play-out that each consumer encounters in 4 min. As in the CBR case, when the number of mobile producers increases, the performance of the system degrades. Again, AB concentrates all traffic on a single node, the anchor, thus giving rise to congestion. In contrast, distributed protocols such as *MAP-Me* are able to better distribute traffic over the network and thus better cope with larger number of users. For the same reason, TB performs better than AB, but worse than *MAP-Me*/GR. Indeed, sending traces to the anchor forces traffic towards upper layers in the network, preventing substantial traffic offload at the edge.

These simulations clearly show the effectiveness of *MAP-Me* in dealing with high loads as it spreads traffic over a more diverse set of paths.

2) Network Cost: Beyond user performance, we evaluate *MAP-Me* in terms of network cost, by computing the overhead and comparing it with all other considered solutions. Fig.8(c) reports the overhead, computed as the number of messages exchanged in the network at each handoff, whereas Fig.8(d) displays link load distribution across the network (in the case of 10 mobile producers in the map). The figures prove that *MAP-Me* successfully offloads the core from local traffic with light overhead, in virtue of its anchor-less characteristics.

Network topology and Mobility: Trace-based simulation have been run with pedestrian mobility and a tree-like network topology [5]. Results show the same behavior for vehicular and pedestrian mobility, while in the case of tree topology TB and *MAP-Me* have similar packet loss (due to higher chances of congestion at the core of the network).

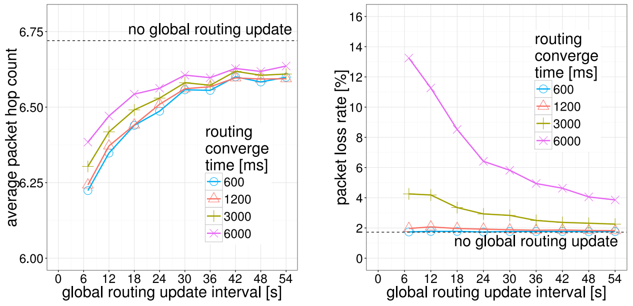
VII. *MAP-Me* and routing

While *MAP-Me* can efficiently manage producer mobility by updating FIB entries, it might however interfere with routing protocol as both can update FIB concurrently. In this section, we discuss their coexistence and show that minimal requirements on the routing and minor modification to *MAP-Me* can allow for both to perform correctly and asynchronously. We conclude by preliminary insights into their joint performance.

Proposed Solution: Our proposal makes minimum assumptions on properties of the routing protocol: (i) the routing protocol is *link-state* so that every node gets a sense of routing convergence state; (ii) every router maintains a counter R_{seq}, incremented each time a non-duplicated routing message (LSA) is received – **Rseq** is expected to be either available or easily deducible from routing; and (iii) a routing instance is also running on the producer so that the producer is informed of network changes. We assume the router generating a new prefix advertisement or detecting a link failure will also increment this counter for global consistency.

On *MAP-Me* side, the idea is to delay *MAP-Me* 's operation on a node until routing seems to converge locally (by checking $R_{\mathbf{seq}}$). We achieve this through a minor modification to the original design: upon sending a special interest, the sequence number field is augmented with the local $R_{\mathbf{seq}}$ information. When IU/IN is received, additional checks are performed before standard *MAP-Me* operation: by comparing R_{seq} in IU/IN ($R^{\mathrm{IU}}_{\mathrm{seq}}$) and the local one from routing ($R^{\mathbf{loc}}_{\mathbf{seq}}$). **Case (i)** if $R^{\mathrm{IU}}_{\mathrm{seq}} = R^{\mathbf{loc}}_{\mathbf{seq}}$, the producer and the nodes might be synchronized, and standard operations can proceed; **case (ii)** if $R^{\mathrm{IU}}_{\mathrm{seq}} > R^{\mathbf{loc}}_{\mathbf{seq}}$, the node has not received all routing updates and the IU is queued until $R^{\mathrm{loc}}_{\mathrm{seq}}$ gets incremented by routing, and eventually the IU pass through the node; **case (iii)** if $R^{\mathrm{IU}}_{\mathrm{seq}} < R^{\mathbf{loc}}_{\mathbf{seq}}$, the IU is discarded as all downstream nodes does not receive all routing updates. Finally, to ensure correctness, we require the producer to issue a new IU each time it receives new routing messages (i.e, $R_{\mathbf{seq}}$ incremented). This IU corrects the route if routing recomputes route towards producer's old location due to network changes and unawareness of producer's new location.

Correctness: This scheme ensures full producer reachability upon global convergence. Considering a single producer update during routing convergence, it is easy to see that the corresponding IU will traverse all routers that have seen the same number of routing updates as the producer. It is otherwise either delayed by case (ii) or dropped (iii). The last IU sent by the producer is guaranteed to complete (as there are no routers with higher R_{seq}, and that the forwarding tree is consistent as all routers have then the latest routing state. During routing instabilities, there is no guarantee of connectivity and the forwarding state might not be loop-free either. It seems natural that we cannot require *MAP-Me* to improve on that situation. The design of a joint routing and mobility management protocol, following the same principle as *MAP-Me*, is an interesting direction left for future work.



(a)

(b)

Fig. 11: *MAP-Me* and routing. Effects of routing update frequency on performance: (a) Packet loss rate. (b) Path stretch.

Evaluation: We now illustrate the behavior of the modified algorithms, and analyze the effect of routing updates frequency on system performance. We consider the previous baseline scenario with 1 pair of mobile nodes, and a speed of 10m/s. The producer triggers a new routing update with varying frequency. Routing convergence time obviously impact performance significantly. It is generally considered that link-state IGP convergence time is in the order of several seconds. while [11] demonstrates the possibility for sub-second convergence time for large ISP networks by leveraging techniques like fast flooding and incremental FIB updates, it is not widely deployed. We thus reasonably assume the routing convergence time lies in the order of sub-second to several seconds. In the evaluation we choose between 600ms and 6s.

Figure 11 illustrates the trade-off in setting the routing update frequency. Obviously, more frequent updates allow for shorter paths as they are re-optimized more often (Fig. 11(a)). However, instabilities due to routing at global scale lead to long-lasting unreachability of the producer after he moves, and thus a high packet loss rate (Fig 11(b)). Routing updates should thus be limited or triggered carefully, for instance in periods of producer stability (e.g based on mobility prediction). Nevertheless, Fig 11(b) shows also that when routing converges in sub-seconds, the interaction with *MAP-Me* runs smoothly and without substantial loss in performance.

VIII. Conclusions

Native support for mobility management at network layer is a recognized strength of ICN, and appears to be a key feature to exploit the design of 5G networks. However, a comprehensive solution for mobility management in ICN still lacks: previous attempts so far have either tried to apply Mobile IP concepts to ICN or looked at partial aspects of the problem, without providing a thorough evaluation of the initial solutions sketched in ICN context. The contribution of this paper is twofold. First, we look at CCN/NDN, two prominent ICN architecture, and define *MAP-Me*, an anchor-less model for managing intra-AS producer mobility even in the presence of latency-sensitive traffic. By design, *MAP-Me* is simple as it only leverages CCN/NDN forwarding plane and reactive notifications to the network, is lightweight in terms of required signaling messages and, to our knowledge, the first one with proven guarantees of bounded stretch and overall correctness for the forwarding update process. Second, we opensourced a simulation framework on top of NDNSim 2.1 using model-based and trace-driven consumer/producer mobility patterns over many topologies, and integrating anchor-based and tracing-based approaches, a reference implementation for *MAP-Me*, as well as a global routing approach useful for comparison. Evaluation takes 802.11n access in small cell outdoor settings and proves WiFi can support mobility using CCN/NDN in general settings.

The reported results confirm our initial objectives and show that *MAP-Me* optimally offloads the infrastructure from communications that are local. All other approaches making use of an anchor, which in practice is also the network gateway, can be optimized only if traffic is non local. Instead, the current propositions in 3GPP to offload the mobile network core stem from the observation that, on the contrary, communications are most likely local. On the other hand, *MAP-Me* would serve non-local communications through one or multiple gateways without binding mobility feature to any specific location.

References

[1] Source code for paper "MAP-Me: Managing Anchor-less Producer Mobility in Content-Centric Networks". https://github.com/mapme-tnsm17, 2017.

[2] A. Afanasyev, C. Yi, L. Wang, B. Zhang, and L. Zhang. SNAMP: Secure namespace mapping to scale NDN forwarding. In *Proc. Computer Communication Workshops*, pages 281–286, 2015.

[3] B. Ahlgren, M. D'Ambrosio, M. Marchisio, I. Marsh, C. Dannewitz, B. Ohlman, K. Pentikousis, O. Strandberg, R. Rembarz, and V. Vercellone. Design considerations for a network of information. In *Proc. CoNEXT*, page 66, 2008.

[4] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. A survey of information-centric networking. *IEEE Communications Magazine*, 50(7), 2012.

[5] J. Augé, G. Carofiglio, G. Grassi, L. Muscariello, G. Pau, and X. Zeng. MAP-Me: Managing Anchor-less Producer Mobility in Content-Centric Networks. Technical report, https://mapme-tnsm17.github.io/, 2016.

[6] G. Carofiglio, L. Muscariello, M. Papalini, N. Rozhnova, and X. Zeng. Leveraging icn in-network control for loss detection and recovery in wireless mobile networks. In *Proc. ACM ICN*, Kyoto (JP), Sep 2016.

[7] A. Compagno, X. Zeng, L. Muscariello, G. Carofiglio, and J. Augé. Secure producer mobility in information-centric network. In *Proc. ACM ICN*, Berlin (DE), Sep 2017.

[8] H. Farahat and H. Hassanein. Optimal caching for producer mobility support in named data networks. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–6. IEEE, 2016.

[9] B. Feng, H. Zhou, and Q. Xu. Mobility support in named data networking: a survey. *EURASIP Journal on Wireless Communications and Networking*, 2016(1):220, 2016.

[10] N. Fotiou, P. Nikander, D. Trossen, G. C. Polyzos, et al. Developing information networking further: From psirp to pursuit. In *Broadnets*, pages 1–13, 2010.

[11] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure. Achieving sub-second igp convergence in large ip networks. *ACM SIGCOMM Computer Communication Review*, 35(3):35–44, 2005.

[12] D. Han, M. Lee, K. Cho, T. Kwon, and Y. Choi. Publisher mobility support in content centric networks. In *Proc. of ICOIN*, 2014.

[13] F. Hermans, E. Ngai, and P. Gunningberg. Global source mobility in the content-centric networking architecture. In *Proc. of ACM NoM Workshop*, 2012.

[14] A. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang. Nlsr: named-data link state routing protocol. In *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, pages 15–20, 2013.

[15] ITU-R. Propagation data and prediction methods for the planning of short-range outdoor radiocommunication systems and radio local area networks in the frequency range 300 mhz to 100 ghz. Recommendation p.1441-9, International Telecommunication Union, Geneva, Jun 2017.

[16] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *Proc. of ACM CoNEXT '09*.

[17] X. Jiang, J. Bi, and Y. Wang. What benefits does NDN have in supporting mobility. In *Proc. of IEEE ISCC*, 2014.

[18] D.-h. Kim, J.-h. Kim, Y.-s. Kim, H.-s. Yoon, and I. Yeom. Mobility support in content centric networks. In *Proc. of ACM ICN 2012*.

[19] D.-h. Kim, J.-h. Kim, Y.-s. Kim, H.-s. Yoon, and I. Yeom. End-to-end mobility support in content centric networks. *International Journal of Communication Systems*, 28(6):1151–1167, 2015.

[20] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. In *ACM SIGCOMM Computer Communication Review*, volume 37, pages 181–192, 2007.

[21] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.

[22] J. Lee, S. Cho, and D. Kim. Device mobility management in content-centric networking. *IEEE Communications Magazine*, 50(12):28–34, December 2012.

[23] M. B. Lehmann, M. P. Barcellos, and A. Mauthe. Providing producer mobility support in ndn through proactive data replication. In *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*, pages 383–391. IEEE, 2016.

[24] D. Li and M. C. Chuah. SCOM: A Scalable Content Centric Network Architecture with Mobility Support. In *Proc. of IEEE MSN*, 2013.

[25] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang. ndnSIM 2: An updated NDN simulator for NS-3. Technical Report NDN-0028, Revision 2, NDN, Nov 2016.

[26] A. Mcgregor and D. Smithies. Rate adaptation for 802.11 wireless networks: Minstrel. Technical report, http://blog.cerowrt.org/papers/minstrel-sigcomm-final.pdf.

[27] Y. Nishiyama, M. Ishino, Y. Koizumi, T. Hasegawa, K. Sugiyama, and A. Tagami. Proposal on routing-based mobility architecture for ICN-based cellular networks. In *Proc. Computer Communication Workshops*, pages 467–472, 2016.

[28] Periscope. Video Streaming, https://www.periscope.tv/.

[29] R. Ravindran, S. Lo, X. Zhang, and G. Wang. Supporting seamless mobility in named data networking. In *Proc. of IEEE ICC*, 2012.

[30] J. W. Roberts. Realizing quality of service guarantees in multiservice networks. In *Performance and Management of Complex Communication Networks*, pages 277–293. Springer, 1998.

[31] I. Seskar, K. Nagaraja, S. Nelson, and D. Raychaudhuri. Mobilityfirst future internet architecture project. In *Proc. AINTEC*, pages 1–3, 2011.

[32] S. Shenker. Fundamental design issues for the future internet. *Selected Areas in Communications, IEEE Journal on*, 13(7):1176–1188, 1995.

[33] G. Tyson, A. Mauthe, S. Kaune, P. Grace, and T. Plagemann. Juno: An adaptive delivery-centric middleware. In *Proc. CCNC*, pages 587–591, 2012.

[34] G. Tyson, N. Sastry, R. Cuevas, I. Rimac, and A. Mauthe. A survey of mobility in information-centric networks. *Communications of the ACM*, 56(12):90–98, 2013.

[35] G. Tyson, N. Sastry, I. Rimac, R. Cuevas, and A. Mauthe. A survey of mobility in information-centric networks: Challenges and research directions. In *Proc. NoM*, pages 1–6, New-York (USA), 2012.

[36] X. Vasilakos, V. A. Siris, G. C. Polyzos, and M. Pomonis. Proactive selective neighbor caching for enhancing mobility support in information-centric networks. In *Proceedings of the second edition of the ICN workshop on Information-centric networking*, pages 61–66. ACM, 2012.

[37] L. Wang, A. Hoque, C. Yi, A. Alyyan, and B. Zhang. Ospfn: An ospf based routing protocol for named data networking. 2012.

[38] L. Wang, O. Waltari, and J. Kangasharju. ‘. In *Proc. of IEEE GLOBECOM*, 2013.

[39] G. Xylomenos, X. Vasilakos, C. Tsilopoulos, V. A. Siris, and G. C. Polyzos. Caching and mobility support in a publish-subscribe internet architecture. *IEEE Communications Magazine*, 50(7), 2012.

[40] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos. A survey of information-centric networking research. *IEEE Communications Surveys & Tutorials*, 16(2):1024–1049, 2014.

[41] R. Ying, L. Hongbin, G. Deyun, Z. Huachun, and Z. Hongke. Lbma: A novel locator based mobility support approach in named data networking. *China Communications*, 11(4):111–120, 2014.

[42] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, et al. Named data networking. *ACM SIGCOMM Computer Communication Review*, 44(3):66–73, 2014.

[43] Y. Zhang, A. Afanasyev, J. Burke, and L. Zhang. A survey of mobility support in named data networking. In *Proc. NoM*, 2016.

[44] Y. Zhang, H. Zhang, and L. Zhang. Kite: A mobility support scheme for ndn. In *Proc. of ACM ICN Poster*, 2014.

[45] Z. Zhu, R. Wakikawa, and L. Zhang. A survey of mobility support in the internet. *RFC 6301*, March 2011.

Jordan Augé received Engineering degree from ENSIIE, in 2004, and Ph.D. from Telecom ParisTech jointly with Orange Labs, in 2014. After joining University of Cambridge then UPMC/LiP6 for 6 years, he is currently a Research Software Engineer at Cisco Systems in the Information-Centric Networking project. His research interests span Information-Centric Networking, Traffic Engineering and Network Performance Evaluation.

Giovanna Carofiglio received Dr. Ing. in telecommunication engineering and in electronic and telecommunication engineering, both from Politecnico di Torino, Italy, in 2004, and PhD in telecommunication engineering jointly from Politecnico di Torino and Telecom Paris-Tech, Paris, France, in 2008. Her graduate research focused on stochastic analysis of wired and wireless networks and has been performed at Politecnico di Torino and Ecole Normale Superieure (ENS Ulm) in the INRIA-TREC group. She served for more than 6 years at Bell Labs as head of the research department on content networking. She presently works at Cisco Systems as Distinguished Engineer. She was general co-chair of ACM ICN 2014 and is a member of the IEEE.

Giulio Grassi received his Bachelor's degree in Computer Science in 2008 and Master's degree in Computer Science in 2012 both from University of Bologna, Italy. He is currently at his third year of Ph.D. program in University Pierre et Marie Curie, Paris, under the supervision of Dr. Giovanni Pau. His research interests are vehicular networks, information-centric networking, mobile systems and networks.

Luca Muscariello received Dr. Ing. in telecommunication engineering in 2002 and Ph.D. in electronics and communications engineering in 2006 both from Politecnico di Torino, Italy. He works at Cisco Systems as Principal Engineer and is a research associate at the IRT SystemX. He served for 10 years at Orange Labs (former France Telecom R&D) doing research and innovation in networking.He was program co-chair of Valuetools 2013, TPC chair of ACM ICN 2014, and general co-chair of ACM ICN 2014. He is a member of the ACM and a senior member of the IEEE and SEE.

Giovanni Pau is the ATOS/Renault smart mobility Chair Professor at the University Pierre at Marie Curie, Paris, France. He holds the Italian Laurea in Computer Science and PhD in Computer Engineering from the University of Bologna in 1998 and 2002, respectively. Before joining UPMC, Dr. Pau was a Senior Research Scientist at the UCLA Computer Science Department. Dr. Pau designed and built the UCLA campus vehicular testbed and the UCLA/MPI urban sensing testbed. More recently, he designed and developed VNDN, the Named Data Network (NDN) protocol stack adaptation for mobile-to-mobile scenarios.

Xuan Zeng is currently a Ph.D. candidate of University Pierre et Marie Curie and IRT SystemX in France, under the supervision of Prof. Giovanni Pau and Dr. Luca Muscariello. He received his M.S. in mobile communications jointly from Telecom Paristech in France and University of Politecnico di Torino in Italy. His main research interests are mobility management in ICN, rate/loss control, and NDN simulation/experimentation.