MAP-Me: Managing Anchor-less Producer Mobility in Content-Centric Networks (Extended Technical Report)

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 architecture, 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 applications. We analyze *MAP-Me* performance and provide guarantees of correctness, stability, and bounded stretch, which we verify on realistic ISP topologies. Finally, we set up a realistic simulation environment in NDNSim 2.1 for *MAP-Me* evaluation and comparison against the existing classes of solutions, considering random waypoint as well as trace-driven car-mobility patterns in 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.

***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.

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 [30], [3]. Among those different approaches, large attention from the research community has been focused on CCN [12] and on one of its evolution, NDN [32]. WHY DID WE PICK THOSE?

As a direct result of CCN/NDN principles, consumer mobility is naturally supported by the design: a change in physical location for the consumer does not translate into a change in the data plane like for IP. The retransmissions of requests for not yet received data by the consumers take 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. 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 [5] 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 [33] for a survey). They still rely on a stable home address to inform about producer movements (e.g. [34]) or on buffering of incoming requests at the producer's previous point of attachment – PoA – (e.g. [14]), which prevents support for latency-sensitive applications.

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

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

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 [22]. *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.
- 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 realistic simulation environment in NDNSim 2.1 [20], 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 waypoint and trace-driven mobility patterns representing V2I scenarios based on 802.11 radio access. Results show that *MAP-Me* outperforms the existing alternatives both in terms of user performance (e.g. loss, delays) and network cost (e.g. signaling overhead, link utilization) metrics.

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. In Sec. III, we introduce the design principles and details of *MAP-Me* operations, before analyzing its correctness and path-stretch guarantees in Sec. IV. A comprehensive evaluation of the benefits of our anchor-less proposal is then performed in Sec. V. Finally, Sec. VI investigates the interaction and possible cooperation between *MAP-Me* and an existing routing protocol, before concluding the paper in Sec. VII.

II. Related Work

A. Mobility management in CCN/NDN

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 [35]. Likewise, within the ICN family, different approaches to mobility–management have been presented [28]. In DONA [16] mobile publisher unregisters 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 [30]. Similarly, NetInf [2] and JUNO [26] report network mobility events to a resolution service, which may incur in network load in case of high mobility [27]. PURSUIT [7] 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. Finally MobilityFirst [24] uses a global name resolution service (GNRS), which is updated when a node changes point of attachment. Say something about MobilityFirst?

Specifically for the CCN/NDN solutions, several surveys of mobility-management approaches can be found [33], [6]. In [33] for instance, the authors distinguish three categories of solutions – routing, mapping, and tracing-based on the type of indirection point (also called Rendez-Vous, RV). We build on such classifications and refine it mainly to distinguish name-based approaches not relying upon the existence of an RV (anchor-less). We classify mobility-management solutions for CCN/NDN into four categries, Global Routing, Resolution-Based, Anchor-Based and Anchor-Less:

1) GR - Global Routing: It uses the routing plane and requires to update all network routers about the movements of a mobile node. It is a shared concern that these solutions suffer from scalability issues even in IP, thus, it may not be considered as a viable solution, in general, in the presence of frequent mobility and especially in CCN/NDN where the naming space is even larger than in IP. In the rest of the paper, we use an ideal global routing approach instantaneously updating all routers' tables as an ideal reference for comparison with other approaches.

2) RB - Resolution-based: These solutions maintain a mapping at routing-layer, involving a resolution of identifiers into locators to be performed at dedicated RV nodes (DNS-like infrastructure). Names are resolved into routable location identifiers: a DNS-like mapping system is updated every time the producer moves in the network and guarantees the correctness of the binding between content names and current producer location [10], [13], [19], [14], [1], [15]. Once the resolution is performed, packets can be correctly routed 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 [10], [13]; however, scalability is achieved at the cost of a large hand-off delay as evaluated e.g. in [14], [6] due to RV update and name resolution. To summarize, RB solutions show good 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 goals of *MAP-Me*.

3) AB - Anchor-based: Similarly to Mobile IP, these solutions keep the mapping at network-layer by using a stable home address (or anchor) as a RV. The anchor is kept aware of mobile node movements and is also responsible for tunneling packets to the new location. The anchor is assumed to be on the path followed by requests to locate the producer, and hence intercepts and relay them to the current producer's

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

location. MobiCCN [29] uses anchors distributed across the network and selects them in virtue of the proximity in the hyperbolic space. Those name resolution routers relay traffic to the mobile producer and are updated in case of mobility. Anchor placement becomes critical for the performance of the entire approach. In [18], instead, the producer needs to change its name prefix each time it moves to a new domain. Then, an update message will be sent by the producer to its anchor to notify its new name prefix. Encapsulation and decapsulation of packets(both in interest and data) must be performed at the anchor so as to relay interests to the producer or data to the consumer. The advantage of this approach is that the consumer does not need to be aware of the producer mobility. It has low signaling overhead because only the anchor needs to be updated. However, it inherits the drawbacks of Mobile IP: e.g. triangular routing and single point of failure. Furthermore, in the CCN/NDN specific case, this approach also has drawbacks specific to CCN/NDN: potential degradation of caching efficiency, bad integrity verification due to the constant change of name prefix while MP is moving. It also prevents CCN/NDN multipath capability and limiting robustness to failure/congestion. In contrast *MAP-Me* maintains names intact and avoid single point–of–passage of the traffic.

4) TB - Tracing-based: TB solutions require 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. They assume that the data is published under a global stable RV prefix and aim to create a "breadcrumb trail" connecting the current producer location to its stable RV location to relay consumer Interests to the mobile producer. Kite [34] leverages Interest notifications sent by the mobile producer to its RV to build a valid path to its current location via traces stored at all traversed routers into the PIT. 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 [9], 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.

5) AL - Anchor-less: AL approaches require the mobile node to signal its movements to the network by propagating name-based messages that are kept by all involved routers to guarantee reachability at the new location without requiring any 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. [23] exploits multicast and directs the same Interest to the nearby PoAs of the producer. In [31] and in the *Interest Forwarding* scheme proposed in [14] instead, 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. [21] 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.

Remarks: In-network caching and name-based routing in CCN/NDN also enable a routing-to-replica approach careless of producer movements (referred to as data depot in [33]). However, such an 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 [13].

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*.

We recall that we focus on an **anchor-less name-based layer-2 agnostic approach operating at CCN/NDN forwarding plane**. In the quest for purely Anchor-Less designs, we target a solution with the following additional characteristics:

- *Transparent:* so that no differentiation is required between mobile consumers and producers, nor any handover prediction or name change. In particular, the latter avoids issues like triangular routing or caching degradation, as well as security weakness.
- *Distributed:* we design *MAP-Me* to be fully distributed, involving routers at the edge of the network and thus realizing effective traffic off-load close to the end-users. Robustness issues like single point-of-passage problem due to centralized mobility management are prevented and higher route diversity allows for better use of multipath capabilities and more efficient resource utilization.
- *Lightweight:* we consider prefix granularity in updates, rather than content or chunk granularity, to minimize signaling overhead and temporary state kept by in-network nodes.
- *Reactive:* we introduce network notifications and discovery mechanisms to support real-time producer tracking and accommodate latency-sensitive communications.
- *Robust:* to network conditions (e.g. routing failure, wireless or congestion losses, and delays), by leveraging hop-by-hop retransmissions.

A. *MAP-Me* description

MAP-Me definition assumes the existence of a routing protocol responsible for creating/updating the Forwarding Information Base (FIB) of all routers, possibly with multipath routes, and for managing network failures. As a data plane protocol, *MAP-Me* builds on top of it and handles producer mobility events thanks to dynamic FIB updates with the objective of minimizing unreachability of the producer.

The rationale behind *MAP-Me* is to let the producer announce its movements to the network by sending a special Interest packet, named *Interest Update* (IU), once the producer reattaches to the network. Such a message looks like a regular interest sent by the producer to "itself" – the Interest Update specifies the prefix advertised by the producer. After a handoff, the IU generated by the producer is forwarded according to now-stale information stored in the FIBs of successive routers (the network does not know yet the new producer's location) until it reaches a former PoA. A special flag carried in the header of the IU causes all routers on the path to update their FIBs to point to the ingress IU face, pointing towards the new location of the producer (in the current version of *MAP-Me*, we assume that producers make use of a single face to serve consumers).

The key aspect of the proposal is to avoid relying on a stable home address (as opposed to Tracing-Based approaches for instance) and rather use name-based forwarding state created by CCN/NDN routing protocols or left by previous mobility updates, to switch FIBs on-the-fly to point to the correct new location of the producer. Such lightweight and distributed behavior allows the protocol to react at a faster timescale than routing – allowing more frequent and numerous mobility events – and to offload some traffic at the edge of the network. We thus expect it to reduce the link load and minimize disconnectivity time, which are the main factors affecting user flow performance, as later shown by our evaluation in Section V.

Let us now describe more in detail the central component of *MAP-Me*, i.e. its mobility update protocol.

B. Mobility update protocol

1) Objective and distributed realization: The update protocol aims at quickly restoring global reachability of mobile prefixes with low signaling overhead while introducing a bounded maximum path stretch. This is in contrast to the objectives of a routing protocol, which typically realizes global path optimization based on an objective such as the shortest path tree towards each prefix.

Given any prefix p, the global reachability objective is achieved by *MAP-Me* in a distributed way, by enforcing a mutual relationship between adjacent routers involved in the forwarding. We impose that when R uses neighbor S as a next hop to reach p (denoted $R \to S$), it also commits to inform S about any change in producer location it learns, which will occur if and only if the producer attaches to a predecessor of S in the DAG formed by the forwarding paths to p. In that case, an IU will propagate to R, then S, and will rightfully flip the link to $S \to R$. Otherwise, the link $R \to S$ remains valid and nothing has to be done by either R or S.

2) Concurrent updates: To prevent inconsistencies due to concurrent updates (e.g. caused by frequent mobility), *MAP-Me* maintains a sequence number that it increases at each handover and associates to the IU. The intermediate routers also remember this sequence number to distinguish the most recent update in case of multiple IU reception. Modification of FIB entries is only triggered when the received IU carries a higher sequence number, while the information associated with an older IU is simply discarded, and the same IU is sent back to the originating interface with an updated sequence number to fix those paths. Finally, in case of equal sequence number (which might happen in case of multipath), the new ingress face is added to the FIB without further propagating the IU (duplicate suppression).

While we believe that *MAP-Me* and routing protocols could share some route-versioning information, this is out of the scope of this paper and left for future work. We will describe *MAP-Me* independently of the routing protocol, and dedicate the last section to how they can safely interact.

3) Loss resiliency: The loss of an IU due for instance to the radio channel might disrupt this distributed behavior. We resolve this issue by introducing a hop-by-hop acknowledgment and retransmission mechanism which fits this mobile environment: the IU is retransmitted if no acknowledgment (IU_{ack}) has been received within τ seconds of its transmission.

4) Dynamic adjacency management: Upon reception of a fresh update (with either higher or equal sequence number), any forwarding information (next hops) associated to lower sequence number is removed and the corresponding faces enter state $\mathcal{F}_{\mathcal{P}}$. New next hops are deduced from the content of the IU, which is then forwarded (unless it reaches a previous position and all nodes have finally been reconnected).

Every router implementing *MAP-Me* has to maintain a dynamic adjacency state with its neighbors. This can be realized by considering the set $\mathcal{F}$ of active faces – those involved in the forwarding DAG – and its partition into three states as $\mathcal{F} = \mathcal{F}_{\mathcal{N}} \sqcup \mathcal{F}_{\mathcal{P}} \sqcup \mathcal{F}_{\mathcal{A}}$:

- $\mathcal{F}_{\mathcal{N}}$ is the set of faces connected to a next-hop according to the current forwarding state; this information is typically already present in the FIB in a static scenario and we simply need to associate it with a sequence number;
- $\mathcal{F}_{\mathcal{P}}$ is the set of faces associated to a previous next-hop, which has been revoked due to an IU reception. The IU has been forwarded as per the commitment made by the router, and this state is maintained for retransmission until an Ack is received. This is a transient list of faces with associated retransmission timers that we denote as TFIB (Temporary FIB). It materializes a link from the original DAG that is not currently active (in either direction), and should thus be restored for correctness.
- $\mathcal{F}_{\mathcal{A}}$ is the set of faces associated to a previous next-hop for which the IU has been acknowledged. The remote router is now pointing towards the new producer's location and will inform about any forwarding update that needs to be considered. Nothing has to be stored.

C. Illustration of the Update protocol

1) Update propagation following producer mobility: We illustrate the different events following a producer's movements in a toy network with a single producer serving prefix p and moving from position P_0 to P_1. Network FIBs are assumed to be populated by a name-based routing protocol. For the sake of clarity, we only represent active links in the graph, and consider the simpler case with no multipath route, which simplifies the forwarding from a DAG to a directed tree (Fig. 1(a)). Initially, all routers consistently point to P_0,

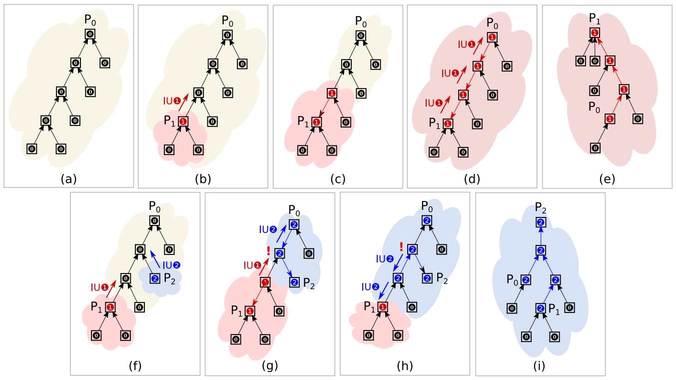


Fig. 1: Interest Update propagation example.

represented as a yellow cloud. We can also assume without loss of generality that all sequence numbers are set to 0.

Once the physical handover is completed, the producer sends an IU with an incremented sequence number (1 in this case) towards p, that is towards (one of) its previous location(s) according to the FIB state (Fig. 1(b)). As the IU progresses towards p, FIBs at intermediate hops – still pointing towards the previous location P_0 — 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 a re-rooting of the forwarding tree originally pointing towards P_0, to the new producer location P_1 (Fig. 1(e)). Looking at the different connected regions, we see that the algorithm proceeds by extending the newly connected subtree (represented as a red cloud, initially the node where the producer is attached and its predecessors) by propagating the IU. At every step, an additional router and its predecessors are included in the connected subtree. We will analyze the properties of such a process in Sec.IV.

2) Reconciliation of concurrent updates: It may happen that IU-traversed routers have different sequence numbers associated to the same IU prefix as a result of partial and concurrent updates. This happens for instance in Figure 1(f), when the producer has moved successively to P_1 and then to P_2 before the first update was over. Both updates propagate concurrently as shown before, 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)).

D. Map-Me Enhancements: Notifications and Discovery

IU propagation in the data plane considerably helps in accelerating forwarding state re-convergence w.r.t. global routing (GR) or resolution-based (RB) approaches operating at routing plane and to 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. To prevent such losses, previous work in the Anchor-Less category has suggested the buffering of Interests at previous producer location [14]. However, such a solution is not suitable for applications with stringent latency requirements (e.g. realtime) 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, due to the wireless medium. To alleviate such issues, we introduce two separate enhancements to *MAP-Me* update protocol, namely *(i)* a *notification* mechanism for frequent, yet lightweight, signaling of producer movements to the network and *(ii)* a scoped *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 in every encountered PoA. It differs from IU only because it doesn't propagate further than the PoA and it carries a special identification flag. Both IU and IN share the same sequence number (producers indistinctly update it for every sent message) and follow the same FIB lookup and update processes. In contrast, the trace left by INs in TFIB consists of the list of next hop faces present in FIB before IN reception. This is required to avoid disruption in the communication once the producer leaves the BS: since the BS does not propagate INs, it has no way to get informed of future producer locations. All traffic is directed to the former PoA where we rely on a discovery process for reaching the producer.

It is worth observing that updates and notifications serve the same purpose of informing the network of a producer movement. However, IU process has higher signaling cost than that of IN due to message propagation. More than simply allowing an efficient handover mechanism minimizing disconnectivity of the producer, the combination of both IU and IN allows controlling the trade-off between protocol reactivity and stability of forwarding re-convergence.

2) Discovery: The extension of *MAP-Me* with notifications relies on low-latency direct links among the neighboring PoAs, which are typical of mobile deployments[2] to enable 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, labeled with the latest sequence number stored in FIB (to avoid loops), and broadcasted at one hop to all neighbors along the dedicated links (see Algorithm 3).

Once received, if there is a match with older or equal sequence number information in FIB, the "discovery" Interest is discarded. Otherwise, it is either forwarded on the available output faces to eventually reach the producer or broadcasted again at one hop along the dedicated links in the presence of a matching rule in TFIB left by an IN (characterized by a ø retransmission timer). The latter is the case of consumer Interest reaching a PoA that had previously been updated by an IN.

As further shown in Sec. V, such a mechanism is important to preserve the performance of flows in progress, especially

[2]This is for instance the case of X2 links in LTE, but it also common in multi-access WiFi networks where access points are logically under the same controller.

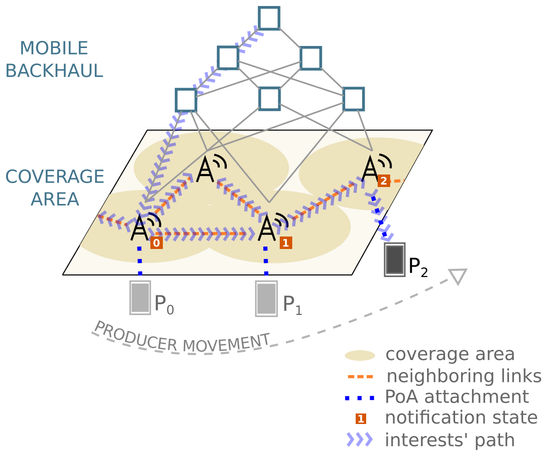


Fig. 2: Use of interest notifications, and discovery process example.

when latency-sensitive. In the rest of the paper, we evaluate a combined update/notification and discovery approach consisting of sending an IN immediately after an attachment and an update 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 to distinguish the gains due to different protocol components.

3) Illustration of combined Notifications and Discovery: Figure 2 illustrates the combined use of notifications and discovery in a mobile access network where the different base stations form the leaves of a fat tree. Neighboring base stations (for which coverage areas overlap) are assumed to be interconnected, as shown by the orange dotted lines.

The content producer, initially in position P_0 has moved successively to P_1 and P_2, sending Interest Notifications 1 and 2. Consumer interests (here originating from the root of the tree) are forwarded using FIB information synchronized with the initial state of the producer and will thus reach the initial PoA. As the producer has moved and face has been destroyed, no valid next-hop face information can be found into the FIB and the interest enters discovery mode: it is tagged with sequence number 0 as found in the FIB, and broadcasted to neighbors. The base station that has received IN with sequence number 1 has fresher information, but no valid forwarding information either; it will thus iterate the discovery. Other base stations will just discard the interest since they have no fresher information about the position of the producer. The process continues until the discovery interest reaches the current PoA – which has valid face information – and finally the producer. Data packets can flow back using PIT entries as usual.

E. MapMe messages, state and algorithms

We summarize in this section the changes to a regular CCN/NDN architecture that are needed to implement MapMe, integrating both the IN and IU procedure. As explained before, the heuristic arbitrating between sending an IN and an IU is based on the timescale parameter T_U. Setting $T_U = 0$ allows the use of MapMe with discovery but no notifications. An

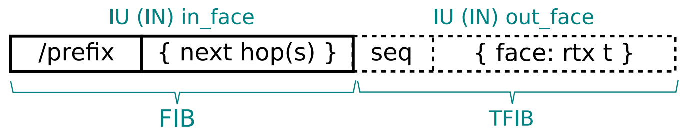


Fig. 3: *MAP-Me* FIB/TFIB description.

additional parameter allows us to further disable discovery (so-called *MAP-Me-IU*).

1) Messages: We introduce two new optional fields in the interest header:

- a sequence number: used to handle concurrent updates and prevent forwarding loops during signaling, and to control discovery interests during consumer interest propagation;
- a special interest type: allowing us to specify four types of messages, Interest Updates (IU) and Notifications (IN), as well as their associated acknowledgement messages (IU_{ack} and IN_{ack}). Those flags are recognized by the forwarding pipeline to trigger special treatment for those interest packets.

2) State: FIB entries are enriched with a sequence number, set to 0 by a routing protocol, and updated according to the number carried by special interests. Temporary data about not-yet-acknowledged signalization is stored in what we denote as *Temporary FIB*, TFIB, to ensure the reliability and robustness of the process. Each TFIB entry is composed of a list of $(face, timer)$ couples corresponding to the faces where IU have been sent but not yet acknowledged and associated to a retransmission timer (ø in the case of IN). FIB/TFIB structure is sketched in Fig.3.

We remark that the update mechanism (beyond name lookup and forwarding) is a constant delay operation at each router/hop, which can be performed at the packet transmission timescale.

3) Dynamic face management: On the producer side, mobility and subsequent layer 2 attachment are reflected at the CCN/NDN layer by a change in the face table. For instance, a new face is created and activated upon attachment to a new base station. This signal is equivalent to a router receiving fresh update information, in which it updates the sequence number and next hop in the FIB, and triggers the reliable transmission of a special interest (IN or IU depending on the implemented heuristic) for every served prefix (we currently detect them by inspecting the FIB for local application faces).

The `SendReliably()` function (defined in Alg. 1) stores timers in the TFIB section of the FIB entry to ensure reliable delivery of IUs. The notation T<*n*, *s*> means that we are crafting an interest packet of type T, with name *n* and sequence number *s*. We assume the possibility to schedule events (function calls) ahead of time, to implement retransmission if an acknowledgement has not been received within τ seconds of the packet transmission.

Upon face removal, we need to restore faces in TFIB with ø timers. It might otherwise happen that a FIB entry without next-hop faces is removed, thus losing TFIB state, and disrupting global connectivity.

4) Forwarding pipeline: The main change lies within the forwarding pipeline where we need to add hooks for process-

Algorithm 1: SendReliably(Faces F, Type T, FibEntry ϵ)

```
foreach f ∈ F do
    f.Send(T<ϵ.name, ϵ.seq>)
    ϵ.TFIB[f] = Schedule(τ, SendReliably, f, T, ϵ)
end
```

ing special interests, as shown in Algorithm 2.

Algorithm 2: ForwardSpecialInterest(SpecialInterest SI, Ingress face F)

```
CheckValidity()
▹Retrieve the FIB entry associated to the prefix
ϵ, T ← FIB.LongestPrefixMatch(SI.name)
if SI.seq ≥ ϵ.seq then
    ▹Send ack back
    s ← ϵ.seq
    ϵ.seq ← SI.seq
    SendReliably(F, SI.type+ack, ϵ)
    if SI.seq > s then
        ϵ.NextHops = []
        if SI.type = IU then
            ▹Forward the IU following FIB entry
            SendReliably(ϵ.NextHops, SI.type, ϵ)
    end
    ▹Cancel an eventual retransmission timer of new next
    hop
    if F ∈ ϵ.TFIB then
        ϵ.TFIB[F] = ∅
    end
    ϵ.NextHops.Add(F)
else
    ▹Send corrected updated backwards reliably
    SI.seq = ϵ.seq
    SendReliably(F, SI, ϵ)
end
```

The forwarding of regular interests is mostly unaffected, only to add support for discovery interests (see Algorithm 3).

Algorithm 3: InterestForward(Interest I, Origin face F)

```
▹Regular CS and PIT lookup
ϵ ← FIB.LongestPrefixMatch(I.name)
if ϵ = ∅ then
    return
▹Discovery interests: discard if no progression
if I.seq != ∅ then
    if I.seq ≥ ϵ.seq then
        return
    end
    I.seq ← ∅
end
if hasValidFace(ϵ.NextHops) or DiscoveryDisabled then
    ForwardingStrategy.process(I, ϵ)
else
    I.seq ← ϵ.seq
    SendToNeighbors(I)
end
```

F. Security considerations

As all mobility solutions, *MAP-Me* affects the forwarding of user traffic and its updates have to be secured. An in-depth study will be the subject of future work, and we restrict our ambition here to demonstrate how the existing solutions from the literature can provide an equivalent security as the current approaches used today in operational networks. It is indeed worth noticing that although our approach seems disruptive, it shares many similarities with the existing macro or micro-mobility solutions. In Anchor-Based approaches for instance (say M-IPv6), it is also up to the producer to issue registration requests to an agent to update the binding between the foreign and the care-of-address (tunnels), which also affects end-to-end forwarding.

The general approach is to distinguish a boostrap procedure when a node enters the domain, in which we typically use PKI-based schemes to do comprehensive vetting of the users' authorizations. It is then possible to use a secure token to quickly authorize subsequent calls from a user, e.g. by using an HMAC function.

Among the many variations that can be found in the literature, one such example is given by the authors of Cellular IP [4] in a distributed context that is close enough to *MAP-Me* to be readily applicable. We limit our description here to the bare mechanism, which can be further secured through the use of random identifiers in HMAC computation.

Assume that all infrastructure nodes share a symmetric key κ. Upon bootstrap, a producer willing to announce prefix p will receive a derived key $\kappa_U = \text{HMAC}_\kappa(p)$ along with random number r.

While all network nodes can compute such a key from the information given by the producer, it won't allow the user to retrieve the network key. Signaling message can then be augmented with a registration token $R = \text{HMAC}_{\kappa_U}(p \| seq)$, where $\|$ means concatenation. R can be verified by all nodes to prove that the user had indeed been authorized during a previous bootstrap. Indeed, all nodes can compute K_U then R based on the information provided by the producer in the special interest (while the producer cannot of course get back the network key).

IV. 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 analysis reports the formal proofs only in case of single-path routing.
The extension to multipath routing is straightforward by using directed acyclic graphs instead of trees.

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. We observe that at any such instant, as illustrated 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 global routing guarantees the existence of a spanning tree (SP) rooted in the original location P_0 at the beginning of the mobility process. The tree is not required to be a minimum SP or a shortest-path tree. About the completion of the update process after a given 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* $2\left(\max_{0\le j<k}(|P_k - P_j| - 1)\right)$*;*

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 more recent 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. $2\left(\max_{0\le j<k}(|P_k - P_j| - 1)\right)$. 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 have merged into the island created by IU_m. With regard to the properties of an island, we can state the following.

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 spanning tree is a rooted directed tree in P_0 giving the routes to reach all nodes in the network. *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 results in fully merging 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 start our evaluations by numerical graph computations of the average path stretch realized by idealized versions of AB, TB and *MAP-Me-IU* , meaning that there is no consideration of control protocols nor signalization propagation. The complete *MAP-Me* algorithm will be later considered in simulation in Section V). We consider the 10 ISP topologies extracted from the Rocketfuel datasets [3], by keeping the largest connected component of the graph. A summary of extracted (and original) graph properties is presented in Table I. We have developed a graph simulator to numerically compute through Monte-Carlo simulations the performance of different mobility proposals after N jumps under uniform and random waypoint mobility models. As done in previous work, the consumer, producer and anchor are chosen randomly in the ISP graph, and the mobile nodes move respectively by jumping to a nodes selected uniformly at random,random between nodes, and by following the shortest path towards a randomly selected waypoint.

Same with graph topologies from the evaluation section ?

We extend this way the basic methodology applied in previous work, namely [29], [34][4].

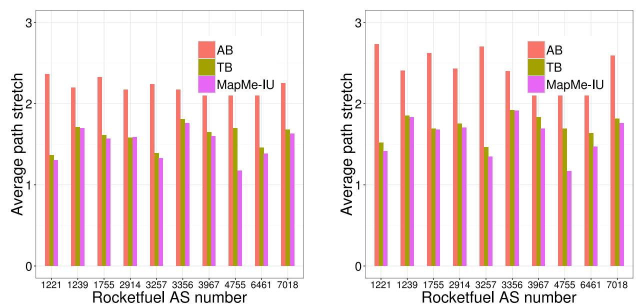
Figure 4 shows the computed path stretch for all mobility models under the two mobility models we just introduced. As expected, both *MAP-Me-IU* and TB show significant improvements over AB, with a slight advantage for *MAP-Me-IU* in most cases, which can be sometimes significant (4755). Correlations in mobility in closeby base stations (due to the

[3]http://research.cs.washington.edu/networking/rocketfuel/

[4]From the published number of nodes and links, we remark that the authors either did not use the same input data, or processed the graph in an undocumented way, which prevents us to reproduce their exact results. We still obtain similar qualitative conclusions.

AS	Name	$\|V\|$	$\|E\|$	comments
1221	Telstra (AU)	318 (378)	758 (779)	
1239	Sprintlink (US)	604 (700)	2268 (2268)	
1755	Ebone (EU)	172 (172)	381 (381)	
2914	Verio (US)	960 (1013)	2821 (2832)	
3257	Tiscali (EU)	240 (248)	404 (405)	
3356	Level3 (US)	624 (625)	5298 (5298)	
3967	Exodus (US)	201 (215)	434 (443)	
4755	VSNL (India)	11 (12)	12 (12)	
6461	Abovenet (US)	182 (202)	294 (310)	
7018	AT&T (US)	631 (656)	2078 (2078)	

TABLE I: Summary of Rocketfuel topologies



(a) Uniform mobility (b) Waypoint mobility

Fig. 4: Path stretch comparison over Rocketfuel topologies

waypoint model) benefit *MAP-Me-IU* even more thanks to localized traffic that does not need to go up to the anchor and back down, to eventually find a shortcut in TB.

Figures 5 and 6 shows how stretch builds up after mobility, and proves that average stretch in bounded in *MAP-Me-IU* , due to preserving the original spanning tree (the shortest path tree constructed by routing). Figure **??** represents the evolution of average *MAP-Me* path stretch over AS 1221 topology under RWP (other patterns show similar trends). We observe that path stretch stabilizes beyond 10 movements, because *MAP-Me* preserves the initial structure of the forwarding tree (it only modifies links direction). Other Rocketfuel topologies show quantitatively the same results. A comparison of the different approaches over the all 10 Rocketfuel topologies is in Fig.4(b) and 4(b). Under both uniform and RWP mobility, *MAP-Me* outperforms AB, achieving up to 55% stretch reduction, as well as TB.

V. Evaluation

A. Simulation setup

While giving good insight into the theoretical properties of the different algorithms, the previous numerical simulations do not take into account the impact of signalization (propagation delays and losses), nor packet-level artefacts impacting the performance of applications. In this section, we thus present exhaustive simulations for assessing the relative performance of mobility solutions over a wide range of input parameters.

1) Implementation of mobility solutions: To this aim, we implemented both the full *MAP-Me* version, as well as the IU-only flavour, that we denote *MAP-Me-IU* , anchor-based (AB), an example of tracing-based (TB) based on Kite ([34]), and global routing (GR) approaches in NFD within the NDNSim 2.1 framework. We leave out of the comparison resolution-based (RB) or existing AL-solutions leveraging buffering, as they fail to support latency-sensitive applications (see discussion in Sec. II).

MAP-Me and *MAP-Me-IU* are direct implementations of the mechanisms described in Section **??**.

AB is realized through specific application at the anchor and base-stations, that prepend prefixes to names for control and data plane. Prefixes of produced content are advertised by the anchor, and routing to anchor and base-stations (BS) prefixes is set-up at the beginning by routing. The anchor then maintains a mapping between each user prefix and the corresponding BS location, that is updated through signalization. The BS finally stores a mapping between the prefix and the face on which is can be reached.

TB is obtained by enabling in Kite ([34]) all optional optimizations that we have found to be mandatory, to let the protocol work in the presence of losses and frequent mobility: (i) traces are reissued right after a handover and (ii) retransmitted if an ACK is not received within 60ms, (iii) Interests from consumers make use of the trace table and the FIB also (`traceonly` flag must be unset). In-network retransmission (*pull*) is disabled for fair comparison. We also built this code from scratch as the public implementation is incompleted and targeted at the former ndnSIM v1.0.

Explain why we need all these extensions for KITE.

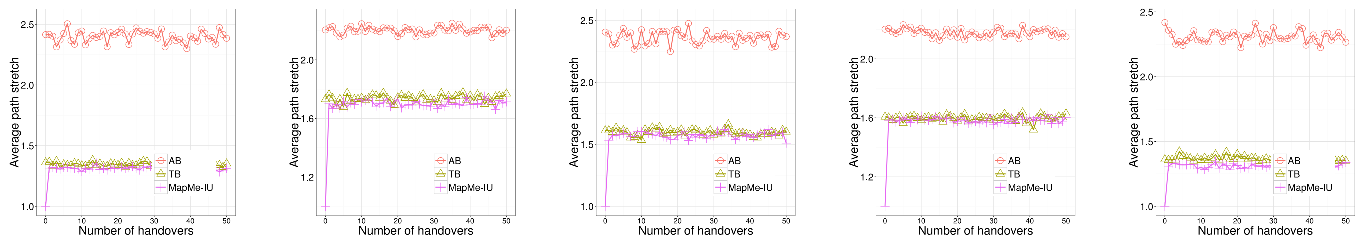
Release of code

2) Simulation parameters: This sections gathers the extensive simulation evalutions we have performed to convince about *MAP-Me* performance and its robustness to parameters such as topology, mobility model, radio conditions and workload. In particular, it gives interesting insight into the impact for an ISP deploying various mobility solutions in term of management and dimensioning.

Graph	parameters	$\|V\|$	$\|E\|$	comments
tree	-	-	-	?
fat-tree	-	-	-	?
cycle	-	30	30	?
grid-2d	-	100	180	?
hypercube	-	128	448	?
expander	-	100	400	?
regular	-	100	150	?
erdos-renyi	-	100	564	?
watts-strogatz	-	100	200	?
small-world	-	100	437	(original directed)
barabasi-albert	-	100	384	?
geant	-	-	-	?
abilene	-	-	-	?
dtelekom	-	-	-	?

Our simulation ultimately aim at realism in order to quantify the added overhead and performance impact of the mobility management solution, in top of layer 2 interfaces. Still, the mix of all parameters might make the interpretation of results difficult and might not underling the relative contribution of various design choices to the final results. For that reason, we progressively build up our environment from simple and artificial setups, to conclude on a streaming application deployed in a trace-based emulated vehicular environment.

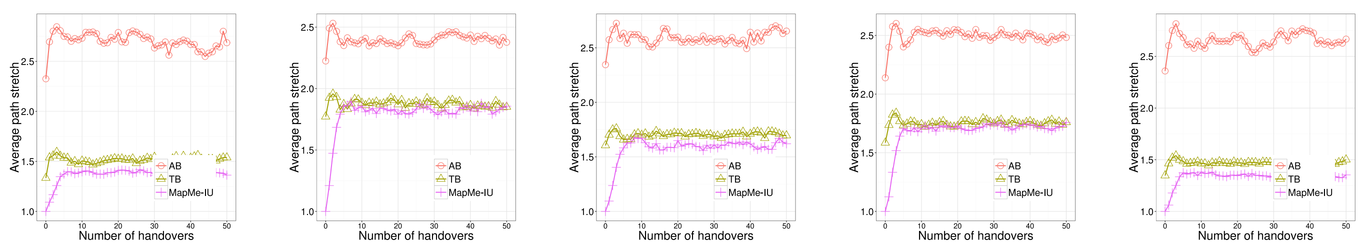
3) Topology:



(a) AS 1221 (b) AS 1239 (c) AS 1755 (d) AS 2914 (e) AS 3257

(f) AS 3356 (g) AS 3967 (h) AS 4755 (i) AS 6461 (j) AS 7018

Fig. 5: Path stretch evolution with uniform mobility model.



(a) AS 1221 (b) AS 1239 (c) AS 1755 (d) AS 2914 (e) AS 3257

(f) AS 3356 (g) AS 3967 (h) AS 4755 (i) AS 6461 (j) AS 7018

Fig. 6: Path stretch evolution with waypoint mobility model.

- Not typical of access networks.
- No geographic mapping.
- Cannot use realistic mobility models, jump process instead.
- No IN.

4) Mobility: Since performance of mobility protocols depends on user mobility patterns, we explore various mobility models that are relevant and available in ns3 to test performance of mapme and other mobility protocols. In particular, the following mobility models are considered in our simulation:

- random direction 2D model: user uniformly and randomly pick a direction and walk until they hit boundary of moving area. Then it picks up a new direction again.
- random gaussion walk model: At each timestep, a new velocity, direction, and pitch angle are generated based upon the previous value. The trajectory of user is smooth without sudden change of direction
- random walk 2D: user uniformly and randomly pick a direction and walk a distance equal to the cell size.Then the same step repeats. This model emulates brownian motion.
- random way point model: user uniformly and randomly pick up a waypoint in moving area and move to it. Then the same step repeats.

5) Radio model: In order to show that our simulation results are not subject to a specific radio conditions, we also consider the evaluation under different wireless environments. We consider the following radio settings:

- ideal wireless channel: we use wired links to simulate an ideal wireless channel without loss, collision and with 0 layer 2 handoff latency. In order to separate out only

the impact of layer 3 mobility protocols on application performance.
- log distance model + releigh fading model:This is the model we used in the section for evaluation. It aims at simulating a more realistic radio environments.
- suburban with light of sight: This configuration based on the parameters from IUT-R p.1441 specification on radio propagation model.
- urban without light of sight: This configuration based on the parameters from IUT-R p.1441 specification on radio propagation model.

6) Workload: Streaming / elastic discussion as done in the letter

7) Baseline scenario: We cannot simulate everything…

B. Artificial evaluations

We start with articial topologies, both constructed from well-known deterministic and probabilistic graph models, and from Rocketfuel topologies as we used in Section **??**. They are summarized in Table **??**, together with their main characteristics.

Compared to the previous graph computations, these simulations take into account more parameters (like the randomness of the mobility pattern, the radio conditions), but more importantly take into account the packet level mechanisms including the propagation of signalization, and its impact on traffic.

1) Markovian Mobility: Let's first remark that those topologies are not representative of access networks, in particular the Rocketfuel datasets. They do not come with any link capacity not base station coordinates. We will thus start with the so-called *Markovian* mobility patterns. While not taking into account radio conditions nor the influence of the mobility pattern, the tend to show the network properties induced by the mobility pattern (including stretch, offloading, etc.).

We assume in this setup that all nodes can be base stations. As previously, we consider two artificial mobility processes, Uniform Jump (UJ) and Random Waypoint Jump (RWPJ), which consist in moving from one node to another immediately, after staying for an exponentially-distributed amount of time in a given node. These two behaviors tend respectively to mimic long-range mobility in the network, as when moving from WiFi to LTE, and local mobility. These are of course extreme cases meant for illustrating the performance of our scheme only, and more realism will be provided in further sections, in particular for scenarios of interest for micro-mobility. In particular, these scenario do not consider proximity between base stations and we cannot thus use not evaluate extensions to Map-Me such as Interest Notifications.

Add figures for `markov-rwp` and `markov-rw`.

2) Markovian Mobility: In the previous simulations, a mobile node could attach to any station. In order to use use a more realistic mobility pattern including radio conditions, we pick 16 nodes uniformly at random in the graph to act as Base Stations, and map them to geographical positions in a 4x4 grid as described earlier. When needed, an anchor is also randomly chosen among the nodes in the graph.

This scenario brings use closer to an access network when base stations are geographically close, and the mobile can freely navigate among them. While the fat-tree topology seems to use the closest to real ISP topologies (hierarchy, redundancy, meshing), others are also useful to underline specificities of each mobility protocol. Depending on the shape of the graph and handovers between base stations, the mobile will sometimes undergo effect he would have while jumping between several access networks connected more deeply in the network (eg moving from WiFi to LTE). These scenario help show the impact of various graph properties of the backhaul network graph.

Wired simulations from Xuan ?

As these topologies are not specific to access network deployments, and are thus not annotated with base-station position and geographical coordinates, it is not possible for us to simulate real mobility. This section is thus a step further towards more realism from graph simulations done in the previous sections. Still, they are relying on an emulated mobility process and ideal radio conditions. Those two parameters will be considered in the next section focused more on real access deployments.

C. Impact of topology, mobility pattern and radio model

1) Impact of topology: As shown in figure 17, the path stretch and hopcount for mapme-iu and kite are similar to each other across different topologies.

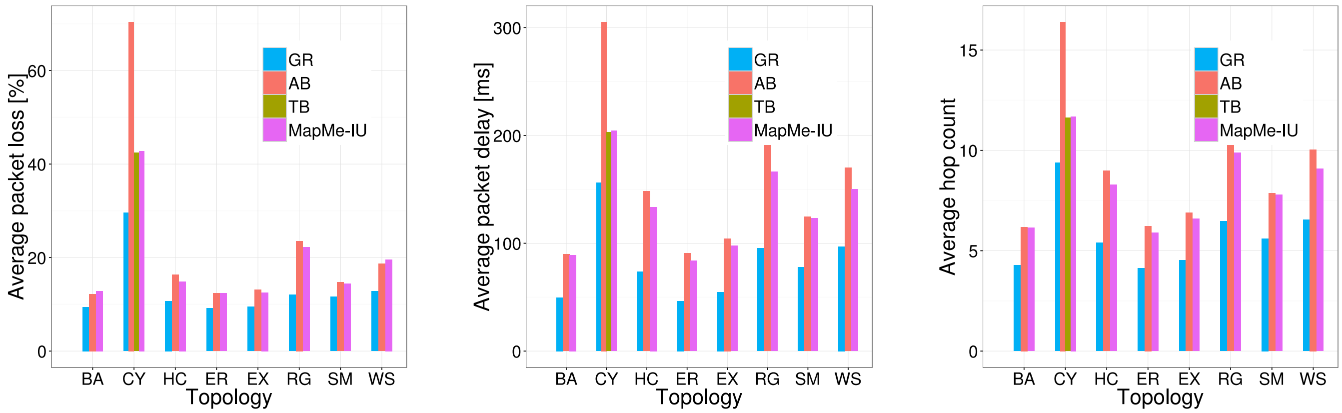
2) Impact of mobility pattern: As shown in figure15, the results under different mobility patterns are similar to each other. And mapme consistently outperforms other mobility protocols with different user mobility patterns.

3) Impact of radio model: As shown in figure16, the results under different radio settings are similar to each other. And mapme consistently outperforms other mobility protocols under different radio settings.

D. RWP mobility

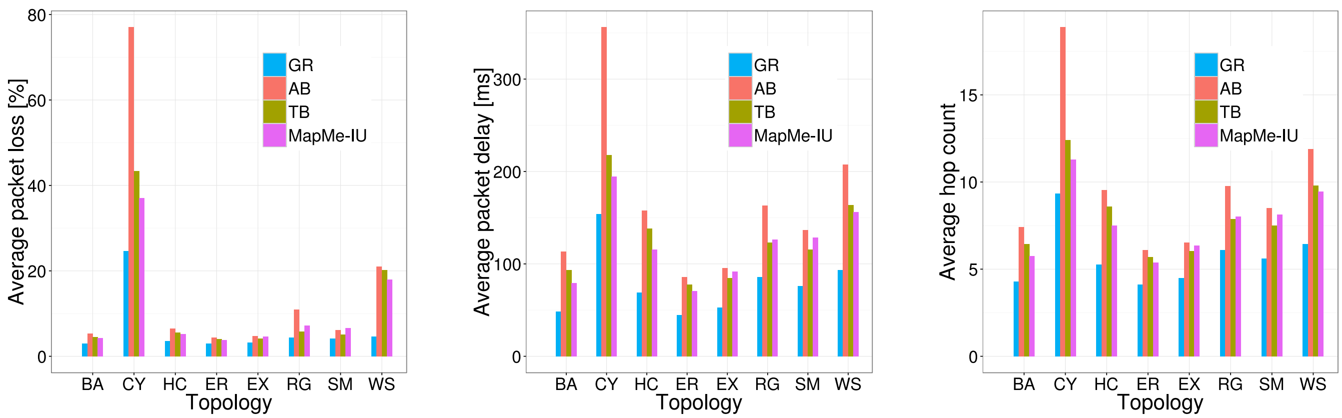
In this section, we assume RWP mobility with uniform points distributed across a squared mobility area and constant speed. To highlight *MAP-Me* benefits in the support of latency-sensitive traffic, we consider a streaming audio/video application, characterized by a CBR rate of 1Mbps. In the following, we report statistics about user performance and network cost.

1) User performance: In Fig.9(a)-9(b), we show two performance indicators for latency-sensitive traffic, average packet loss and delay, in case of $N = 5$ consumer/producer pairs as a function of mobile speed (from pedestrian, i.e. 1m/s or 3km/h to vehicular, i.e. 15m/s or 54km/h). 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



(a) (b) (c)

Fig. 7: Uniform markovian mobility on various topologies

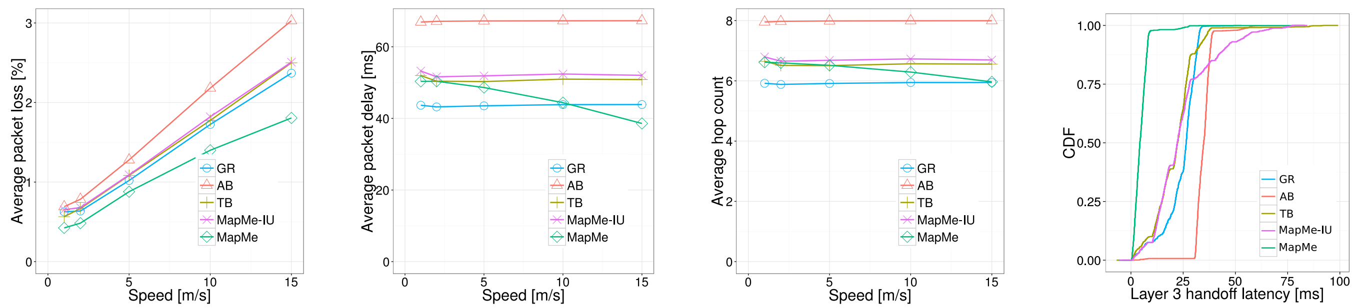


(a) (b) (c)

Fig. 8: RWP mobility with grid arrangement of BS randomly connected to various topologies

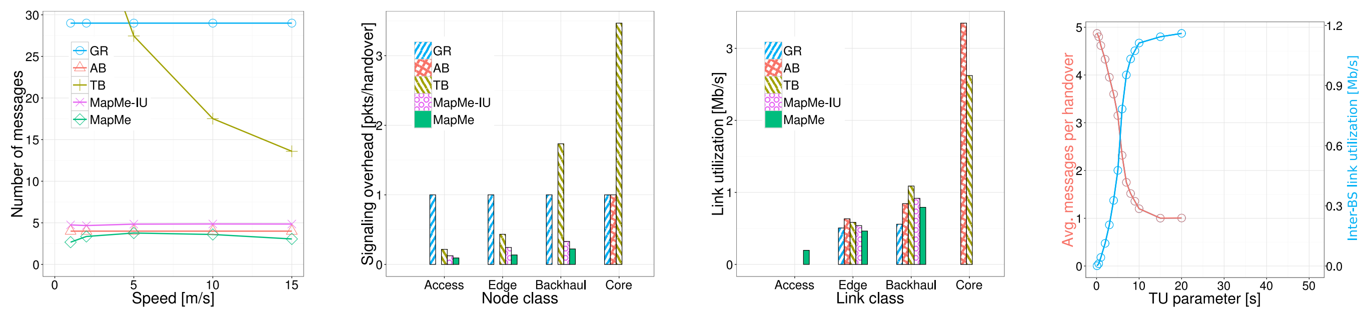
performs because of worse update completion time and path stretch. The experienced average packet delay in Fig.9(b) is a consequence of the path stretch of different approaches: high for AB, medium for TB or *MAP-Me-IU*, low for GR.*MAP-Me* achieves better performance especially at high speed when the 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.9(c), and from the L3 hand-off latency, i.e. the time required for L3 reconnection after a handover, see Fig.9(d). The L3 hand-off latency illustrates the reactivity of the mobility-management protocol and highlights the significant improvement brought by *MAP-Me*, which reduces latency to zero. 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 the number of 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.9(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.10(a), and the volume of signaling messages per handover to be processed by routers at different positions in the network, in Fig.10(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.**??**. 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.9(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



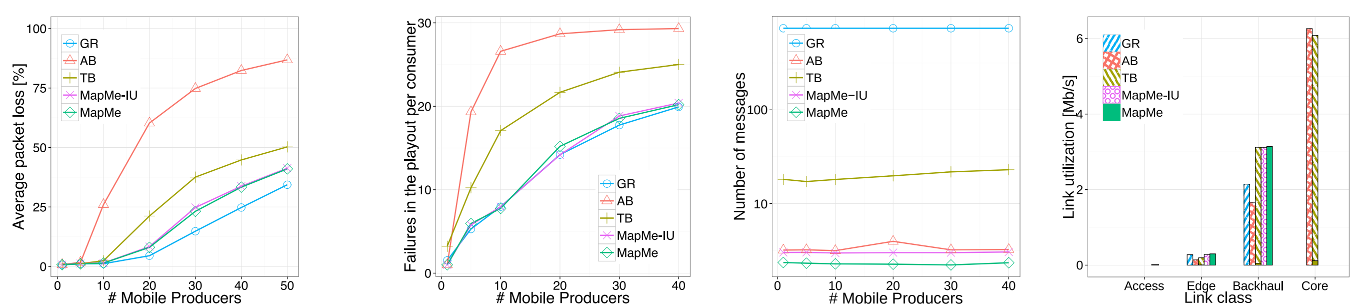
(a) (b) (c) (d)

Fig. 9: User performance: packet loss (a), delay (b), and hop count (c); CDF L3 hand-off latency (d).



(a) (b) (c) (d)

Fig. 10: 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. 11: User performance: CBR average packet loss (a), Periscope playout failures (b). Network Cost: CBR overhead (c) and Periscope link utilization (d).

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.10(c)[5]). Finally, we report about *MAP-Me* sensitivity to parametrization, i.e. the impact of TU settings. In Fig.10(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).

E. *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 [25] has in the area, i.e. we dislocate 729 WiFi APs, with the same wireless settings as in the previous experiments, connected to the Internet through the fat-tree topology in Fig.**??**.

Mobility: We generate realistic vehicular mobility patterns using SUMO [17], with maximum car speed set according to road speed limits[6]. 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

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

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

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.

Applications: Two types of applications are considered: in the first set of simulations, the previously detailed streaming application is characterized by a rate of 1Mbit/s. In the second set, a pseudo real-time video streaming application, reproducing the popular application Periscope [22] 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. The Interest sending rate is regulated by TCP-like congestion control and slow–start mechanism. Retransmission time for expired Interests is defined by a Mean–Deviation RTT estimator [11]. 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. 11(a) shows the average packet loss for *MAP-Me* and other protocols, 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 the performance, because links are getting congested and routers start to lose packets. However, as shown in Fig. 11(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.

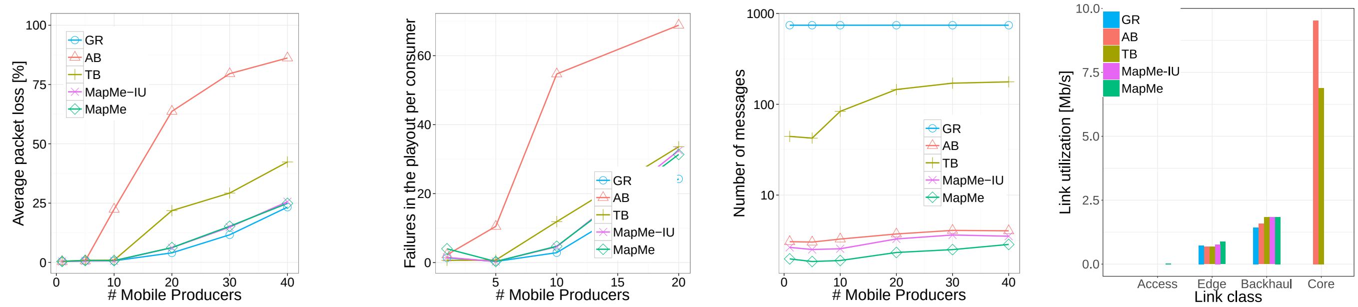
User Satisfaction: We evaluate user satisfaction by analyzing the number of failures in the play-out of the video stream for the pseudo real-time video streaming (Periscope-like). Fig. 11(b) shows the number of failures in the video play-out that each consumer encounters in 4 min. As in the CBR case, when the number of mobile producers increases, the performance of the system degrades. Similar to what is observed in CBR case before, 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. It is worth noticing that the application runs a classic window-based congestion control with no video rate adaptation and no specific mobility loss recovery mechanisms (all packet losses are recovered based on timer expiration at the consumer). The design of such schemes and the analysis of the interaction with mobility-management protocols is out of the scope of this paper and left for future work.

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.11(c) reports the overhead, computed as the number of messages exchanged in the network at each handoff, whereas Fig.11(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.

3) Effects of mobility: In other to evaluate the possible effects that different mobilities may have on *MAP-Me*, we repeated the same analysis using synthetic traces of pedestrian mobility (: description of the mobility; here or before). Similarly to Figure 11, Figure 12 shows *MAP-Me* performance in terms of CBR packet loss (Fig. 12(a)), number of failures in the Periscope-like application (Fig. 12(b)), overhead in the network (Fig. 12(c)) and link utilization (Fig. 12(d)). Comparing Figure 12 with Figure 11 (car mobility), we see that the performance and the benefits of *MAP-Me* and *MAP-Me-IU* remain similar, confirming that the type of mobility (car vs pedestrian) doesn't affect *MAP-Me* (the number of playout failures with pedestrian mobility seems higher, but only because the simulation last longer — 580 seconds against 240 seconds of the car-trace case). In addition, it must be noticed that, due to a larger interval time between two consecutive handoffs, in the case of TB the overhead is higher with pedestrian mobility (Fig. 12(c)). This is caused by keepalive messages that need to be periodically sent in the TB approach and that are not needed in *MAP-Me*.

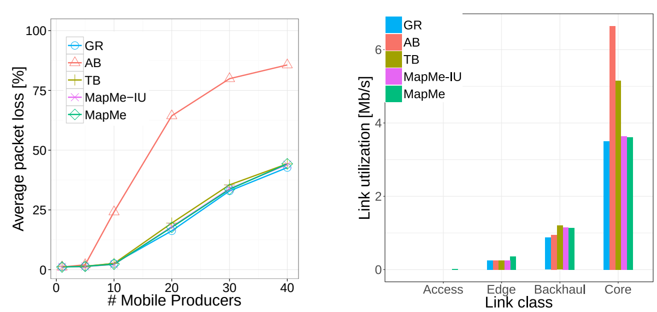
4) Effects of topology: To evaluate the effects of network topology, we simulated the aforementioned scenario (with car mobility), changing network topology: instead of a fat–tree topology, now leafs are connected to a tree-like network topology topology description? or reference to description in the paper. Figure 13(a) shows the packet loss of the CBR application: while AB still remains the worst approach, *MAP-Me* and TB behave similarly. Indeed, in a simple tree–topology, IUs cannot use the "shortcuts" provided by the fat–tree topology, but need to go more deep in the network to reach the sub–tree where the producer was previously connected. This results in an increase load of links close to the core of the network, which may experience congestion, similarly to TB approaches, where the area of the network close to the RV point is usually at higher risk of congestion.

results discussion– check final results about link load; Figure 13(b) instead shows the traffic distribution over the network. In contrast with the fat-tree topology results shown in Figure 11(d), in the tree topology the core is subject to higher load with all the protocols, due to the absence of some links interconnecting the internal nodes of the network. It must be noticed, however, that while *MAP-Me* and *MAP-Me-IU* approaches have a behavior similar to the optimal GR, TB and in particular AB, still show a higher load on the core of the network.



(a) (b) (c) (d)

Fig. 12: User performance with pedestrian mobility: CBR average packet loss (a), Periscope playout failures (b). Network Cost: CBR overhead (c) and Periscope link utilization (d).



(a) (b)

Fig. 13: User performance with tree–topology: CBR average packet loss(a) and link utilization (b).

VI. *MAP-Me* AND ROUTING

The core of forwarding decisions — the FIB — is affected by both *MAP-Me* and routing protocols. To avoid inconsistencies in the network, the aforementioned mechanisms must cooperate. In this section, we discuss how *MAP-Me* may coexist and interact with a hypothetical routing protocol, benefiting from route updates to shorten path stretch while providing producer reachability after hand-offs.

By design, *MAP-Me* takes the burden of avoiding inconsistencies in FIBs, leaving the routing protocol unaware of the mobility-management process.

A. Proposed solution

To guarantee the coexistence of *MAP-Me* and routing protocol, we propose a minor modification to the original *MAP-Me* design described in Section III. The new mechanism focuses on the interaction between Interest Update and routing, which may cause inconsistencies in FIBs if not properly handled.

The following assumptions about routing protocols have been made:

- *Link-state based*: to have a better view of routing convergence state in each router, the routing protocol must be link-state based.
- *Routing state sequence number:* the routing protocol must provide a sequence number indicating local version of the routing state. This is used to chronologically sort routing and *MAP-Me* updates. Such a sequence number may be implemented for instance as a counter of received non duplicate routing messages (LSA). In the rest of the paper, we refer to such information as **Rseq**.
- An instance of such a routing protocol is running on the mobile producer, which only exchanges LSAs with the network to update its Rseq rather than computing routes to producer's prefixes.

The modifications to the original *MAP-Me* design are summarized as follows: a new field in the IU packet is introduced to carry the Rseq information. Before sending IU, the producer gets its local Rseq from the routing state and copies it into Rseq field of IU. Note that this field is different from the sequence number used for ordering IU/IN messages. It is used indeed to chronologically sort routing and *MAP-Me*'s updates.

For each router receiving IU, before processing the packet as described in Section III, the Rseq carried by the IU is compared with the sequence number of the local routing state. Depending on the comparison result, three cases can be distinguished:

Case 1: IU's Rseq = local Rseq. This case falls into the normal operational mode of *MAP-Me*. Routing and *MAP-Me* are already in sync; thus, the IU can be processed as previously defined in Section III.

Case 2: IU's Rseq > local Rseq. The receiving node has not been updated yet by the routing protocol. The IU is queued until the node receives a routing update – until the local Rseq is incremented. Afterwards, the Rseq carried by the IU is compared again with the local sequence number, falling once more in one of the three cases listed here.

Case 3: IU's Rseq < local Rseq. The IU is discarded. Indeed, in this case, the nodes in IU's path have outdated FIB information (from routing point of view), which should be overwritten by routing when convergence is reached on these nodes. Thus, it is meaningless to propagate the IU any further.

Finally, additional steps must be taken to update FIBs with the most recent information about the mobile producer after a routing update. Indeed, whenever a routing update occurs, it suppresses IU propagation (i.e. aforementioned case 3) and overrides FIBs, potentially with stale information about the producer's location – producer may have moved after the routing update process started. For this reason, when the routing update reaches the mobile producer, the latter generates a new IU, which restores the most recent information about the producer's location.

B. Correctness discussion

In the following section, we discuss how the proposed modification allows *MAP-Me* and routing protocols to cooperate in a consistent manner and avoid conflicts when "simultaneous" FIB updates happen. For simplicity, here we assume that the producer only sends one IU during routing convergent phase. It is, however, straightforward to generalize the discussion by verifying the path followed by each IU.

In the rest of the section, we use the term "routing convergence on a node" when the router receives all routing LSAs and updates its FIB to the latest. We use instead "global routing convergence" when every node in the network has the same local Rseq.

The following three situations may occur: if the network reaches a global routing convergence state, the system falls into the aforementioned "case 1" and IU is propagated as in the original *MAP-Me* design. If instead routing converges at the producer but not globally (producer has the highest Rseq), then IUs arriving at PoA either (1) propagate through the PoA if the node has also converged; or (2) wait at the PoA until the node converges, which then processes and properly forwards the update. The mechanism repeats across the entire network, until the IU reaches its destination, guaranteeing that IUs are always propagated according to the latest FIB and reach the old producer's location. A forwarding tree rooted at the producer's new location is then established once the routing converges globally.

Finally, if the network has not reached a global convergence yet and the producer has not converged either, the IU propagates until it reaches a converged node, where it is dropped because it has a lower Rseq ("case 3"). In contrast, routing updates propagate to the entire network, overriding FIBs that have been affected by the IU – removing any trace of the IU from the network. Finally, when the producer converges, it issues a new IU (with the updated Rseq), which restores producer reachability in the case the latest routing update was based on stale information about the producer's location.

While *MAP-Me* reduces the disconnectivity time of a producer after a hand-off, in the occurrence of a routing update, the proposed modification only guarantees producer reachability as quickly as routing converges. Thus, the resulting performance of the system is suboptimal in terms of forwarding plane convergence time when a routing update occurs. We leave the jointly optimal design of a routing protocol and *MAP-Me* as future work.

C. Evaluation

In the following section, we present a preliminary evaluation of the proposed modification. In particular, we analyze the effects of routing updates frequency on system performance.

Simulation settings: We set up a simulation scenario where one pair of mobile consumer/producer communicates with each other using the aforementioned modification of *MAP-Me*. The consumer runs CBR streaming application at 1Mbps rate and both nodes' speed is 10m/s. The same fat-tree topology shown in Figure **??** and the same 802.11 radio channel settings described in section III are also used here.

In the simulation, in addition to run *MAP-Me* to maintain producer reachability, we also periodically force the producer to announce its name prefix from its new PoA, to trigger a global routing update. We change producer prefix advertisement frequency to show the trade-off between route optimization and packet loss.

It is obvious that the routing convergence time may play a key factor in the performance of this proposal. While it is generally known that link-state IGP convergence time is in the order of several seconds, [8] has demonstrated that link-state IGP routing protocol can even achieve sub-second convergence time for large ISP networks by leveraging techniques like fast flooding and incremental FIB updates. Without loss of generality, we evaluate *MAP-Me* varying routing convergence time from 600 ms to 6000 s.

Results: Figure 14(b) shows how routing update frequency affects path stretch (or hop counts as in the plot). At every routing update, the path to the producer's latest location is optimized, thus, as shown in the figure, increasing routing updates has beneficial effects in reducing the path stretch.

Figure 14(a) reports instead the effects of routing updates on packet loss (during convergence phase, producer reachability is not guaranteed in case of a handover). The figure shows that increasing routing updates frequency increases packet loss, due to a larger amount of time without reachability guarantees. This may suggest that the decision of triggering a routing update may benefit from an integration with producer mobility prediction: producer can safely advertise its new location if it will remain attached to the same PoA longer than the convergence time.

Figure 14(a) also shows that when routing convergence time is short (less than 1.2 s), there is almost no increase in packet loss whatever the routing update interval is. Indeed, when convergence time is short, most likely, the producer does not make any handover.

In Figure 14(c), we see that network update overhead increases when reducing the update interval. This is because each routing update spreads all over the network.

In summary, we see that by reducing the update interval, we obtain a shorter path stretch at the cost of increased packet loss and network overhead. A trade-off must be made among path stretch, packet loss, and network overhead.

VII. 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

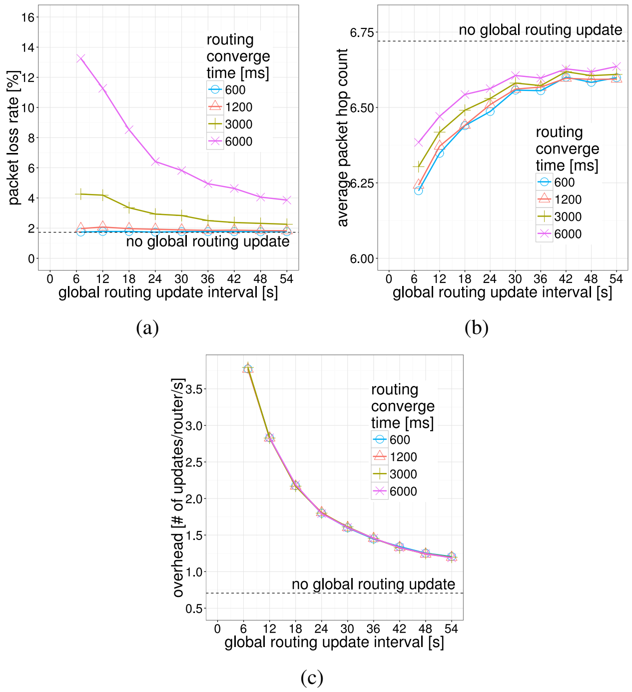


Fig. 14: *MAP-Me* and routing. Effects of routing update frequency on performance: (a) Packet loss rate. (b) Path stretch. (c) Overhead.

proven guarantees of bounded stretch and overall correctness for the forwarding update process. Second, we develop a simulation framework on top of NDNSim 2.1 integrating *MAP-Me*, anchor-based and tracing-based approaches and ideal global routing as a reference mobility model, using model-based and trace-driven consumer/producer mobility patterns. Evaluation takes 802.11n access in small cell outdoor settings and proves how wifi can support mobility using CCN/NDN in general settings.

The reported results 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] A. Afanasyev, C. Yi, L. Wang, B. Zhang, and L. Zhang. Snamp: Secure namespace mapping to scale ndn forwarding. In *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*, pages 281–286. IEEE, 2015.

[2] 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 *Proceedings of the 2008 ACM CoNEXT Conference*, page 66. ACM, 2008.

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

[4] A. T. Campbell, J. Gomez, S. Kim, A. G. Valkó, C.-Y. Wan, and Z. R. Turányi. Design, implementation, and evaluation of cellular ip. *IEEE personal communications*, 7(4):42–49, 2000.

[5] G. Carofiglio, L. Muscariello, ichele Papalini, N. Rozhnova, and X. Zeng. Leveraging icn in-network control for loss detection and recovery in wireless mobile networks. In *ACM SIGCOMM ICN'2016*, Kyoto, Japan, Sept. 2016.

[6] 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.

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

[8] 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.

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

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

[11] V. Jacobson. Congestion avoidance and control. In *ACM SIGCOMM computer communication review*, volume 18, pages 314–329. ACM, 1988.

[12] 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*.

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

[14] 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*.

[15] 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.

[16] 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. ACM, 2007.

[17] 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.

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

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

[20] 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, November 2016.

[21] 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 *Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on*, pages 467–472. IEEE, 2016.

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

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

[24] I. Seskar, K. Nagaraja, S. Nelson, and D. Raychaudhuri. Mobilityfirst future internet architecture project. In *Proceedings of the 7th Asian Internet Engineering Conference*, pages 1–3. ACM, 2011.

[25] Time Warner Cable. TWC WiFi coverage map, http://coverage.twcwifi.com/.

[26] G. Tyson, A. Mauthe, S. Kaune, P. Grace, and T. Plagemann. Juno: An adaptive delivery-centric middleware. In *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, pages 587–591. IEEE, 2012.

[27] 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.

[28] G. Tyson, N. Sastry, I. Rimac, R. Cuevas, and A. Mauthe. A survey of mobility in information-centric networks: Challenges and research directions. In *Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications*, pages 1–6. ACM, 2012.

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

[30] 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.

[31] 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.
[32] 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.
[33] Y. Zhang, A. Afanasyev, J. Burke, and L. Zhang. A survey of mobility support in named data networking. In *Proc. of IEEE INFOCOM NOM*, 2016.
[34] Y. Zhang, H. Zhang, and L. Zhang. Kite: A mobility support scheme for ndn. In *Proc. of ACM ICN Poster*, 2014.
[35] 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. He is currently a researcher at Cisco Systems in the Information-Centric Networking project. His research interests are information-centric networking, mobile networks, scheduling, 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 where he still retains the position of an Adjunct Professor. Dr. Pau designed and built the UCLA campus vehicular testbed and the UCLA/MPI urban sensing testbed. His research contributions lead to the VERGILIUS and CORNER simulation suites for mobility and propagation modeling in urban environments. More recently, Dr. Pau 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.

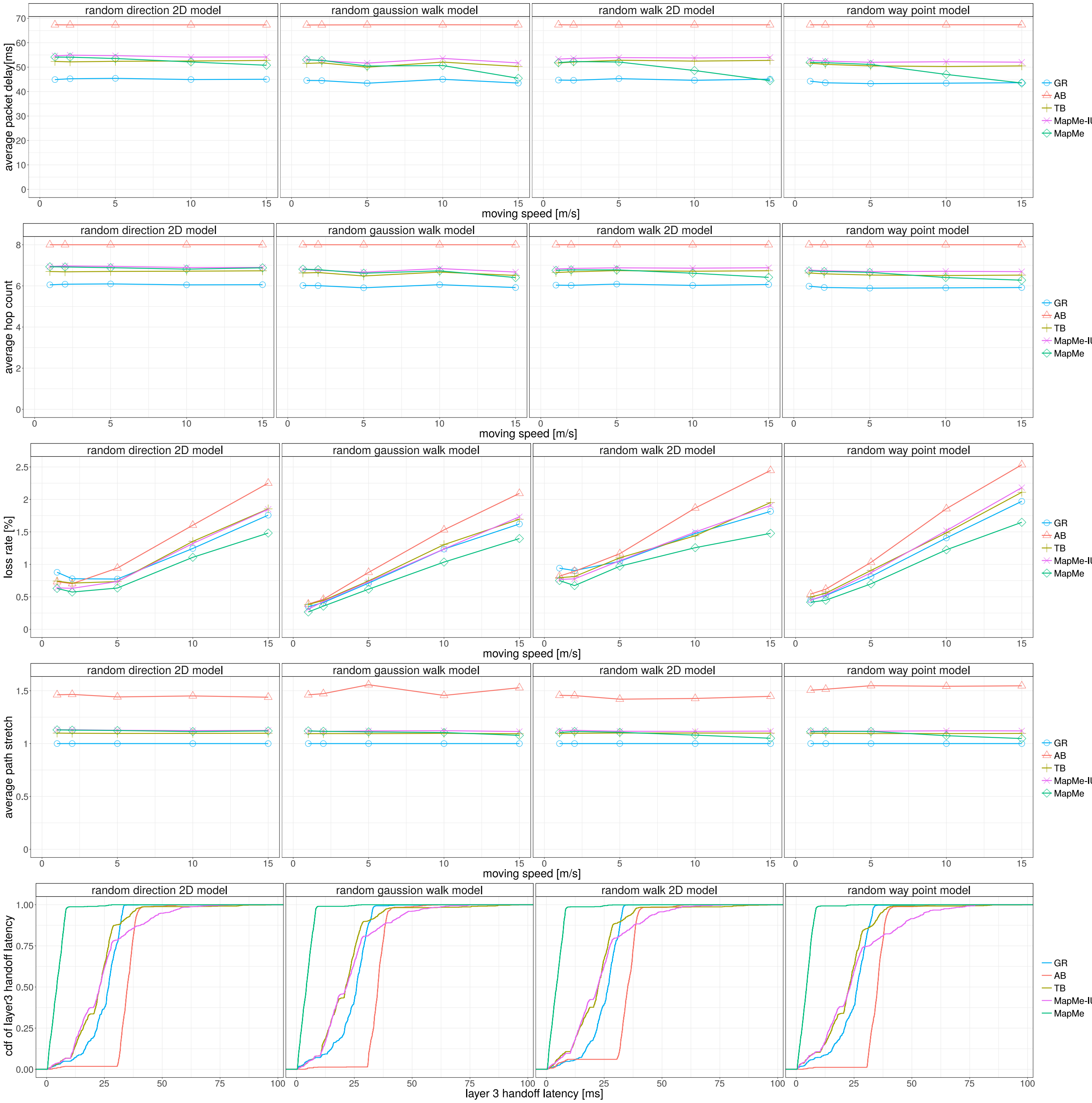


Fig. 15: simulation results for different mobility patterns

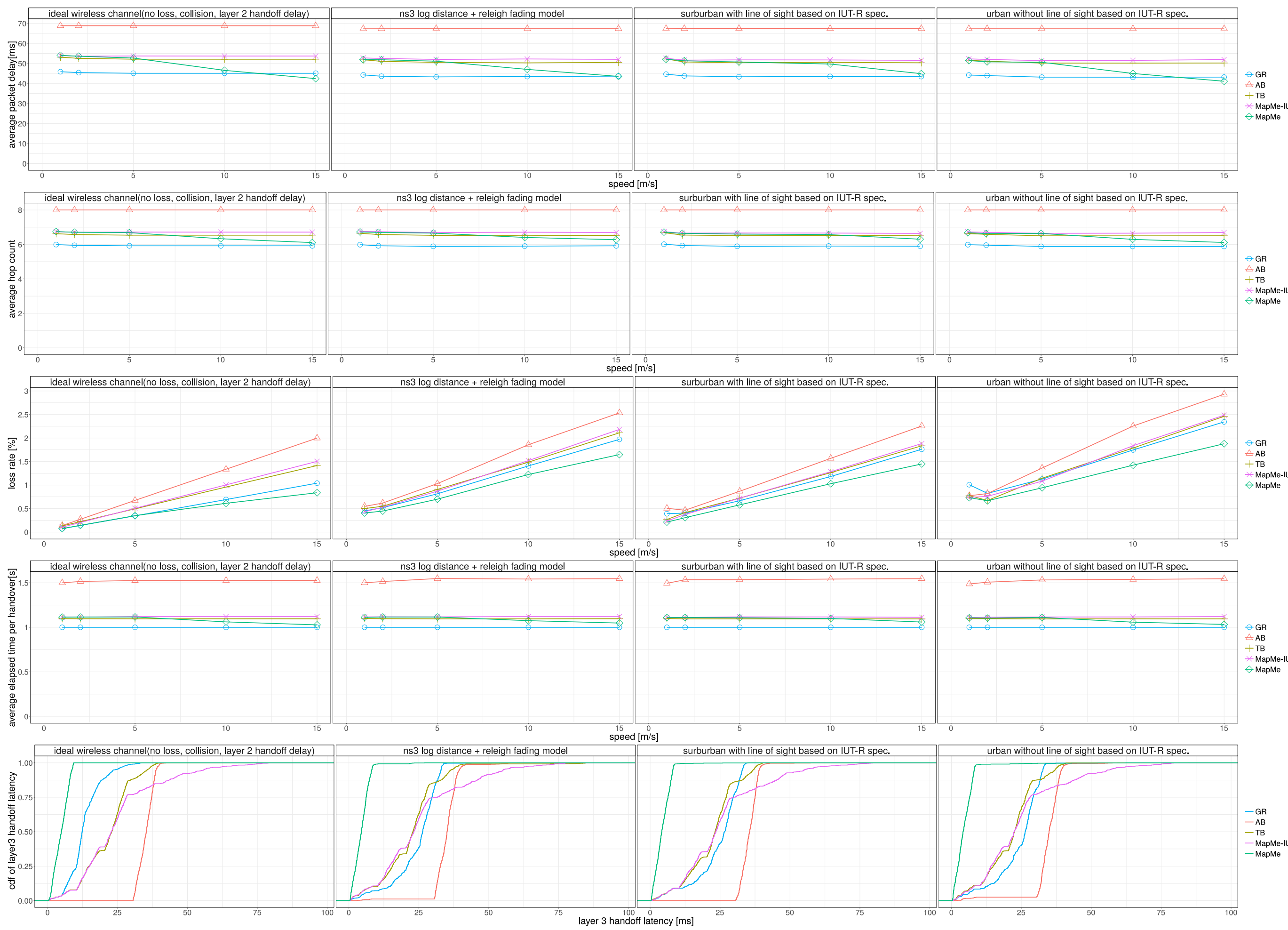


Fig. 16: simulation results for different radio settings

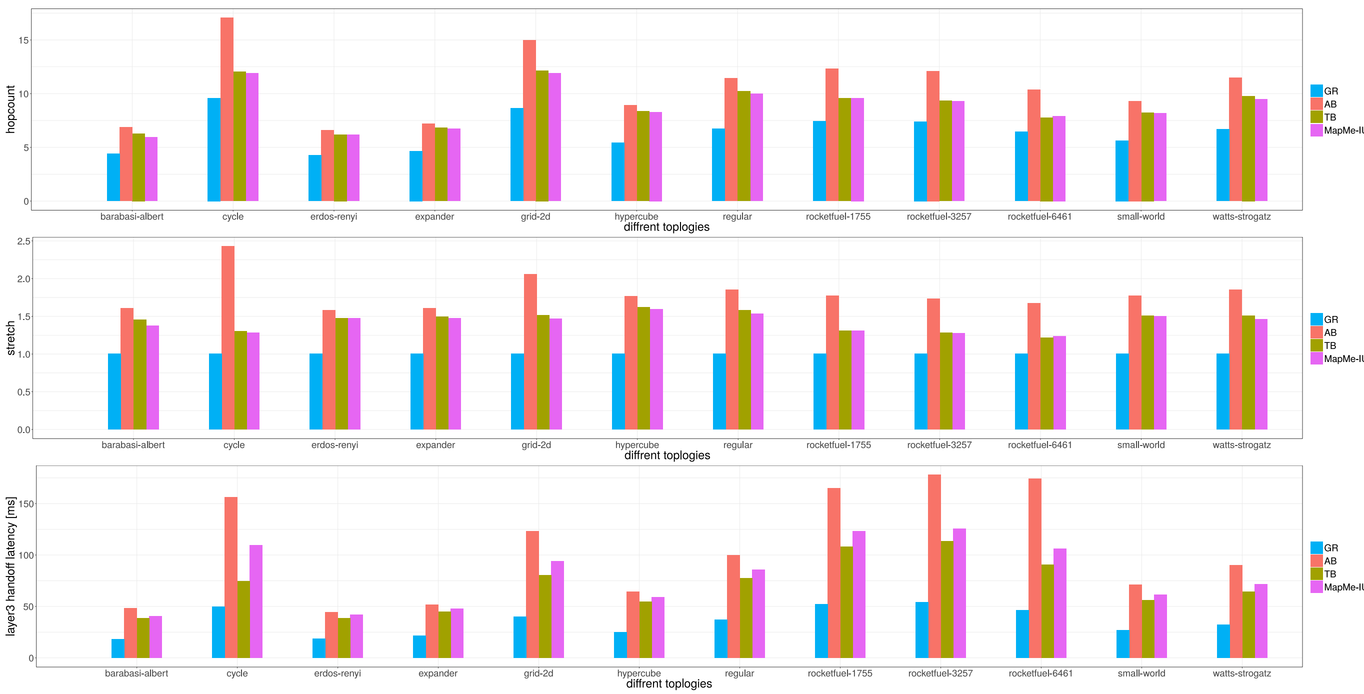


Fig. 17: simulation results for different topology settings