Towards seamless mobility in ICN : connectivity, security, and reliability

Xuan Zeng

THÈSE DE DOCTORAT DE
l'UNIVERSITÉ PIERRE ET MARIE CURIE

Spécialité

Informatique

École doctorale Informatique, Télécommunications et Électronique (Paris)

Présentée par

Xuan ZENG

Pour obtenir le grade de

DOCTEUR de l'UNIVERSITÉ PIERRE ET MARIE CURIE

Sujet de la thèse :

Vers une mobilité transparente dans le réseau ICN: connectivité, sécurité et Fiabilité

soutenue le 18 juin 2018

devant le jury composé de :

M. Giovanni Pau	UPMC, France	Directeur de thèse
M. Luca Muscariello	Cisco, France	Co-directeur de thèse
Mme. Eve Schooler	Intel, USA	Rapporteur
M. Ioannis Psaras	UCL, UK	Rapporteur
M. Serge Fdida	UPMC, France	Examinateur
M. Xiaoming Fu	Univ. of Gottingen, Germany	Examinateur

Abstract

With the phenomenal spread of mobile devices, mobility has become a basic requirement as well as a compelling feature to integrate into 5G. Recent statistics from Cisco Visual Networking Index projects that: *by 2021, traffic from wireless and mobile devices will be more than 63% of total IP traffic and connected mobile devices will grow to 11.6 billion, significantly outnumbering fixed hosts.* These numbers confirm the emergence of a mobile Internet.

However, despite the numerous efforts devoted to enabling mobility within IP networks in the past decades, the resulting set of mechanisms are mostly relying anchors, and hence inefficient, complex and access-dependent solutions (e.g., mobility management in 3G/4G). In this context, research community proposed Information-Centric networking(ICN), a data-centric networking paradigm, to address Internet mobility as well as other issues with IP-based network. While ICN has some intrinsic support of mobility, some research challenges remain as open research questions in the mobile ICN domain.

In the thesis, we explored such challenges in ICN to fully support mobility. In particular, we have focused on three important ones: 1) the producer mobility management. 2) the security associated with producer mobility. 3) congestion control (i.e., transport layer) performance in mobile ICN network.

To address the producer mobility management problem, we present our design, implementation and evaluation of *MAP-Me*, a novel anchor-less micro producer mobility (i.e., inter-AS mobility) management protocol in ICN, aiming at supporting latency sensitive traffic of stringent service requirements. *MAP-Me* defines a name-based mechanism operating in the forwarding plane. It preserves ICN key benefits such as multi-path, caching. Thorough evaluation of *MAP-Me* under a variety of impacting network parameters including mobility pattern, topology, wireless radio models against several existing alternatives using ns3 based simulations demonstrate that *MAP-Me* improves user performance of handoff latency, packet loss, and path stretch while retaining low network overheads.

We further extend the above work by investigating security implication in producer mobility. Specifically, we focus on the type of *prefix hijacking attack*: attacker diverts interests under a name prefix to himself by misusing mobility protocols. This also serves as basis to launch other attacks such as black-hole, cache-pollution attack or to collect consumer's privacy. To prevent prefix hijacking, we propose a light-weight, fully distributed and very low-overhead protocol for *name prefix attestation* based on hash-chaining. The *prefix attestation protocol* can protect *MAP-Me* as well as other tracing-based producer mobility proposals in ICN. First results show order of magnitudes improvement in verification latency with respect to signature verification, the leading alternative approach to thwart prefix hijacking attacks in ICN literature. The mechanism is also resistant to *replay-based prefix hijacking*, not addressed by prior work.

Finally, beyond providing connectivity guarantees, additional transport-layer mechanisms are needed to preserve performance in mobile environment. Therefore, we investigate issues with congestion control in mobile ICN networks. Specifically, we focus on addressing the adverse effect of wireless/mobility loss on ICN's receiver-driven congestion control in mobile networks. We introduce *(i) WLDR* and *(ii) MLDR* to achieve in-network loss detection and recovery to facilitate congestions. The approach leverages ICN's in-network processing capabilities to improve congestion control in mobile networks. We demonstrate by ns-3 based simulations that a significant reduction in terms of flow completion time (up to 20%) and request satisfaction time, i.e., the time between the first request emission and the corresponding data packet reception at the consumer. Additionally, our proposal provenly removes any dependence from network/application timers that existing ICN solutions can have issues with.

Contents

List of Figures

List of Tables

List of Publications

[P1] Jordan Augé, Giovanna Carofiglio, Giulio Grassi, Luca Muscariello, Giovanni Pau, and **Xuan Zeng**. Anchor-less producer mobility in icn. poster. In *Proceedings of the 2nd International Conference on Information-Centric Networking*, pages 189–190. ACM, 2015.

[P2] Giovanna Carofiglio, Luca Muscariello, Michele Papalini, Natalya Rozhnova, and **Xuan Zeng**. Leveraging icn in-network control for loss detection and recovery in wireless mobile networks. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, pages 50–59, 2016.

[P3] Alberto Compagno, **Xuan Zeng**, Luca Muscariello, Giovanna Carofiglio, and Jordan Augé. Secure producer mobility in information-centric network. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*, pages 163–169. ACM, 2017.

[P4] J. Augé, G. Carofiglio, G. Grassi, L. Muscariello, G. Pau, and **X. Zeng**. Map-me: Managing anchor-less producer mobility in content-centric networks. *IEEE Transactions on Network and Service Management*, PP(99):1–1, 2018.

Chapter 1

Introduction

With the phenomenal spread of mobile devices, mobility becomes a basic premise of communication as well as a compelling feature to integrate into 5G. Over the past decades, Internet has witnessed an exponential increase in both the mobile data traffic and connected mobile devices. Recent statistics from Cisco Visual Networking Index [1] projects that: *by 2021, traffic from wireless and mobile devices will account for more than 63% of total IP traffic. Mobile devices connected to the Internet will grow to 11.6 billion, significantly outnumbering fixed hosts.* These numbers highlight the emergence of a mobile Internet.

Meanwhile, mobility can play a vital role in the design of next generation 5G networks. Notably, 5G is expected to support communications over dense heterogeneous wireless access and in the presence of mobility [2, 3]. where frequent handovers across small cells impose new architectural challenges. Moreover, 5G will accommodate the bandwidth-demanding and latency-sensitive applications such as HD video streaming or VR/AR [2, 4], which will place stringent requirements to support mobility.

However, today's Internet (TCP/IP) still falls short to support mobility. In the past two decades, the need for a mobility-management paradigm to apply within IP networks has driven lots 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.

In this context, the research community has proposed Information-Centric Networking (ICN[5]), a promising Future Internet Architecture to address mobility as well as other issues within IP-based networks. ICN offers a number of advantages with respect to IP-based networks. One of the recognized strengths is its superior support of mobility. In particular, consumer (data requester) mobility can be naturally supported. However, a number of research challenges still remain before ICN can fully support mobility.

The research work of the thesis focuses on various architectural challenges of ICN to fully support mobility and proposes novel solutions to address them.

The rest of the introduction chapter is structured as follows: In Section 1.1 we describe the limitations in current Internet architecture to support mobility. In Section 1.2, we describe how ICN could relieve some of those limitations and provide better support of mobility. In Section 1.3, we highlight some of the remaining architectural challenges of ICN to fully support mobility, which the thesis aims to address. Finally, in Section 1.4, we summarize the contributions of the thesis and present the organization.

1.1 Current Internet Mobility Support

While mobility support becomes increasingly important, current Internet is limited in its support of it. Inspired by the survey of Deguang and al. [6], we identify the following challenges that make IP-based networks fall short to support mobility:

Mobility Management: After relocation of a mobile node (a change in its point-of-attachment(**PoA**)), network needs to keep session-continuity as well as network reachability to the mobile node. Handling such mobility has been a significant challenge that has plagued the Internet over the past decades. The problem stems from the fact that in the current Internet IP address plays the roles of both locator and identifier. Indeed, when node moves, the TCP connection established in the previous network will break due to a change to its IP address.

Such challenge has striven lots of efforts from the research community and the standardization bodies (e.g, 3GPP) over the past decades, all resulting in a complex access-dependent set of mechanisms implemented via a dedicated control infrastructure (e.g Mobile IP). Recent studies show that such an approach employed by current 3G/4G mobile EPC (evolved packet core [7], i.e., the new cellular system architecture introduced in 3G/4G) incurs long data access latency [8] as well as scalability issues when the population of mobile devices increases [9].

Multi-homing: Multi-homing refers to the practice of connecting a host to multiple networks simultaneously. For instance, a mobile phone can be connected to both a wifi network and a 3G network simultaneously via its 2 wireless interfaces. It provides new opportunities for improving application performance (e.g increasing reliability or throughput via its available multi-path in presence of multi-homing). However, multi-homing is not supported at network layer by current IP-based Internet. Complex transport layer techniques (e.g., Multi-path TCP) must be used to enable multi-homing [10]

TCP Congestion Control in Mobile Internet: Since traditional TCP is designed for fixed networks, it does not adapt well to Internet mobility and often suffers from throughput degradation in a mobile environment. Specifically, TCP assumes a contemporaneous end-to-end path, which is often violated in mobile scenarios. To cite a few of the traditional concerns when TCP runs in mobile environment: 1) misinterpretation of wireless losses as congestion signals and consequently throughput degradation. 2) increased end-to-end delays due to loss detection and recovery at sender-side. 3) packet loss caused by connection state migration and re-establishment in case of mobility.

Security: Current Internet practice is to secure connections and security is tied to IP addresses. Therefore, upon mobility, user migrates or re-establishes connections, requiring to update security associations as well, which is insufficient.

1.2 ICN: a New Paradigm to Address Internet Mobility

Native support for mobility at network layer is a recognized strength of ICN as its data-centric nature relieves several limitations of traditional approaches used in IP. In particular, if offer the following benefits in supporting mobility compared to IP based network:

First, regarding mobility management, unlike IP-based network, ICN brings some native support at network layer. In particular, consumer (data requester) mobility is naturally supported: thanks to ICN's data-centric nature, a change in physical location for the consumer does not translate into a change in the data plane like for IP. The retransmission of requests for data not yet received by the consumers can take place without any needs to signal the network. Moreover, since ICN's communication model is *connection-less*, relocation of mobile nodes does not necessitate the re-establishment of a connection, which is a cost that can be significant in IP-based networks.

Second, in contrast to an IP-based network, an ICN network natively supports multi-homing and multi-path forwarding at network layer. This is because ICN interest packet cannot loop thanks to its *tasteful forwarding plane* [5] (i.e., ICN routers maintain states of forwarded requests until the requested data comes back or the request is eventually timed out).

Third, in regard to congestion control in mobile environment, ICN's stateful forwarding plane offers the potential to overcome limitations that exist when TCP-like congestion control is applied in mobile environment. More specifically, ICN's stateful forwarding plane can facilitate rate and congestion control. However, such potential is under explored by existing ICN research, and it will become clearer in chapter 6, where we leverage this potential to improve ICN congestion control in mobile network.

Finally, for security, ICN secures content rather than connections, which does not require updates to the security association (as opposed to required by IP) in case of mobility.

1.3 Issues with ICN Mobility Support

While ICN provides benefits to support mobility, it brings new research challenges as well. In particular, based on the survey of Tyson and al. [11] as well as our own extensions, we identify several key research challenges that remain in the mobile ICN domain. We begin by describing several unsolved challenges at ICN network layer, and end with one challenge at ICN transport layer.

Producer Mobility: while supporting consumer mobility comes at no extra cost in ICN, supporting producer mobility remains a challenge. In the case of producer mobility, the topology no more reflects the naming structures, and it is necessary to update (global) routing information to reflect the changes. However, relying purely on routing is not an option as it introduces significant overhead and scalability issues in the network. Moreover, It is desired to preserve ICN key benefits such as multi-path, caching, multi-homing etc. The problem will be more challenging for scenario with real-time (audio, video) communication, where stringent latency requirements (i.e, in the order of milliseconds) needs to be met regardless of high speed mobility. User mobility in an infrastructure-based network can be broadly classified in 2 categories:

- **Micro Mobility** is the movement of a mobile user (or device) within a single autonomous system across different point of attachment (or base station).
- **Macro Mobility** is the movement of a mobile user across different autonomous systems.

In general, micro-mobility occurs more frequently over shorter time scales compared to macro-mobility. Therefore the goal of micro-mobility management is often to maintain continuous and seamless connectivity, while the goal of macro-mobility is to ensure mobile users can reestablish communication after movements rather than to provide continuous connectivity.

How to tackle such problems leveraging ICN key primitives is still an open research question. Previous attempts have been made in ICN literature to go beyond the traditional IP approaches, by using the existing ICN request/data packet structures to trace producer movements and to dynamically build a reverse-forwarding path (see [12] for a survey). However, They still rely on a stable home address to track producer movements [13] or on buffering incoming requests which can fall short to support latency-sensitive applications.

Mobile Security: in ICN, security of mobile consumer is straightforward to enforce: as ICN secures the content itself rather than the connection, a baseline security for mobile consumer is that it can check the validity of received data by verifying signature embedded in the data packet.

On the other hand, security of mobile producers need more investigation. In fact, deploying a producer mobility management protocol without adequate security mechanisms pose serious security threats for both the network and the producers. In particular, attacker can perform the *prefix hijacking attack* [14], i.e., an attacker can divert consumers' requests to itself by misusing the mobility protocol (e.g forging control message for producer mobility). By doing so, the attacker then can: perform black-hole attacks to its victims [15], make genuine content cached in the network unreachable or pollute in-network caches with bogus content [16], prevent consumers from receiving the content they asked for [17], collect consumers' interests to attack their privacy [18].

while several protocols have been proposed to address the challenges of producer mobility, However, this imposes significant verification cost at routers and potentially opens the door of DDoS attack [19]. Second, in addition to enabling seamless mobility, we also need to guarantee associated security in producer mobility. Namely, with producer mobility, we should retain the same security level as that ICN intrinsically supports.

Request Staleness: As a result of ICN's stateful forwarding plane, a mobile consumer can leave a large number of stale pending requests in the network, each leads to out-of-date consumer locations. However, currently there is no way to remove such stale requests and it inevitably wastes bandwidth. Moreover, to achieve high performance in congestion control, the congestion window size could grow to be very huge, resulting in ah high percentage of wasted bandwidth.

Congestion Control in Mobile ICN networks: beyond providing connectivity guarantees, additional transport-layer mechanisms are required to guarantee flow performance. In particular, the high level of diversity of wireless media characteristics, combined with frequent end-point mobility, present a challenge to congestion control: the packet loss due to wireless or mobility have adverse effect on receiver-driven congestion control of ICN. In particular, the misinterpretation of wireless or mobility losses as congestion signals will cause unnecessary window size reduction, degrading throughput at the receiver side.

In fact, this is a shared concern for TCP-like congestion control as well for receiver-driven congestion designed for fixed ICN network. However, ICN creates new opportunities to overcome such issues: soft-state associated with pending requests in ICN enables fully distributed in-network decisions that can help rate and congestion control, which would be otherwise performed at the consumer side only. Leveraging such in-network processing capability of ICN to facilitate congestion control in mobile environment is not yet explored by prior work.

1.4 Thesis Contributions

In this thesis, we explored several important architectural challenges in ICN to fully support mobility. Here, we focus on three of the aforementioned challenges : 1) the producer mobility management. 2) the security associated with producer mobility. 3) congestion control (i.e, transport layer) performance degradation in mobile ICN network. We have proposed novel solutions to address each of them. The thesis collects personal and collaborative work done during the author's PhD research and makes the following specific contributions:

- The design, implementation and evaluation of *MAP-Me* a micro producer mobility management protocol, aiming to support latency-sensitive applications. Unlike prior work, *MAP-Me* does not require anchor for producer mobility and is fully distributed (**Chapter 4**).

- A comprehensive performance comparison between *MAP-Me* and other state-of-the-art proposals to manage producer mobility in ICN, under a variety of network conditions including different mobility patterns, topologies, wireless radio models using ndnSIM 2.1. The extensive simulation results demonstrate MAP-Me improves user performance of handoff latency, packet loss, and *path stretch* (i.e., the ratio between the actual communication path length and the shortest path length) while retaining low network overheads (**Chapter 4**).

- a mobility management protocol simulation framework on top of NDNSim 2.1 that has been made open source, including implementation of *MAP-Me*, other state-of-the-art proposals, a wide range of real-world and synthetic topologies, mobility patterns, and radio models for test (**Chapter 4**).
  This and the above 2 results have been **published in** [P1, P4].

- a *prefix attestation protocol* based on hash-chaining is proposed that can secure *MAP-Me* as well as other trace-based mobility protocols against prefix hijacking attack. Analytical results show that it introduces minimal computational and storage overhead compared to signature base alternatives. Hence, our proposal can run unchanged on commodity hardware deployed at network access and mobile cores (**Chapter 5**).

- the *prefix attestation protocol* is resistant to *replay-based prefix hijacking attack*, which is not considered by prior work (**Chapter 5**).
  This and the above result have been **published in** [P3].
- develop and evaluate 2 mechanisms WLDR/MLDR to facilitate receiver-driven congestion control in mobile ICN network. They leverage ICN's in-network processing capability, which is not explored by prior work. Simulation results demonstrate that WLDR/MLDR can effectively reduce flow completion time up to 20%. (**Chapter 6**).
- WLDR/MLDR provenly removes any performance dependence on network/application retransmission timers that existing ICN solutions rely on and are not easy to set (**Chapter 6**).
  This and the above result have been **published in** [P2].

The rest of the thesis is structured as follows: in Chapter 2, we presents an overview of ICN and highlight architectural aspects relevant to the thesis context. Chapter 3 surveys existing work on producer mobility management, prefix attestation and congestion control in mobile and wireless networks.

In Chapter 4 we present our protocol to address producer mobility management in ICN networks. In particular, we present the design, implementation and evaluation of *MAP-Me*, a novel anchor-less micro producer mobility (i.e, inter-AS mobility) management protocol in ICN, aiming at supporting latency-sensitive traffic. The focus of latency-sensitive traffic is due to its more stringent performance requirements such as minimal packet loss and low end-to-end delays compared to the other classes of traffic. *MAP-Me* defines a name-based mechanism and operate in the data plane. It preserves ICN key benefits such as multi-path and caching. We have thoroughly evaluated *MAP-Me* under a variety of impacting network parameters including mobility pattern, topology, wireless radio models and compared it against several existing alternatives in the ICN literature. Extensive ns3-based simulations results demonstrate that *MAP-Me* improves user performance of handoff latency, packet loss, and path stretch while retaining low overhead in the network.

In Chapter 5, we extend the work of Chapter 4 by further investigating the security implications of producer mobility. Specifically, we focus on a type of *prefix hijacking attack*. This also serves as a basis to launch other types of attacks such as black-hole, cache-pollution attack or to collect consumer's privacy. To prevent this class of attacks, we propose a lightweight, fully distributed and very low-overhead protocol for *name prefix attestation* in the network based on hash-chaining. The *prefix attestation protocol* can secure *MAP-Me* as well as other trace-based producer mobility proposals in ICN. First results show orders of magnitude improvement in verification latency with respect to signature verification, the leading alternative approach to thwart prefix hijacking attacks in ICN literature. The mechanism is also resistant to *replay-based prefix hijacking attacks*, which has not been considered by prior work.

In Chapter 6 we address the issue of congestion control in mobile ICN networks. In particular, we focus on addressing the adverse effect of wireless and mobility loss on receiver-driven congestion control in mobile ICN networks. We introduce *(i) WLDR* and *(ii) MLDR* to achieve in-network loss detection and recovery to facilitate congestion control. The approach leverages ICN's in-network processing capabilities to improve congestion control in mobile networks, which has not been explored by prior work. We demonstrate by ns-3 based simulations a significant reduction in terms of flow completion time i.e., the time from when the first data packet of an ICN flow is received until the last data packet of that flow is received and request satisfaction time, i.e., the time between the first request transmission and the corresponding data packet reception at the consumer, which is particularly important in case of latency-sensitive applications. Additionally, our proposal provenly removes any dependence from network/application timers that exist in current ICN solutions.

Finally, chapter 7 summarizes the conclusions of the thesis work and Chapter 8 discusses the future work.

Chapter 2

Background on ICN

In this chapter, we briefly review Information-Centric Networking (ICN)'s design principle and only highlights some of its architectural aspects relevant to the context of the thesis. For additional information, a seminal paper on ICN and its survey can be found in [5, 20].

2.1 Motivation and Principle of ICN

Since Internet was designed in 1960s, its usage has evolved enormously. one of the biggest shifts has been that the Internet is increasingly used for large-scale information dissemination (e.g., web-pages, videos), rather than for pair-wise communication between end hosts. According to Cisco Visual Network Index [1], video traffic will account for 79% of total Internet traffic by 2018. In response to such shift, the research community has proposed a data-centric Internet architecture to better meet current Internet usage. This proposal is called ICN (Information-Centric Networking)

ICN is based on the unique principle of *named data* in contrast to *named host* (i.e, IP addresses) employed by today's Internet. In other words, an ICN packet names a data rather than an end-point. ICN brings a number of known advantages over today's TCP/IP architecture, including: in-network caching to save bandwidth and reduce latency, multipath delivery, improved support for mobility, etc.

There have been several active projects adopting ICN's information-centric approach for developing future Internet architecture, including but not limited to: CCN/NDN [5, 21], DONA [22], NetInf [23], JUNO [24], PURSUIT [25], MobilityFirst [26]. A good survey highlighting similarities and differences among such ICN architectures can be found in [20].

In this thesis, however, we focus on the CCN/NDN [5, 21] architecture, because they attract more attention within the ICN research community. Also because they provide better source code availability. In the rest of the chapter, we will describe the details of CCN/NDN architecture as it is used as the reference architecture of ICN in the rest of thesis.

2.2 Packet Types and Communication Model

There are two types of packets exchanged in CCN/NDN network: 1) **Interest** that carries a request for a piece of information. 2) **Data** that carries a specific piece of information. Figure 2.1 shows their respective formats. The most important field is the name that identifies the information requested/contained by the Interest/Data packet.

CCN/NDN adopts a pull-based communication model: user who wants a data must initiate its request by sending an interest to the network. Interest is then forwarded in a name-based fashion up to the data source, which will respond with the requested data. Optionally, intermediate routers can serve the request (i,e, the interest) directly by their cache if they have data matching the interest. In CCN/NDN, we have the notion of **Consumer** and **Producer**: consumer is the node who sends interests to request for data and producer is the node who generates and publishes data under a specific name prefix (e.g, `/www/youtube`).

2.3 Packet Forwarding

To enable the aforementioned communication model, a CCN/NDN node needs to maintain three tables and perform operations based on them: 1)**CS**, Content Store, 2) **PIT**, Pending Interest Table and 3) **FIB**, Forwarding Information Base. We detail each of them as follows:

CS is a memory buffer for data packets. Each data packet passing a node can be cached locally in its CS such that the copy can be reused to satisfy subsequent interests asking for the same data. The node may apply different replacement policies to manage its CS. Each entry in CS maps a name to cached data.

PIT records temporary states of interests forwarded but not satisfied yet. In particular, when an interest arrives at a node, its name and its ingress face are added to a PIT entry temporarily (until interest timeout). In this way, when the corresponding data comes back, the PIT entry can be used to route back the data to the consumer via the ingress face. Each PIT entry maps a name to a list of ingress faces.

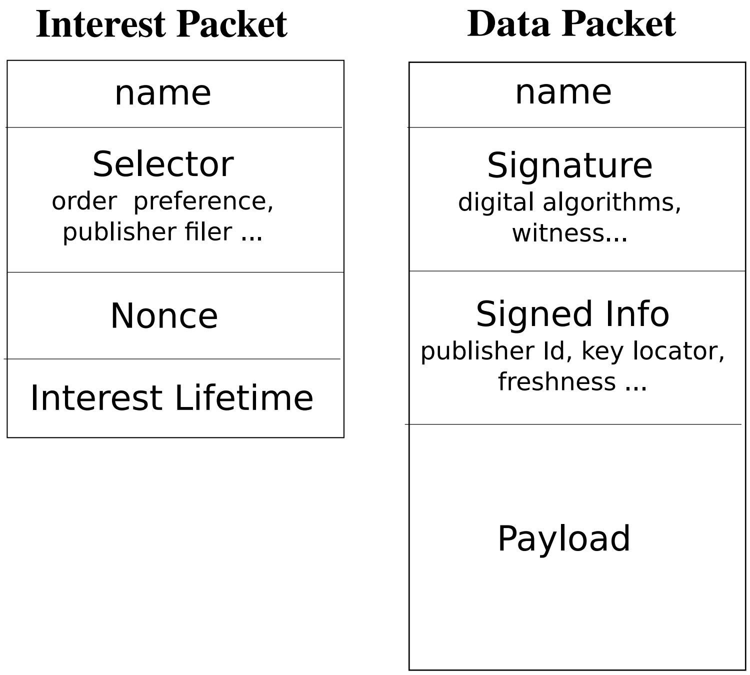


Figure 2.1 – CCN/NDN packet format

FIB is used to forward Interests toward potential source(s)(i.e., producers). Each FIB entry associates a name prefix to a set of outgoing faces that can be used to forward interests under the name prefix. It is almost the same as an IP FIB except that it allows for multiple outgoing faces as opposed to a single face required by IP architecture. This enables CCN/NDN to support multipath and multi-source forwarding. Given the definition of the 3 main data structures, as shown in Fig. 2.2 a CCN/NDN node fowards interest/data in different ways:

For interest arriving at a CCN/NDN node, it is checked against CS, PIT, and FIB sequentially: first, an exact match lookup of interest's name in the CS is performed. Thus if there is already a Data packet in the CS matching the Interest, it will be sent back directly to satisfy the interest. In the meanwhile, the Interest will be discarded (as it is satisfied). Otherwise, an exact-match lookup in the PIT is proceeded. If a match exists,the interest's incoming face will be added to the PIT entry's list of incoming face and no further processing is needed. Otherwise, new PIT entry is created from the Interest and its incoming face. Finally, a longest-prefix lookup in the FIB is performed to send out the interest through one face in the list resulting from FIB lookup.

For data packet, it is checked against PIT and then CS: first, an exact match PIT lookup is performed. If there is no match, it implies data is unsolicited and can be discarded. If

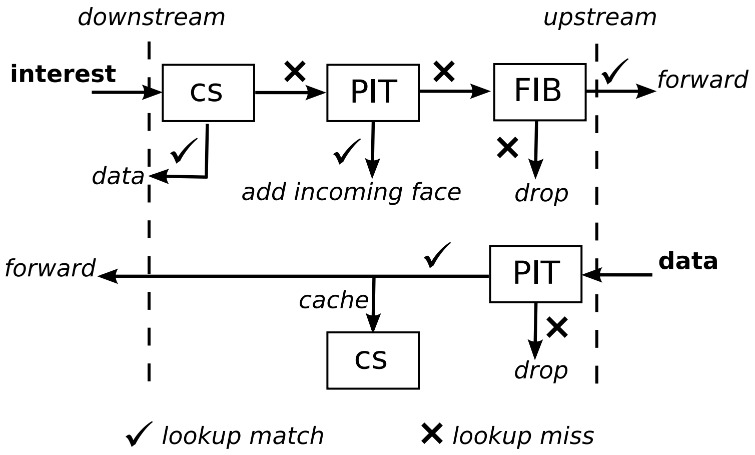


Figure 2.2 – CCN/NDN packet processing pipeline of interest and data respectively

there is a match, data shall be sent through every face in the list resulting from PIT lookup. Afterwards, the data packet is optionally cached in the CS depending on the local caching policy.

2.4 Other Architectural Aspects

While the aforementioned information covers the basic background of CCN/NDN architectures, some other architectural aspects are useful to understand in the thesis context. Therefore, we sketch each of them in this section.

Naming: Even though the naming scheme in CCN/NDN is still an open research topic, initially CCN/NDN has proposed to use hierarchical names consisting of arbitrary number of components, which is similar to URLs. For instance, **/www/youtube/com/video1/001** can be a data name referring to the first chunk of a youtube video and the name components are separated by / character. One of the motivations to use hierarchical names is to reuse IGP/BGP routing protocols from IP network. The names are supposed to be meaningful to upper layers. In particular, the last name component is conventionally the sequence number used by the CCN/NDN transport layer (i.e., similar to TCP ACK's sequence number in functionality).

Routing: In CCN/NDN, only Interest packets are routed, while data just follows "breadcrumbs"(i.e., PIT entries) to go back to the consumer. CCN/NDN employs a name-based routing to route interests. While routing is still an open research topic, CCN/NDN's routing defaults to an hierarchical routing similar to the IGP/BGP protocols used in IP-based networks with the needed customization for a name-based routing.

Security: CCN/NDN employs content-based security. Unlike traditional IP-based networks where one secures connections between two endpoints (e.g., TLS), CCN/NDN secures the data directly. In particular, every data packet is signed with the producer's public key and the signature is computed over the name, the payload and some meta data (i.e., signed info in Fig.2.1) from the data packet. Therefore, the consumer can validate received data by checking the digital signature carried in the data packet with the producer's public key. Note that such content-based security is the enabler for dynamic caching and closest copy retrieval in CCN/NDN design.

Mobility: Since interest and data are forwarded in 2 different ways as shown before, mobility in CCN/NDN can be further classified in 2 sub-problems : consumer and producer mobility. Consumer mobility is naturally supported in CCN/NDN design. When a consumer moves and changes its access point, it is sufficient for the consumer to retransmit the interests for not received data packets. The retransmitted interests are likely to be satisfied by a cached data from intermediate nodes rather than by the producer, which reduces handover latency. On the other hand, producer mobility is more challenging and remains an open research topic in CCN/NDN, which we will discuss later in chapter 4.

Strategy Layer: As mentioned before, CCN/NDN intrinsically supports multi-path forwarding. This gives rise to a new layer between transport and network layer called the *strategy layer* in CCN/NDN architecture. The strategy layer is responsible for making fine-grained, optimized choices of forwarding interface of interests when multi-path is available. The choice can be made based on, for instance, the per-interface statistics measured locally by the strategy layer.

Chapter 3

State of the Art

In this chapter we provide a review of the state of the art for solving research challenges in mobile ICN network. In particular, as mentioned before we focus on 3 issues in mobile ICN network: 1) producer mobility management. 2) security enforcement in producer mobility. 3) improving congestion control in mobile ICN network. For the rest of the chapter, we will review the state-of-the-art solutions from the literature for each of them.

3.1 Manage Producer Mobility

To tackle mobility-management for IP networks, many efforts have been made in the last two decades. However, they are often extremely complex and not implemented proposals. A good survey of these approaches is RFC 6301 [27]. Likewise, within the ICN family, different approaches to mobility–management have been presented [11]. In the following, we will first sketch the mobility-management solutions adopted on ICN proposal-specific basis, then we focus on the CCN/NDN proposal, and review their solutions to the producer mobility management problem.

Among the ICN family of proposals, DONA [22] requires mobile publishers to unregister and re–register their information at each handoff between the hierarchical resolution handlers. Such an update process, however, may incur a non-negligible messaging overhead to eliminate stale registration across the network [20]. Similarly, NetInf [23] and JUNO [24] report network mobility events to a resolution service, which may incur considerable network load in case of frequent mobility [28]. PURSUIT [25], based on a subscribe-publish paradigm instead uses a rendezvous system (i.e., a system responsible for matching subscriber's interest to publications) to handle network mobility, which requires notification to its topology manager at each handoff and, in some cases, the re-computation of the forwarding identifier used to compute the path to the information publisher, prolonging the handoff delay [20, 28]. Finally MobilityFirst [26] uses a global name resolution service (GNRS), which is updated when a node changes its **PoA** (point of attachment, e.g., the eNodeB in LTE network or the access point in wifi network). When facing high-frequency mobility, each of these Resolution-Based **(RB)** approaches presents a similar trade-off: for every packet the consumer has to resolve the producer's location or use stale information and run the risk to reach an old position, incurring a timeout, or Nack, etc.

For the CCN/NDN architecture, several surveys of mobility-management approaches can be found [12, 29]. In [12] for instance, the authors identified four categories of solutions – routing, mapping, and trace-based – depending on the type of indirection point. We build on such classification, refine it and introduce also a new class of approach that does not rely on the existence of any anchor point (*i.e., Anchor-less approach*). To summarize, we classify producer mobility-management solutions into the following five categories:

a) **Routing-based (RT)** solutions rely on intra-domain routing, and require updating all routing in the AS after a mobile device's movement. Scalability of these solutions is widely recognized as a concern which explains why they are usually ruled out, in particular for CCN/NDN where the name space is even larger than IP. For comparison with other approaches, we also define an idealized approach (not feasible in practice) that can instantly update all routers's forwarding information to point to producer's new location upon producer's handover. we call it Global Routing (**GR**), which will be use in the evaluation of *MAP-Me* in chapter 4.

b) **Resolution-based (RB)** solutions rely on dedicated RV nodes (similar to DNS) which map content names into routable location identifiers. To maintain an updated mapping, the producer signals every movement to the RV node [30, 31, 32, 33, 34, 35]. Once the resolution is performed, packets can be correctly routed from the consumer along the shortest path, with unitary path stretch (defined as the ratio between the realized path length over the shortest path one). Requiring explicit resolution, together with a strict separation of names and locators, RB solutions involve a scalable CCN/NDN routing infrastructure able to leverage forwarding hints [30, 31]; however, scalability is achieved at the cost of a large hand-off delay as evaluated in [33, 29] due to RV update and name resolution. In essence, RB solutions show good scalability properties and low stretch in terms of consumer to producer routing path, but ultimately are unsuitable for frequent mobility and for reactive rerouting of latency-sensitive traffic, which are key objective of *MAP-Me*proposed in the thesis.

c) **Anchor-based (AB)** proposals are inspired by Mobile IP, and maintain a mapping at the network-layer by using a stable home address advertised by an anchor. This acts as a relay, forwarding through tunneling both interests to the producer, and data packets coming back. For instance in [36], as a result of the hierarchical routing, the producer needs to change its name prefix each time it moves and changes subnet. Then the producer sends an update message to its anchor to notify the change. In such a context, the anchor's placement is critical for the performance of the approach. MobiCCN [37] uses distributed anchors and selects the closest in a hyperbolic space.

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

d) **Trace-based (TB)** solutions allow the mobile node to create a hop-by-hop forwarding reverse path from its RV back to itself by propagating and keeping alive traces stored by all involved routers. Forwarding to the new location is enabled without tunneling. Like AB though, this approach assumes that the data is published under a stable RV prefix. Kite [13] introduced this approach and proposed storing traces in the PIT to build a breadcrumb trail which could be followed by crossing consumer interests and thus provide a shortcut towards the producer. While it exploits CCN/NDN data plane features without requiring a separate control infrastructure, Kite requires extra 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 [38], where the authors propose a similar trace-based approach, leveraging updates in FIB, rather than in the PIT, and sending updates to both RV and previous PoA.

e) **Anchor-less (AL)** approaches allow the mobile nodes to advertise their mobility to the network without requiring any specific node to act as a RV. They are less common and introduced in CCN/NDN to enhance the reactivity with respect to AB solutions by leveraging CCN/NDN name-based routing. [39] exploits multicast and directs the same Interest to the nearby PoAs of the producer. In [40] and in the *Interest Forwarding* scheme proposed in [33], 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 Interest buffering and introduces a network notification and discovery mechanism to reduce the handoff latency.

[41] instead introduces proxy nodes at the edge of 3G/4G architectures and uses tunnels to forward Interests from the former PoA to the current edge. The solution, however, is specific to cellular networks.

In addition, it is worth to mention that there is an orthogonal class of cache-based mechanisms that can be combined with the aforementioned solutions to enhance performance for both consumer and producer mobility: the *proactive-caching class*. It is orthogonal in the sense that it is a technique that can be integrated with any of the aforementioned solutions to reduce handoff delays. On the consumer side, [42, 43] propose to pre-fetch content at selected nodes before handover occurs to reduce handover delays, while on the producer side, it pro-actively pushes to the network contents to be requested in the near future when handover is imminent [44, 45]. Content is then served by caches when producer is disconnected. Such approaches leverage ICN's in-network caching to keep high content availability regardless of producer mobility. However, such mechanisms can be insufficient for certain realtime applications (e.g., a video call) where content is generated online and not available in advance for pushing.

Finally, in-network caching and name-based routing techniques also enable a routing-to-replica approach abstracting consumers from producer movements (referred to as data depot in [12]). However, such an approach is not well suited for realtime applications or unpopular content, which may have their contents in cache replaced by others due to memory limitations. A study of the advantages for popular items can be found in [31].

3.2 Enforcing Security in Producer Mobility

Security of producer mobility have only been considered to a limited extent by previous work in the ICN literature. Therefore, we begin by briefly reviewing the proposals to secure producer mobility from the ICN literature and presenting their pros and cons. After that, we survey the existing solutions in the IP world for *prefix attestation*[1], which is an important topic to defend against in producer mobility.

Among the mobility management protocols proposed for ICN [46, 47, 48, 49, 50, 51, 52, 13], Kite [13] is the only one that takes security into account in its design and protects the network against prefix hijacking. Specifically, the authors propose to sign traced interests (which corresponds to our Interest Updates), through the producer's private key in order to handle mobility in a secure fashion. Every router receiving a traced interest verifies the signature before updating its network state. The producer trust context (i.e. the public key to trust for validating content signature) attests the producer's entitlement to generate

[1] In IP such a mechanism is usually called address attestation. We maintain the name "prefix attestation" for ease of exposition.

traced interests. However, this approach has some drawbacks: routers must be aware of the producers' trust context, signature verification and certificate chain traversal increases latency during handover, revoking of a prefix to a producer faces the same problem of certificate revocation, where traditional approaches based on Certification Revocations List (CRL) are quite slow and expensive [53].

In the IP world, few works have proposed prefix attestation mechanisms to prevent prefix hijacking in mobility protocols for IP networks. Cellular IP [54] and TeleMIP [55] both adopt the following approach: the first time a mobile host connects to the network, it is assigned an address and a host id by the gateway. Based on that, the mobile host generates a session key and uses the session key to prove its ownership of its assigned IP address. The session key is calculated from a network key, the IP address and the host id. During handover, the host uses the session key to authenticate itself and prove the ownership of the IP address to the new access point. The main drawback of such an approach is the use of a single network key. In the case that the key is stolen, e.g., when a router is compromised, a new network key and a refresh of all session keys must be performed.

Prefix attestation has also been proposed for preventing IP prefix hijacking in inter-domain [56, 57, 58, 59, 60] and intra-domain [61, 62] IP routing. A widely used approach for achieving address attestation exploits digital signatures and certificates. A trusted address holders issues a singed certificate that attests the router's right of announcing a specific address prefix in the network. Both sBGP [58] and soBGP [59] use a public key infrastructure to establish trust between address holder and BGP routers. Similarly, authors of [61] propose to use signed certificates to attest the list of network prefixes an OSPF router can announce to different OSPF areas (i.e., through Router Links LSA messages). While the same approach can be applied to the trace-based mobility protocols, it would suffer the same issues we discussed for Kite.

Finally, two different approaches for address attestation are proposed in psBGP [60] and s-RIP [62]. psBGP proposes a decentralized mechanism: each AS creates a prefix assertion list (PAL), that contains address ownership assertions of the local AS-es and its peers. An origin claim is validated by checking the consistency between the PALs of peers around the advertising origin. S-RIP [62] achieves prefix attestation pre-distributing the mapping between router ids and prefixes to announce in every router of the network. Both mechanisms work well when the mapping between router and prefixes is almost stable. It is worth noting that, instead, in trace-based mobility protocols for ICN such mapping may vary more frequently.

3.3 Improving Congestion Control in Mobile ICN Network

Designing congestion control suited for the mobile environment is a shared concern for IP-based network as well for ICN networks. In this section, we summarize the techniques that have been proposed to improve congestion control in the mobile environment. While a large body of work, beyond ICN, has highlighted the issues of traditional TCP-based congestion control over wireless and mobile network, very few works (in the ICN literature) have considered the same issues in the context of mobile ICN.

Therefore, we begin by discussing the approaches for improving TCP-based congestion control in wireless IP networks as they can potentially be extended also to the ICN architectures. Then we discuss the few related work existing in the ICN literature to optimize congestion control in a mobile environment.

3.3.1 Improving Congestion Control in IP-based Mobile Networks

Extensive research results have improved TCP's performance in mobile environments. Reusing the classifications by Balakrishnan et al.[63], we can divide existing proposals for IP networks into 3 categories: *link-layer, split-connection*, and *end-to-end* solutions.

Split-Connection solutions split the TCP connection into 2 separate connections at the base station: one between the mobile device and the base station, and the other between the base station and the fixed host. Here it is assumed that only one end is a mobile host, and the other end (e.g a server) is in the fixed network. In this way, they can operate more efficiently over wireless links with a specialized TCP implementation tuned for wireless links. I-TCP [64], Freeze TCP, Mobile TCP [65] and SplitTCP [66] adopt this approach. Besides the differences, the advantages of such approaches relate to the capability of shielding the wired segment from the lossy wireless part without requiring end-host modification. Moreover, they are also designed to handle wireless loss as well as mobility loss. As a drawback, the buffering at the proxy between the wired and wireless segments can be considerably high and may introduce additional latency. Also, some of these approaches break the end-to-end semantics, complicating rate control at the server side.

Link-layer solutions attempt to hide wireless losses from the TCP sender through link layer techniques such as local retransmissions and forward error correction (FEC). TULIP [67], MAC MIB [68] adopt this common approach and they leverage ACKs or MAC layer observations to identify and recover wireless losses. Such link-layer proposals have the advantage of operating independently from upper layers and not maintaining per-connection state. However, the local retransmission of such approaches may cause out-of-order data delivery, leading to competing and redundant retransmissions at the TCP layer[63]. Therefore, additional mechanisms based on knowledge of TCP messaging is needed to mitigate the issue. Snoop TCP [69] is one such instance integrating TCP-aware mechanisms. It involves link interface sniffing at the base station for any segment to and any ACK coming from the mobile host and performs retransmission as well as duplicate acknowledgment suppression (to avoid unnecessary fast retransmission) at the base station. However, while such an approach can address wireless loss efficiently, it can not deal with mobility/handoff losses.

End-to-End solutions attempt to make TCP handle non-congestion related losses via modifications only at the sender and receiver. Apparently, end-to-end (E2E) approaches have the benefit of easier deployment. The E2E approaches can be further divided in 2 sub-groups based on whether it requires aid from the receiver side. SMART[70] and E2E-ELN[69] take the approach of exploiting aid from the receiver side: [70] proposes to use SACK(selective acknowledgment) to recover from multiple losses in the same TCP congestion window. E2E-ELN[69] proposes to use ELN (explicit loss notification) from receiver to sender to notify losses due to link error. Also, there are other approaches seeking to improve TCP performance with only sender side modifications. For example, [71] leverages inter-arrival times and [72] uses a relative one-way trip time (ROTT) for congestion signaling. More precisely, the spikes in ROTT measurements are used to differentiate different degrees of congestion. When spikes are observed, losses are considered as due to congestion, otherwise due to wireless transmission. TCP Westwood [73] can handle wireless losses efficiently based on bandwidth estimation: while continuously monitoring the rate of returning ACKs, it uses the rate to estimate bandwidth available and to reset the congestion window and slow start threshold upon timeouts. Since bandwidth estimation is almost unchanged before and after wireless losses, TCP Westwood would not reduce the congestion window upon wireless losses. Finally, [74] compares different E2E solutions without aid from receivers and proposes its own ZigZag scheme. The comparison shows that each of these techniques may perform well in some particular conditions and are less effective otherwise. The existing E2E approaches have targeted to address wireless losses efficiently but it is unclear how they can deal with mobility/handoff losses in E2E approaches.

3.3.2 Improving Congestion Control In Mobile ICN networks

So far, the ICN literature has only marginally considered congestion control implications of mobile and wireless communications. The solution space is far from fully explored as in the case for IP-based networks. Here we briefly summarize each of the proposals from the ICN literature.

[75] deals with ad hoc networks and proposes Interest rate regulation and retransmission timer adaptation according to RTT variations in the wireless network. It is more efficient

than mechanisms relying on a fixed retransmission timer. However, since it is designed for mobile ad hoc networks, it can be insufficient when applied to infrastructure-based mobile ICN network, which is the context of this thesis. Moreover, the solution does not take advantages of ICN primitives to distinguish the nature of the losses nor does it employ faster in network recovery, which is the primary goal of the mechanisms proposed in chapter 6 (i.e, MLDR and WLDR).

Apart from that, the work in [76] combines timer adaption based on RTT and simple ECN (explicit congestion notification) scheme that marks the incoming Data packets in case of congestion and considers all losses not due to congestion as due to wireless. While the proposal partially makes use of ICN's in-network processing capability to improve performance, it does not deal with mobility losses.

Finally, a solution similar to the category of **link-layer solution** of IP network but rather for wireless ICN network is discussed by Klaus et al. in [77]. The solution proposes to use a link adaptation layer that detects loss by observing gaps in sequence numbers or timeouts of local acks and recovers losses by local retransmissions. However, such an approach can not deal with mobility losses (i.e., the same drawback as other link-layer solutions).

In summary, the existing proposals in ICN literature have focused on using timer adaptation, explicit notification or link-layer solutions to improve congestion control in mobile environments. However, approaches taking advantage of ICN's new features such as in-network processing to facilitating congestion control are still missing, which is the motivation of our proposal in chapter 6.

Chapter 4

Producer Mobility Management

4.1 Introduction

As mentioned in section 1, unlike consumer mobility that is naturally supported in ICN, producer mobility remains a challenge. In this chapter, we tackle the problem of producer mobility by presenting our protocol design of *MAP-Me*. Its goal is to manage producer mobility while removing completely the need of any anchor and minimizing handover latency.

MAP-Me defines a name-based mechanism operating in the forwarding plane. It manages producer mobility by exploiting ICN name-based forwarding and letting the producer send a special interest to notify the network about its movement. This special interest will update the FIBs of a subset of routers in hop by hop fashion, establishing new paths to the producer's most recent location. We prove the correctness of the protocol and analyze its stability and its characteristics on path stretch by performing formal analysis.

We thoroughly evaluate performance of *MAP-Me* under a variety of impacting factors including mobility pattern, topology, and wireless radio models in NDNSim 2.1 [78]. We compare *MAP-Me* against the following existing alternatives in literature: 1) *ideal Global Routing* as a benchmark of optimal, where network FIBs are assumed to be instantly and optimally updated on producer movement; 2) *anchor-based*, which is a mobile-IP like solution [79]; 3) *trace-based*, which is based on KITE [13] while incorporating bug fixes identified by us for KITE to support real-time traffic. The extensive simulation results demonstrate *MAP-Me* improves user performance of handoff latency, packet loss, and *path stretch* (i.e., the ratio between the actual communication path length and the shortest path length) while retaining low network overhead. We further evaluate *MAP-Me* under realistic V2I scenarios using trace-driven mobility patterns and real-time applications and *MAP-Me* has shown superior performance.

The key contributions of the work presented in this chapter are the design, implementation and evaluation of *MAP-Me*, a novel producer mobility management protocol in ICN aiming at latency-sensitive traffic. *MAP-Me* has the following beneficial characteristics:

- *MAP-Me* addresses micro (e.g., intra Autonomous Systems) producer mobility. Addressing macro-mobility is a non-goal of this chapter, 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 [80]. *MAP-Me* leverages core CCN/NDN features like stateful forwarding, dynamic and distributed Interest load balancing to update the forwarding state at routers, and relaying former and current producer locations.

- *MAP-Me* is designed to be access-agnostic, to cope with highly heterogeneous wireless access and multi-homed/mobile users.

- Finally, the design also targets low overhead in terms of signaling, additional state at routers, and computational complexity, to provide a solution able to scale to large and dynamic mobile networks.

- We have released all the source code developed for this chapter as opensource [81]. Additional results, as well as more details about the implementation of proposals are available in a technical report [82].

The rest of the chapter is organized as follows: we introduce the design principles of *MAP-Me* in Sec. 4.2, and detail its operations in Sec. 4.3, before analyzing its correctness and path-stretch guarantees in Sec. 4.4. A comprehensive evaluation of the benefits of our anchor-less proposal is then performed in Sec. 4.5. Finally, Sec. 4.6 investigates the interaction and possible cooperation between *MAP-Me* and an existing routing protocol, before concluding the chapter in Sec. 4.7.

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

4.2 Design

In this section, we present the design of our producer mobility management protocol, i.e, ***MAP-Me***. As a data plane protocol, *MAP-Me* handles producer mobility events by means of dynamic FIB updates with the objective of minimizing the unreachable time of the mobile producer. It relies on the existence of a routing protocol responsible for creating/updating the FIB of all routers, possibly with multipath routes, and for managing network failures (e.g., [83, 84]). *MAP-Me* consists of 2 components:
(1) an **Update protocol** (*MAP-Me*-IU) (Sec.4.2.1), which is the central component of our proposal; producer issues an IU message upon each of its movement, which updates router FIBs along with IU's propagation so as to make producer reachable at its new location.
(2) a **Notification/Discovery protocol** (Sec.4.2.2), to be coupled with the Update protocol (the full approach is referred to as *MAP-Me*) to further minimize the unreachable time of the producer in order to meet the demand from realtime/latency-sensitive applications.

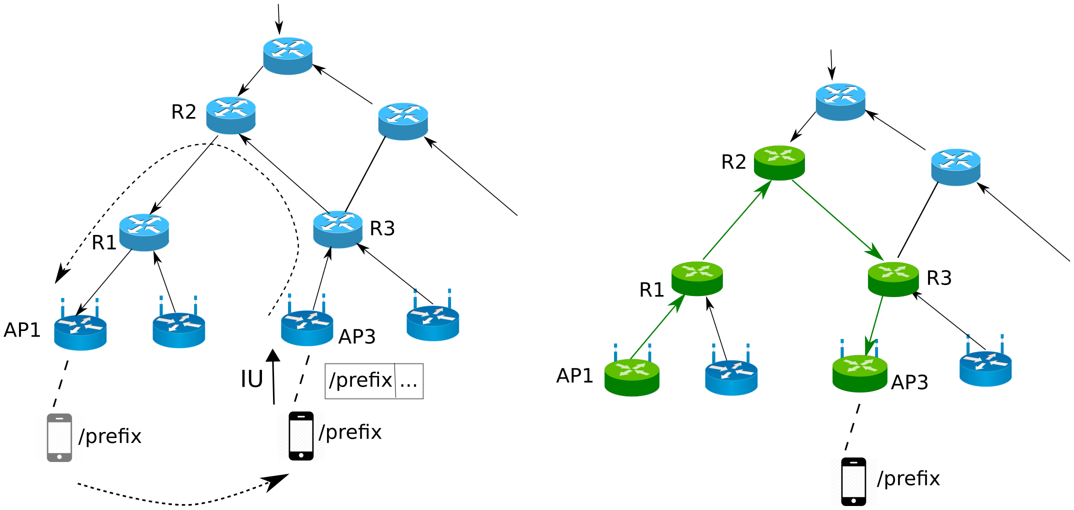
In the following, we describe each of the 2 protocols components in detail, and then present the combination of the two.

4.2.1 *MAP-Me* Update protocol

Rationale

The rationale behind *MAP-Me*-IU is that the producer announces its movements to the network by sending a special Interest Packet, named *Interest Update* (IU) to "itself" after it reattaches to the network. Such a message looks like a regular Interest packet named with the prefix advertised by the producer. As such, it is forwarded according to the information stored in the FIBs of traversed routers towards previous locations of the producer known by router FIBs. A special flag carried in the header of the IU allows all routers on the path to identify the Interest as a mobility update and to process it accordingly to update their FIBs (a detailed description of the IU processing will be provided in Sec.4.3.2).

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



(a) producer moves and sends IU (b) result of IU update process

Figure 4.1 – *MAP-Me*-IU illustration.

Updates propagation

MAP-Me-IU aims at quickly restoring global reachability of mobile prefixes with low signaling overhead, while introducing limited path stretch (i.e., ratio between number of hops of the selected and the shortest path).

Let us illustrate the behavior of the interest update protocol through the example in Fig. 4.1. In this example a mobile producer with prefix */prefix* moves around different network access points(APs). First let us focus on Fig. 4.1(a), where initially the producer connects to one of the APs(i.e AP1) and we assume that network FIBs for */prefix* are then populated by a name-based routing protocol such that they represent valid paths to the producer's initial location AP1. This is illustrated in Fig. 4.1(a), where the arrows between neighbor routers indicate the forwarding output face of FIB at each router for */prefix* and note that together they form a valid forwarding tree rooted at AP1. For convenience, in the following, we use FIB to refer to FIB entry of */prefix*.

Once the producer moves to a new access point AP3, it issues a special interest (i.e called interest update or IU) to update routers it traverses and steer interests to itself. The IU carries the name */prefix* and is forwarded only based on FIB(i.e matching with PIT or CS are skipped). Since the FIBs are now still pointing to the producer's previous location AP1 , IU will be forwarded to AP1 following the path in FIBs as indicated in Fig. 4.1(a) As IU propagates, each router receiving it will first forward it using output face in its FIB, and then update its FIB with the ingress face of IU. For instance, in Fig. 4.1(a) AP3 first forwards IU to R3, and then update its FIB output face to that connected with producer. Next, R3, R2 and R1 perform the same operation upon receiving it. Eventually, IU will stop at its previous location AP1, where AP1 now has no output face in FIB to forward IU (because the output face was destroyed after producer moves).

As a result, only a subset of routers on the path from AP3 to AP1 are updated, which are highlighted in green in right part of Fig. 4.1(b) Further, it can be verified in Fig. 4.1(b) that the result is a new forwarding tree rooted at the new location of the producer (i.e., AP3). Therefore the producer now is reachable again by consumer interests.

For subsequent movements of the producer, it will issue a new IU to the network at each movement, which will update a subset of routers and always make producer reachable at the new location. The updated routers are always those on a path between producer's current and previous location.

We will elaborate in detail the proof of correctness of the IU mechanism for general topology and multiple handovers in section 4.4. Here we just provide some intuition on the effect of each IU update process to help understand how it works: initially, the FIBs populated by routing should form a directed spanning tree rooted at producer's initial AP (as this is what routing does). Essentially the effect of each IU update process can be viewed as a "flipping" of a subset of directed links in the existing forwarding tree such that the current AP of the producer becomes the new root of the forwarding tree.

Concurrent updates

Frequent mobility of the producer may lead to the propagation of concurrent updates, i.e., the producer issues a new IU to the network before the previous IU completes its propagation to a previously connected AP. Under this condition, IUs may arrive at a router out of order(i.e., an older IU arrives later at a router than a new IU does) possibly due to network congestion. Once this occurs it can break the protocol correctness for the rest of the handover events. This issue will become clear when we deduce the proof of protocol correctness in section 4.4.

To solve this problem, we let the producer maintain a sequence number for the *MAP-Me*-IU protocol incremented at each handover which identifies every IU packet. Network routers also keep track of such sequence numbers in the FIB to verify IU freshness. Without detailing the specific operations in *MAP-Me* to guarantee update consistency (whose description is provided in Sec.4.3.2), we can say that modification of FIB entries is only triggered when the received IU carries a higher sequence number than the one locally stored in the FIB. Conversely, the reception of an IU with lower sequence number w.r.t that stored in the FIB triggers sending back an IU through the ingress face of IU. The sequence number of that IU is set to equal to that in the FIB. The reasoning behind the latter operation is the following: the routers that the IU with lower sequence number must come from a recently visited AP rather than the latest one. Therefore the routers updated by this older IU have outdated information. As a result, we need to send back an IU with the most recent sequence number to update those routers to point to the producer's latest location. This will be further explained in section 4.4.

4.2.2 Map-Me Notification/Discovery protocol

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

Previous work in the Anchor-Less category has suggested the buffering of Interests at the previous producer location [33] to prevent such losses. However, such a solution is not suitable for applications with stringent latency requirements (e.g. real-time application). Moreover, the negative effects on latency performance might be further exacerbated by potential IU losses and retransmissions in wireless links. To alleviate such issues, we introduce two separate enhancements to *MAP-Me*-IU protocol, namely *(i)* an **Interest Notification** mechanism for frequent, yet lightweight, signaling of producer movements to the network and *(ii)* a scoped **Producer Discovery** mechanism for consumer requests to proactively search for the producer's recently visited locations.

Interest Notification

An Interest Notification (IN) is a breadcrumb left by producers at every encountered AP. An IN message looks like a normal Interest packet carrying a special identification flag and a sequence number, like IUs. Both IU and IN packet share the same sequence number that the producer increments without distinction for every sent message and follow the same FIB lookup and update processes. However, unlike IU packets, INs do not propagate further than to the currently connected AP. Rather it is used by the discovery process to route interests to the producer even before *MAP-Me* update process completes.

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

Producer Discovery

The extension of *MAP-Me* with notifications relies on a local discovery phase: when a consumer Interest reaches a AP with no valid output face in the corresponding entry, the Interest is tagged with a "discovery" flag and labeled with the latest sequence number stored in the FIB (to avoid loops). From then on, it is broadcast to one hop neighbors and it will be discarded if it does not find the breadcrumbs left by the producer to track him (notifications). The notifications can either allow to forward consumer Interests directly to the producer or give rise to a repeated broadcast in case of no valid output face. The latter is the case of a breadcrumb left by the producer with no associated forwarding information because the producer has already left that AP as well. A detailed description of the process is reported in Sec.4.3.2.

As further shown in Sec. 4.5, the notification/discovery mechanism is important to preserve the performance of flows in progress, especially when latency-sensitive.

4.2.3 Full *MAP-Me* approach

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

Figure 4.2 illustrates the combined use of interest notifications and interest update in a mobile access network where the different APs are the leaves of a fat tree. The producer is initially in position AP3, having sent an Interest Update (IU 3) message to make itself reachable. Then it moves to AP4 and later to AP5, sending each time an Interest Notification (respectively IN 4 and IN 5). Consumer interests are forwarded using FIB information synchronized with the initial state of the producer and thus reach the initial AP, AP3. Once the producer moves and the face is destroyed, no valid next-hop face information can be found in the FIB and consumer Interests reaching AP3 enter in discovery mode: they are tagged with the sequence number 3 found in the FIB, and broadcasted to one-hop neighbors, which may either forward them directly to the producer (this is the case for the current AP of the producer) or broadcast them one hop further if they have been notified of producer attachment by means of INs, but there is no valid forwarding information. Other network nodes reached by a Discovery Interest (including AP1 in the

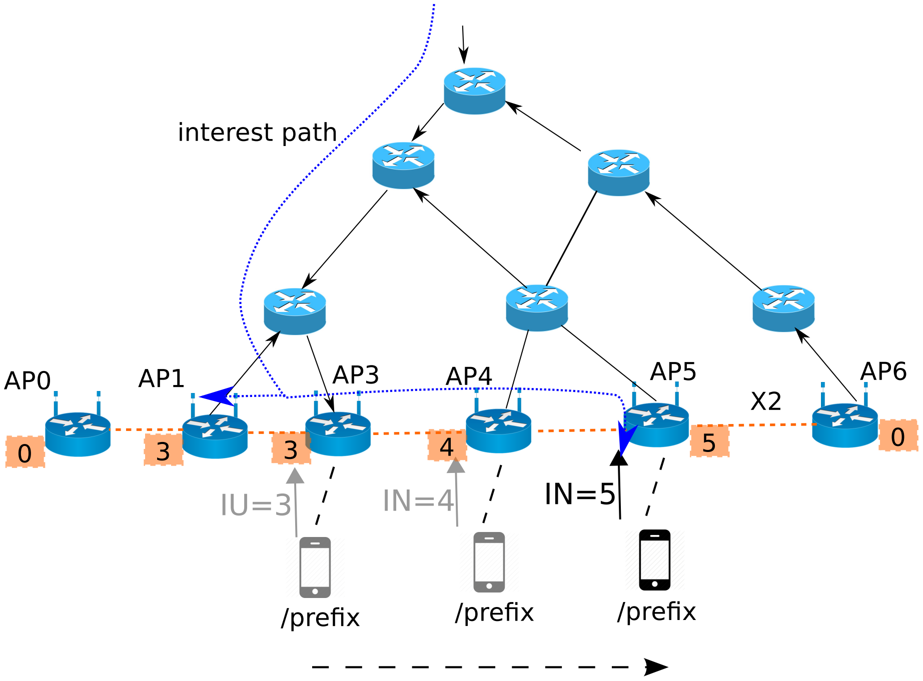


Figure 4.2 – combined IU/IN process example.

example) just discard the packet when they have a lower sequence number than that in the discovery interest because it means they have no fresher information about the position of the producer. The discovery process iterates until the producer is reached.

4.3 Implementation

4.3.1 *MAP-Me* implemented in a CCN/NDN network

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

MAP-Me **Messages**

Two new optional fields are introduced in a CCN/NDN Interest header:

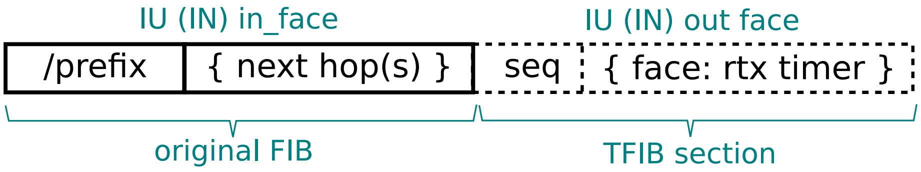


Figure 4.3 – *MAP-Me* FIB/TFIB description.

- a special *Interest Type* (T) to specify four types of messages: Interest Updates (IU), Interest Notifications (IN), as well as their associated acknowledgment (Ack) messages (IU_{ack} and IN_{ack}). Those flags are recognized by the forwarding pipeline to trigger special treatment.

- a *sequence number* to handle concurrent updates and prevent forwarding loops during signaling, and to control discovery interests propagation;

MAP-Me additional Network Information

FIB entries are enriched with a sequence number, initialized to 0 by the routing protocol and updated by *MAP-Me* upon reception of IU/IN messages. The Data about not-yet-acknowledged messages are temporarily stored in what we denote as the **Temporary FIB buffer, TFIB,** to ensure reliability of the process, and removed upon reception of the corresponding acknowledgement. As sketched in Fig.4.3, each TFIB entry is composed of an associative array ($F \rightarrow T$) mapping a face F on which IU has been sent with the associated retransmission timer T (possibly null, denoted $\perp$). Note that compared to the processing of a regular interest, the only extra operation required by an IU/IN is the update of one FIB entry, which costs little [85]. Therefore, IU/IN can be processed at line rate, which helps making *MAP-Me* 's handoff fast.

4.3.2 Algorithm description

IU/IN generation at producer

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

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

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

IU/IN processing at network routers

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

The operations in the forwarding pipeline for IU/IN processing are reported in Algorithm 1.

Consumer request forwarding in case of producer discovery

The forwarding of regular Interests is mostly unaffected in *MAP-Me*, except in the case of discovery Interests that we detail in Algorithm 2. The function `SendToNeighbors(`I`)` is responsible for broadcasting the Interest I to all neighboring PoAs (i.e.,, all the PoAs that are connected with the current PoA via X2 links).

When an Interest arrives at a PoA which has no valid next hop for it (because the

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

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

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

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

producer left and the face got destroyed), it enters a discovery phase where the Interest is flagged as a Discovery Interest and with the local sequence number, then broadcasted to neighboring PoAs. Upon reception of a Discovery Interest, the PoA forwards it direcly to the producer if still attached, otherwise it repeats the one-hop brodcast discovery to neighboring PoAs if it stores a recent notification of the producer presence,
reie an entry in the TFIB having an higher sequence number than the one in the Discovery Interest. Otherwise, the Discovery Interest is discarded. It is worth observing that the discovery process is initiated only in the case of no valid forwarding next hop and not every time a notification is found in a traversed router. This is important to guarantee that the notification/discovery process does not affect IU propagation and IU process completion.

4.3.3 Security considerations

Unlike in the centralized (anchor-based) approach to mobility management, where the security can be enforced at one single node, in *MAP-Me* , the mobility management is distributed and security needs to be enforced in a coordinated way across the entire network. Therefore, we will leave the non-trivial investigation of securing *MAP-Me* as well as trace-based protocols to Chapter 5.

4.4 Proof of Correctness and Stability Analysis

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

We consider m consecutive movements of the producer in network positions $\{P_0, P_1, ..., P_m\}$ and focus on forwarding state variations determined by *MAP-Me* at the time instants corresponding to either producer movements or Interest Update processing. At any such instant, as in Fig.4.1, the network is partitioned into a set of islands, whose number varies in $[1, m+1]$ as a function of producer movements and hence of the number of ongoing update processes. we assume that at the beginning, global routing builds a spanning tree rooted at first location P_0. The tree can be a minimum SP or a shortest-path tree depending on the routing. About the completion of the update process after a movement k, we can state that

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

Proof. Assuming that IU losses are handled by the retransmission mechanism described in Sec.4.2, 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 higher sequence number, hence the IU is sent back with the encountered higher sequence number towards the originating routed position of the producer. Since the maximum sequence number is bounded by m, the maximum number of hops traversed by IU with sequence number k is finite.

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

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

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

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

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

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

Proposition 3. *MAP-Me path stretch for node i over the tree rooted in P_m, created after producer's m-th movement, is upper bounded by the ratio $(|i - P_0|_{P_0} + |P_0 - P_m|_{P_0})/|i - P_m|_{P_m}$ as $m \longrightarrow \infty$, 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 or not P_0 is on the path between i and P_m on the P_m-rooted tree. 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 $m \longrightarrow \infty$, because it is intrinsically related to the properties of the initial tree. □

4.5 Evaluation

4.5.1 Simulation setup

This section assess simulation results of *MAP-Me* over different mobility patterns, radio conditions and network topologies. We implemented both *MAP-Me* and *MAP-Me-IU*, anchor-based (AB), tracing-based (TB) – based on Kite ([13]) – and GlobalRouting (GR) approaches in ndnSIM 2.1 simulator [86]. In evaluation of TB, we have enabled all optional extensions described in [13], because otherwise it yields unreasonable user performance (e.g., handoff latency more than 2s). Also, because the TB approach with this setting performs in its optimality in terms of user performance. Moreover, we don't consider here resolution-based (RB) or other AL solutions as they are not appropriate for latency-sensitive applications (as we have discussed in Sec. 3.1 of chapter 3).

We first evaluate all mobility protocols in a baseline scenario, before varying parameters such as radio conditions, mobility model and network topology in order to gain insight into their sensitivity. All plot data is averaged over many runs, or a large number of handovers (at least 250 per mobile node per run) depending on the context; although, for clarity, we chose not to display confidence intervals in the paper. The full set of results is available in the technical report of *MAP-Me* [82].

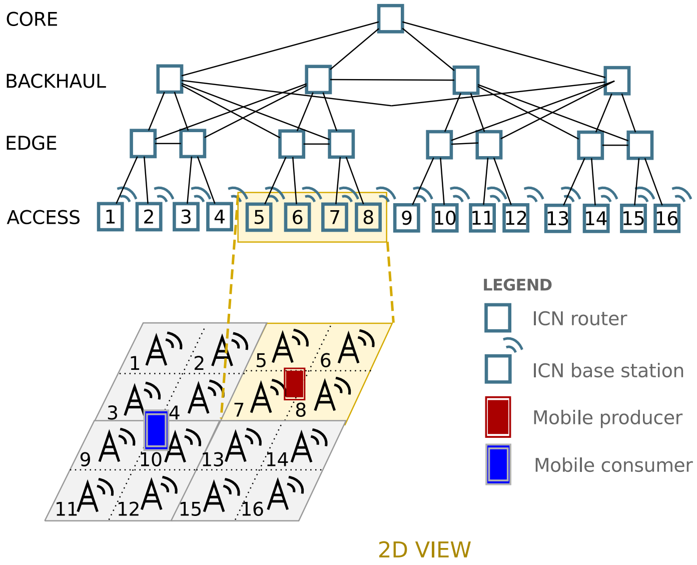


Figure 4.4 – Network with link capacity C=10Mb/s.

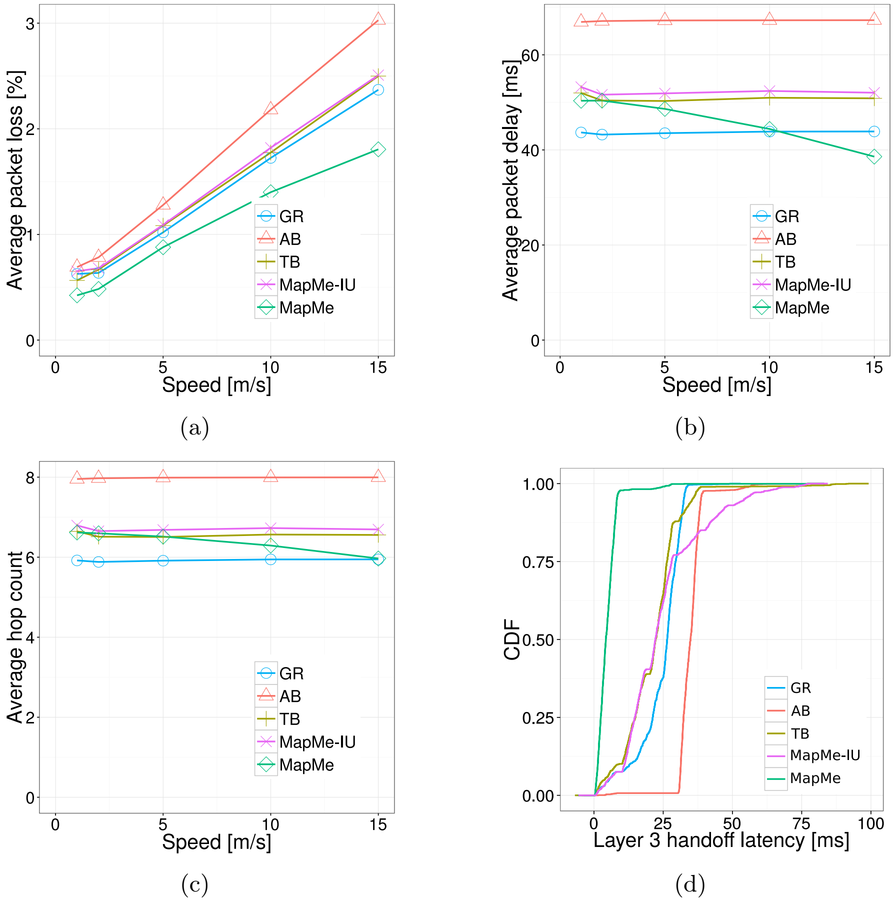
4.5.2 Baseline scenario description

Topology: In the baseline scenario, we use 802.11n access network composed of a 4-by-4 grid of base stations (BS) with square-shaped cell of side $s = 80m$. They are connected to a fat-tree backhaul network represented in Fig. 4.4. This choice is motivated by the similarity in terms of redundancy and meshing found in real ISP access network. Wired links have a capacity of C=10Mb/s and 5ms delay. We complement the baseline topology with a wide range of well-known topology models and Rocketfuel topologies to cover all types of graph metrics in variants of the baseline scenario in section 4.5.4.

Radio and Mobility: We use IEEE 802.11n WiFi on 5GHz frequencies, with Minstrel rate adaptation [87] and log-distance radio propagation model plus Rayleigh-fading model for wireless channel. Mobile nodes move in the 4x4 cells under full radio coverage. We choose *random way point* (**RWP** [88]) mobility model for user mobility. We also vary the mobile's moving speed from 1m/s to 15m/s (i.e, pedestrian to vehicular speed). A range of other radio propagation models and mobility models are also used in the variant scenario in section 4.5.4.

Application: We assume N disjoint pairs of mobile consumers and producers. In particular, we choose N=5 for baseline scenario and also its variants. To highlight *MAP-Me* benefits in the support of latency-sensitive traffic, we consider a *constant bit rate* (**CBR**) audio/video streaming application, characterized by a bit rate of 1Mb/s with no retransmission in the baseline scenario, and further extend it with an adaptive protocol inspired by the Periscope streaming application in Sec. 4.5.6. While, the CBR application has the nice property of reflecting network performance, the adaptive one has a closed-loop behavior that is more realistic but might be affected by wireless and mobility losses. More in-depth study of these interactions is out of scope for the thesis.

4.5.3 Results for baseline scenario: Fat-Tree + RWP + CBR



(a)

(b)

(c)

(d)

Figure 4.5 – User performance: packet loss (a), delay (b), and hop count (c); CDF (i.e., Cumulative distribution function) of layer 3 hand-off latency (d).

User performance

In Fig.4.5(a)-4.5(b), we show two performance indicators for latency-sensitive traffic, average packet loss and delay, both as a function of mobile speed (from 1m/s to 15m/s). We can distinguish two kinds of losses: due to the wireless medium, occurring irrespective of the mobility management approach, and those due to mobility. The fraction of mobility losses is consistently reduced by *MAP-Me*, especially in the presence of the notification/discovery mechanism, as a result of in-flight re-routing of Interests towards the new location of the producer, which prevents Interest timeouts. *MAP-Me-IU* like TB (or alternative AL solutions) enables re-routing of Interests only after the interval of time required for an update to complete. A longer time is required for a global routing update, but the resulting path is the shortest possible, which explains the equivalent performance w.r.t. *MAP-Me-IU*/TB. AB under performs because of worse update completion time and path stretch. The experienced average packet delay in Fig.4.5(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 by 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 with each mobility update process, see Fig.4.5(c), and from the L3 hand-off latency, i.e., the time required for L3 reconnection after a handover, see Fig.4.5(d). The L3 hand-off latency illustrates the reactivity of the mobility-management protocol and highlights the significant improvement brought by *MAP-Me*, which significantly reduces handoff latency compared to other approaches. It is interesting to observe that AB shows a constant latency value of around 30ms due to update propagation up to the anchor, while for GR, TB, and *MAP-Me-IU*, such latency varies according to network distance between producer and routers to be updated, as a function of producer movement in the considered topology. Latency variations can be visualized at the inflection points in the corresponding CDFs in Fig.4.5(d).

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.4.6(a), and the volume of signaling messages per handover to be processed by routers at different positions in the network, in Fig.4.6(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.4.4. 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.4.5(c). TB approaches involve a much higher signaling overhead due to "keep-alive" messages periodically sent to refresh update information. By reporting the way traffic is spread across the network and where signaling traffic goes, we can draw some key observations. Every mobility protocol relies on the control plane that enforces a routing state across the network (shortest-path routing in this paper), which corresponds to the initialization state for mobility. All protocols relying on an anchor have routing pointing to the anchor's location, whereas for AL mechanisms, it points to the producer's position at the routing update time. Thus, AL approaches are able to offload mobile backhaul and core networks from all local traffic, seamlessly as shown in Fig.4.6(c)[2]. Finally, we report about *MAP-Me* sensitivity to parametrization, i.e., the impact of T_U settings. In Fig.4.6(d), we observe that *MAP-Me* has robust parametrization as long as T_U is not too small (signaling overhead and path stretch quickly converges to the best settings) or too high (load on access).

4.5.4 Impact of mobility pattern, radio conditions and topology

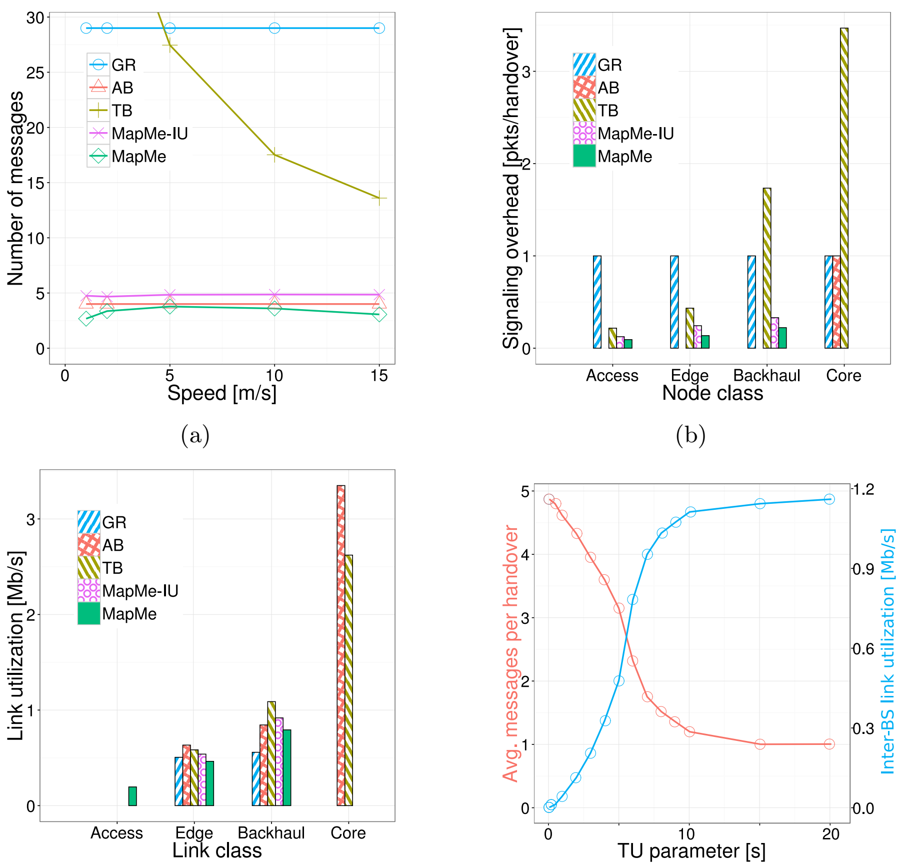
We have performed extensive simulations to evaluate the sensitivity of *MAP-Me* and other competing solutions, by varying several parameters in our baseline scenario [82]. We report here our most significant results and confirm the wide applicability of conclusions from the previous sections.

Impact of mobility pattern and radio conditions

For mobility patterns, we have included the previous jump models across base-stations and classical models available in NS-3 (Random Direction 2D, Gauss-Markov and Random-Walk 2D models). For radio conditions, we have considered an ideal wireless channel (no loss nor interference at layer 2) by dynamically switching wired links up and down to emulate mobile handover across base stations, and the two radio models from ITU specifications [89], namely urban environment without line of sight (LoS) and suburban with LoS.

The impact of both radio and mobility patterns are negligible, and the plots show no significant difference between performance metrics. Comparative simulations with the ideal wireless channel (not represented here) show that the loss rate is not only due to the wireless channel, but also impacted by the mobility scheme in place, and more specifically the time to re-establish connectivity at L3 (denoted L3 handover time). Moreover, with

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



(a)

(b)

(c)

(d)

Figure 4.6 – Network cost: Signaling overhead vs mobile speed (a), overhead (b), and link utilization (c) per router class. Map-Me sensitivity analysis (d).

ideal wireless channel where we can extract out the impact of only L3, we see the relative order of performance of protocols are the same as those in Fig 4.5(a), confirming superior performance of *MAP-Me* in reducing mobility losses.

Impact of topology

We cover a wide range of network characteristics through the use of deterministic and stochastic graphs drawn from well-known models [82], as well as Rocketfuel topologies. Table 4.1)[3] provides detailed information about the topologies. While not being representative of access networks, they provide insights into the performance of *MAP-Me-IU* in non-local mobility (e.g., jumps from WiFi to LTE networks). Edge nodes are randomly picked from graph nodes (or Rocketfuel leaves) to be connected to the previously described grid, while others form the backhaul.

Graph	$\|V\|$	$\|E\|$	comments
fat-tree	29	42	
tree	29	28	
cycle	30	30	$n = 30$
grid-2d	100	180	$m = n = 10$
hypercube	128	448	$n = 7$
expander	100	400	$n = 10$
regular	100	150	$d = 3, n = 100$
erdos-renyi	100	564	$n = 100, p = 0.1$
watts-strogatz	100	200	$n = 100, k = 4, p = 0.1$
small-world	100	437	$n = 10, p = q = 1, r = dim = 2$
barabasi-albert	100	384	$n = 100, m = 4$

Table 4.1 – Topology properties.

As expected, topology is the most impactful parameter for absolute performance, a direct consequence of the forwarding trees built on top of it. Figure 4.7 shows path stretch and L3 handoff latency. As shown in previous simulations, *MAP-Me-IU* and TB both offer lower stretch than AB – sometimes close to optimum (GR) – with a slight advantage to *MAP-Me-IU* in almost all cases. Those variations can be interpreted as the ability for the spanning tree (shortest path tree rooted in anchor for TB, and first producer location for *MAP-Me-IU*) to offer short paths between consumer and producer, and thus offload traffic at the edge.

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

As shown in Figure 4.7, the average handoff latency of TB approach (with all optional options enabled) is slightly better than that of *MAP-Me-IU* with different topologies. However, this is achieved at the cost of redundant transmission of every consumer interest. More

[3] For more information about graph generators: `https://networkx.github.io/documentation/development/reference/generators.html`

precisely, in such setting of TB, consumer interests will be delivered generally along 2 paths in parallel: 1. path following the traces (not via the anchor) to reach producer, and 2. path first reaching the anchor and then to the producer [13]. The handoff latency of TB thus depends on the packet propagation delay to a node either in path 1 or path 2, whichever comes first. This multi-cast increases the possibility that update message of TB hits a node with consumer interests earlier than *MAP-Me-IU* does, and thus leads to a better handoff latency of TB. The cost is that each consumer interest is transmitted twice on two paths, wasting network resources.

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

Beyond confirming our previous observations, these simulations open the way to further extensions of *MAP-Me* by considering how an alternative routing might lead to better performance – for instance using more efficient spanning trees (ST) such as minimum diameter ST (see Prop.3 in Sec. 4.4) – and how more appropriate graph spanners and random strategies could allow the exploration of more than one path.

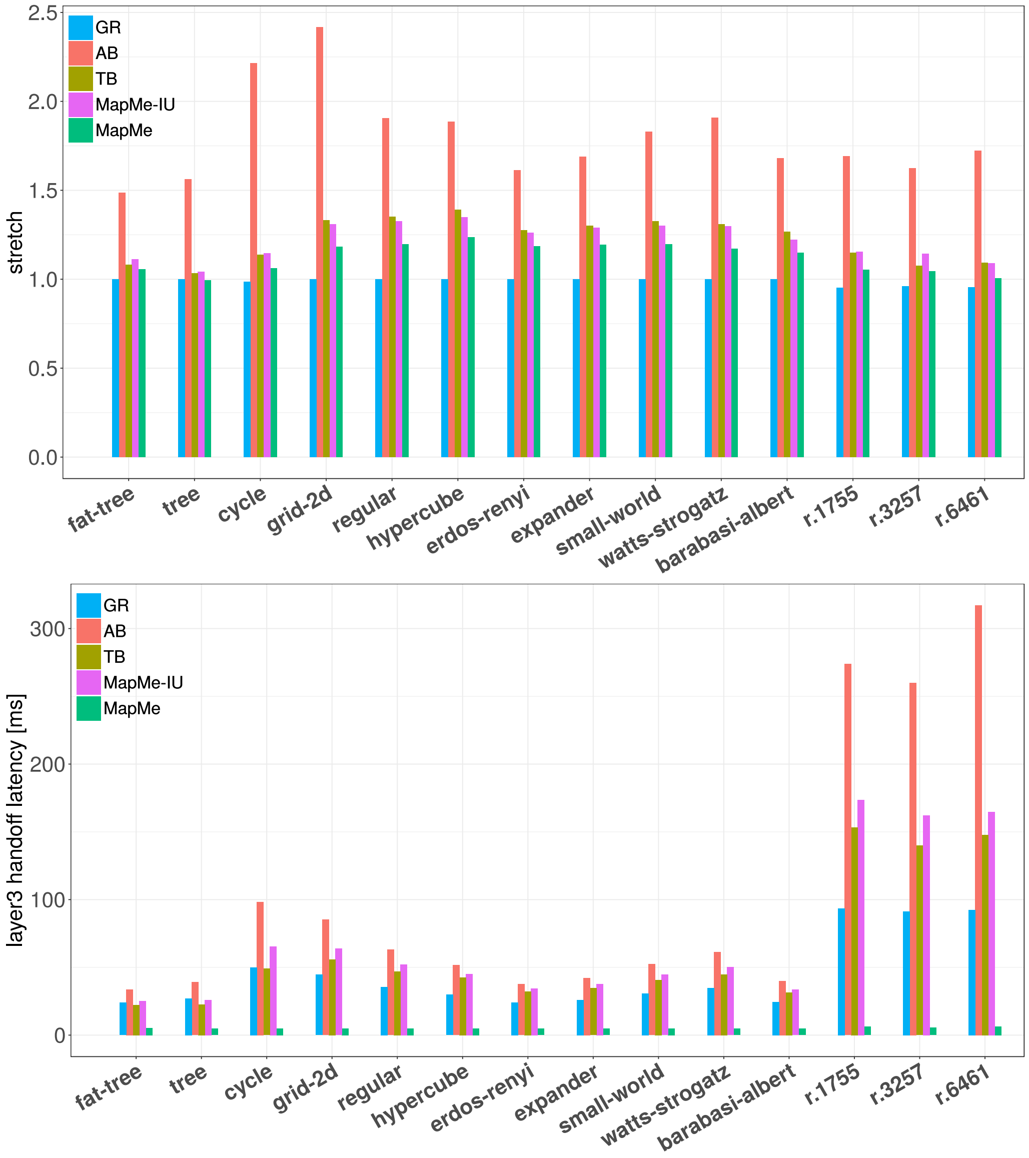
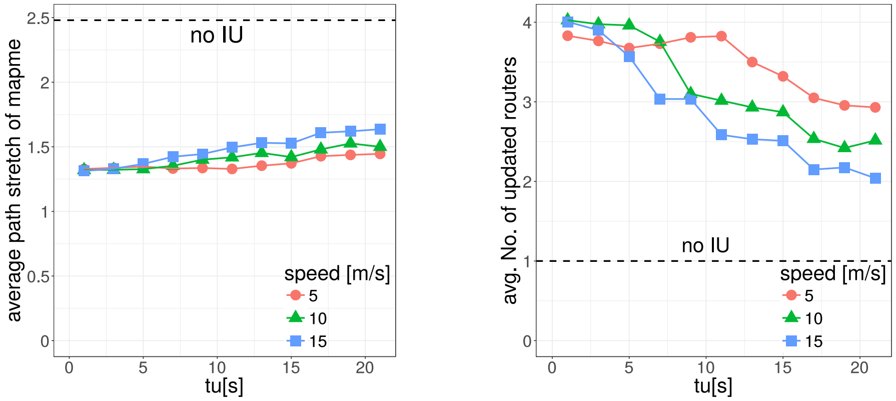


Figure 4.7 – Path stretch and handoff latency for simulated network topologies (r.1755, r.3257, r.6461 refers to Rocketfuel topologies).

4.5.5 Impact of notifications on path stretch

As we have seen, the use of notifications improves performance during fast mobility by using inter-PoA links with the risk of increasing path stretch. We show here that the use of T_U as per the selected mechanism (Sec. 4.2.3) changes the root of the IN's breadcrumb chain and thus limits its length. We thus evaluate the trade-off offered by *MAP-Me* through the adjustment of this T_U parameter by slightly modifying our baseline scenario. Instead of a grid, the PoA are arranged on a line. The producer now moves back and forth across them at a constant speed parameter, while the consumer is now static at the root of the fat tree.



(a)

(b)

Figure 4.8 – Effectiveness of T_U timer: a) Path stretch b) Network overhead (No. of updated routers per handover).

Fig. 4.8(a) shows the average path stretch of *MAP-Me* as function of T_U. The dashed line indicates the path stretch limit reached when no IU is sent. In general path stretch slowly increases with T_U at any given speed and remains well below the no-IU threshold. At low speed, stretch remains constant up to higher T_U values (as an IU is sent for every handover).

If we now consider network overhead depicted in Fig. 4.8(b), we notice that a slight increase of path stretch allows for a significant reduction of network overhead (which peaks here at 50% for a speed of 15m/s). This confirms the interest notifications in absorbing high-frequency mobility while preserving appropriate flow performance. The T_U threshold thus appears a very useful setting to allow a network to cope with challenging mobile workloads.

4.5.6 Trace-driven urban mobility

Topology: To evaluate our approach under more realistic mobility patterns, we consider an urban residential environment spanning a 2.1×2.1 km^2 area in Los Angeles, with a WiFi Hot Spot deployment similar to what Time Warner Cable [4] has in the area, i.e., we have 729 WiFi APs, with the same wireless settings as in the previous (baseline) experiments, connected to the Internet through the fat-tree topology in Fig.4.4.

Mobility: We generate realistic vehicular mobility patterns using SUMO (Simulation of Urban Mobility, i.e., a traffic simulator for large road networks) [90], with maximum car speed set according to road speed limits[5]. We place mobile producers in moving cars and analyze system dynamics on a given time interval (4 minutes, roughly corresponding to 33 handovers), so that all monitored cars are in the map at the same time. In such a scenario, we consider a group communication between one mobile producer and two non mobile consumers requesting different data. Consumers are connected to two APs that are picked at random, uniformly across the network coverage.

Applications: Two types of applications are considered: in the first set of simulations, the previous 1Mbit/s CBR application; in the second, an application that mimics Periscope [80], a popular live video streaming app for smart phones is used. The mobile producer generates two different video streams, each one downloaded and played by one consumer, using a 5 second play-out delay buffer. If the video play-out stops because the consumer has no Data available, we consider this as a failure and momentarily stop the consumer: after a short period of time (few seconds), the consumer restarts downloading new data and to play-out the video. The video data rate is 1Mbit/sec, corresponding to a 480p video resolution. Traffic is scaled up by increasing the number of groups, each of which is identified by the producer serving data.

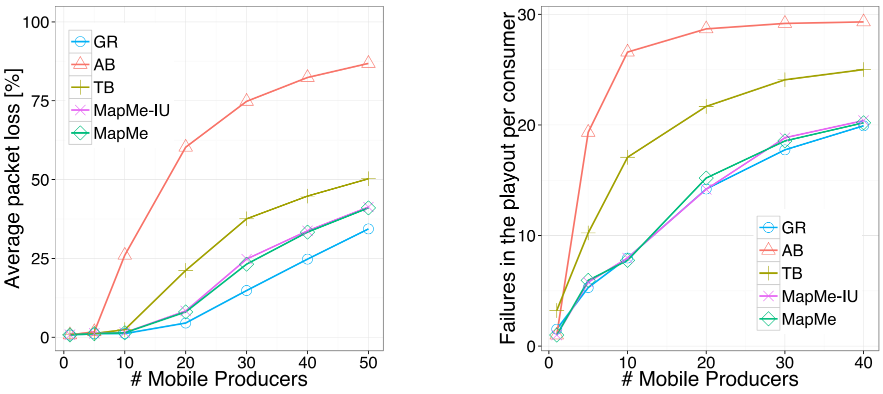
User Performance

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. 4.9(a) shows the average packet loss, while increasing the number of mobile producers in the system. As expected, increasing the number of active users in the

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

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



(a) (b)

Figure 4.9 – User performance: CBR average packet loss (a), Periscope playout failures (b).

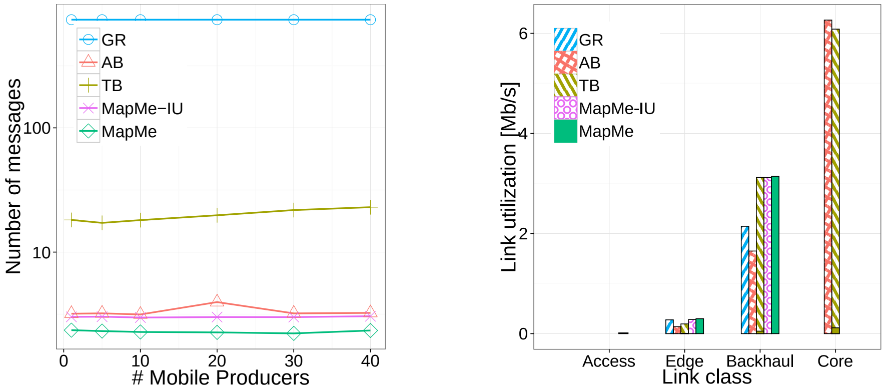
network has a negative effect on performance, because links are more congested and routers start to lose packets. However, as shown in Fig. 4.9(a), the performance of *MAP-Me* and *MAP-Me-IU* is close to the ideal GR, while TB leads to higher loss rate and with AB, we observe an even more rapid increase 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 that the user notices in the play-out of the video stream for the real-time video streaming (Periscope-like). Fig. 4.9(b) shows the number of failures in the video play-out that each consumer encounters in 4 min. As in the CBR case, when the number of mobile producers increases, the performance of the system degrades. Again, AB concentrates all traffic on a single node, the anchor, thus giving rise to congestion. In contrast, distributed protocols such as *MAP-Me* are able to better distribute traffic over the network and thus better cope with larger number of users. For the same reason, TB performs better than AB, but worse than *MAP-Me*/GR. Indeed, sending traces to the anchor forces traffic towards upper layers in the network, preventing substantial traffic offload at the edge.

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

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.4.10(a) reports the overhead, computed as the number of messages exchanged in the network at each handoff, whereas Fig.4.10(b) 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.



(a) (b)

Figure 4.10 – Network Cost: CBR overhead (a) and Periscope link utilization (b).

Network topology and Mobility: Trace-based simulation have been run with pedestrian mobility and a tree-like network topology [82]. Results show the same behavior for vehicular and pedestrian mobility, while in the case of tree topology TB and *MAP-Me* have similar packet loss (due to higher chances of congestion at the core of the network).

4.6 *MAP-Me* and routing

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

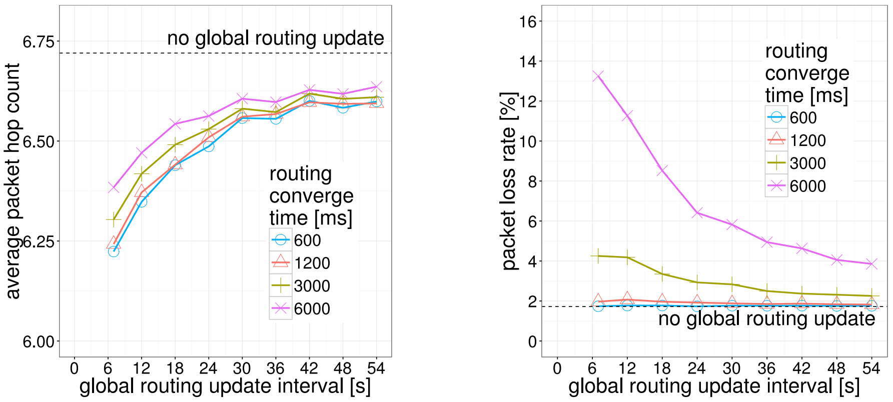
Proposed Solution: Our proposal makes minimum assumptions on properties of the routing protocol: (i) the routing protocol is *link-state* so that every node gets a sense of routing convergence state; (ii) every router maintains a counter R_{seq}, incremented each time a non-duplicated routing message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

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

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

Evaluation: We now illustrate the behavior of the modified algorithm, and analyze the effect of routing updates frequency on system performance. We consider the previous baseline scenario with 1 pair of mobile nodes, and a speed of 10m/s. The producer triggers a new routing update with varying frequency. Routing convergence time obviously impacts performance significantly. It is generally considered that link-state IGP (Interior Gateway Protocol) convergence time is in the order of several seconds. While [91] demonstrates the possibility for sub-second convergence time for large ISP (Internet Service Provider) networks by leveraging techniques like fast flooding and incremental FIB updates, it is not widely deployed. We thus reasonably assume the routing convergence time lies between sub-second and several seconds. In the evaluation we choose between 600ms and 6s.

Figure 4.11 illustrates the trade-off in setting the routing update frequency. Obviously, more frequent updates allow for shorter paths as they are re-optimized more often (Fig. 4.11(a)). However, instabilities due to routing at global scale lead to long-lasting un-



(a)

(b)

Figure 4.11 – *MAP-Me* and routing. Effects of routing update frequency on performance: (a) Packet loss rate. (b) Path stretch.

reachability of the producer after he moves, and thus a high packet loss rate (Fig 4.11(b)). Routing updates should thus be limited or triggered carefully, for instance in periods of producer stability (e.g., based on mobility prediction). Nevertheless, Fig 4.11(b) shows also that when routing converges in sub-seconds, the interaction with *MAP-Me* runs smoothly and without substantial loss in performance.

4.7 Conclusions

Native support for mobility management at the 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 is still lacking in ICN: 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 an ICN context. The contribution of this thesis chapter is twofold. First, we looked at CCN/NDN, two prominent ICN architectures, and define *MAP-Me*, an anchor-less model for managing micro-level (i.e., intra autonomous system) producer mobility even in the presence of latency-sensitive traffic. By design, *MAP-Me* is simple as it only leverages CCN/NDN forwarding plane and reactive notifications to the network, is lightweight in terms of required signaling messages and, to our knowledge, the first one with proven guarantees of bounded stretch and overall correctness for the forwarding update process. Second, we opensourced a simulation framework on top of NDNSim 2.1 that offers model-based and trace-driven consumer/producer mobility patterns over many topologies, integrated anchor-based and trace-based approaches, a global routing approach as well as a reference implementation for *MAP-Me*. Evaluation considers 802.11n access in small cell outdoor settings and proves WiFi can support mobility using CCN/NDN in general settings.

The reported results confirm our initial objectives and show that *MAP-Me* optimally offloads the infrastructure from communications that are local. All other approaches making use of an anchor, which in practice is also the network gateway, can be optimized only if traffic is non local. Instead, the current proposals 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.

Chapter 5

Security in Producer Mobility

5.1 Introduction

In the previous chapter, we presented our protocol, *MAP-Me* to efficiently manage ICN producer mobility. However, as mentioned in chapter 1, deploying *MAP-Me* without security mechanisms in place can expose mobile producers under the *prefix hijacking attack* (i.e., an attacker diverts consumer requests by forging Interest Updates with another producer's prefix). Such a basic attack can be a first step to enable further attacks described in [15, 16, 18]. As a result, a security mechanism to prevent prefix hijacking attacks becomes mandatory for *MAP-Me* . Moreover, we find that this security issue holds not only for *MAP-Me* , but also in general for any other trace-based producer mobility protocols in the ICN literature(e.g., KITE [13]).

Therefore, in this chapter we focus on the design of a security mechanism that can protect *MAP-Me* as well as other trace-based producer mobility protocols from prefix hijacking attack. Throughout the chapter, for simplicity we refer to both *MAP-Me* and other trace-based protocols as trace-based protocols.

To prevent prefix hijacking attacks to trace-based protocols, we apply an one-way hash-chain to the design of a prefix attestation protocol for the producer to prove to the network its right to express Interest Updates for a given prefix. The contributions are summarized as follows:

- Our protocol is fast and lightweight compared to a signature based mechanism adopted in most of the prefix attestation proposals [13, 56, 57, 58, 59, 60]. Results show that our lightweight approach maintains 90% of the original goodput (i.e., the maximum number of regular interests processed by the router per second excluding Interest Updates.), while in the signature approach the goodput drops close to 0% even with a small fraction of Interest Updates. In terms of storage requirements, our protocol only requires on the order of tens of megabytes on each router to manage billions of mobile producers.

- it can run unchanged on off-the-shelf hardware deployed at network access (e.g., micro, nano, small 4G/5G cells as well as Wifi access points) and at the mobile core network, whereas the traditional signature based approach may suffer from high mobility events (see section 5.5).
- our proposal can be applied to secure any trace-based producer mobility protocol in ICN literature, including *MAP-Me* .

The rest of the chapter is organized as follows: Section 5.2 presents the high level design of our prefix attestation protocol. Section 5.3 describes the details of our proposal. Section 5.4 presents the security considerations of our protocol and Section 5.5 evaluates its performance in terms of computation and storage overhead. Section 5.6 discuss further issues with our security protocol. Section 5.7 concludes the chapter.

5.2 Prefix Attestation Protocol Design

We design our protocol on top of the proposed tracing-based mobility protocols (*MAP-Me* [46] and KTIE [13]), extending them by: (i) introducing *bootstrap phase* that authenticates a mobile producer when it first connects to the network; (ii) adding a *Secure Interest Update Validation* mechanism. Instead, we leave the underlying tracing-based mobility protocol to decide when and how to stop propagating Interest Updates.

The bootstrap phase will authenticate the producer to the network, giving evidence of its entitlement to announce its prefix(es). We want to highlight that our producer authentication is different in principle from the user authentication employed in many mobile networks (e.g., 3G/4G and Wifi). We recall that the latter is used to allow (or deny) a user to connect a device to a mobile network and it does not give any insight into what a device can publish. Moreover, we believe that user identities and producer identities should be managed separately. A user might own different devices authenticated to the network with the same user identity(e.g., the case in the password-based authentication mechanism adopted in many Wifi networks), while such devices might need to announce different sets of prefixes due to the producer applications they run. For instance, considering the case of a user owning a mobile phone and a laptop. The mobile phone can run VoICN application generating content for the prefix $\backslash$`A`, while the laptop does not run the same application.

In this case, the laptop should not be entitled to announce the prefix `\A` until the VoICN application is installed. If this rule is not guaranteed, malware running on the laptop might announce prefix `\A` and perform the attacks described in [16, 17, 18] to hijack the user's mobile phone. Authenticating user identities and producer identities separately gives flexibility to cover the aforementioned case.

The secure Interest Update validation mechanism guarantees that only the entitled producer can generate a legitimate Interest Update for the prefix(es) under its responsibility. Moreover, it allows each router to verify the validity and freshness of the Interest Update through attestation: every Interest Update carries a fresh proof that it has been generated by the entitled producer. In the following we present the system model and the threat model of our proposal.

5.2.1 System Model

Our protocol involves four network entities: a registration server, core router, edge router and mobile producer. Figure 5.1 depicts the system model considered throughout the chapter.

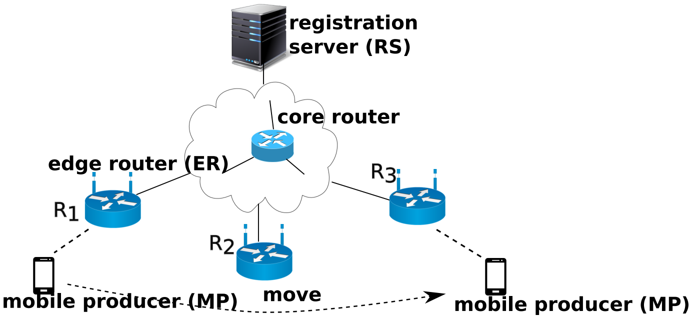


Figure 5.1 – System Architecture

The registration server is mainly responsible for: (1) authenticating the mobile producer and verifying the ownership of the prefix(es) it announces, (2) generating and distributing to the network the necessary cryptographic material to validate Interest Updates for the producer's prefix(es). We call such cryptographic material the **security context**. We assume that the **security context** of a given prefix can be stored as an additional field in the forwarding state of the same prefix (e.g., PIT entry or FIB entry). We define a mobile producer as a mobile device storing a producer identity entitled to publish content under one or more prefixes. A pair of private/public keys is associated with the producer identity and used to sign/verify content. Finally, we consider the network to be composed of edge and core routers forming a single *Autonomous System* (**AS**).

We consider the access to the network to be heterogeneous (e.g., edge router can be 4G/5G cell or WiFi access points). Moreover, every mobile device is in possession of the user credentials to connect to the network infrastructure. Once the mobile device is authenticated by the edge router, the communication between the mobile device and the edge router is considered secure (i.e., encrypted and integrity protected).

5.2.2 Threat Model

In this work we consider an attacker is an entity that can connect to the network, e.g., the attacker buys a valid SIM. The attacker targets genuine mobile producers and aim to generate legitimate Interest Updates for the prefixes used by its victims.

We assume that edge routers can be compromised by the adversary, while core routers and the registration server are more robust to attack. The rationale behind this assumptions is that edge routers (e.g., Wifi AP) are low cost devices that are often placed in unattended environments lacking physical protections, whereas core routers and registration servers are often under the control and protection of the mobile operators. In fact, these assumptions are actually consistent with those made in existing mobile networks [92].

Moreover, we assume an intrusion detection mechanism is in place and it is able to detect a compromised edge router [93]. As soon as a compromised edge router is detected, it is disconnected from the network along with the devices connected through it. Finally, we assume that the attacker can access the information stored in the compromised edge router.

5.3 Prefix attestation

Our proposal exploits route versioning to verify that an Interest Update is fresh and not the result of a message replay. In fact, our route versioning can be used together with *MAP-Me* presented in chapter 4 of the thesis or it can be considered an additional mechanism. In particular, for a given prefix a router stores a sequence number in the corresponding forwarding state of such prefix. A router considers an Interest Update fresh only if the interest carries a greater sequence number than the one stored in the router.

We make use of a **one-way hash-chain mechanism** to guarantee that a producer can generate Interest Updates only for its own prefix(es). One-way hash-chain is a simple mechanism initially proposed by Lamport [94] as a replacement for password-based user authentication and authorization (e.g, a user `A` that wants to log-in to a server `B` for accessing its service). The mechanism works as follows: user `A` generates a sequence of values by applying n times a cryptographic hash function H to a random value s as depicted in Figure 5.2. The value s is called *root of the chain.*

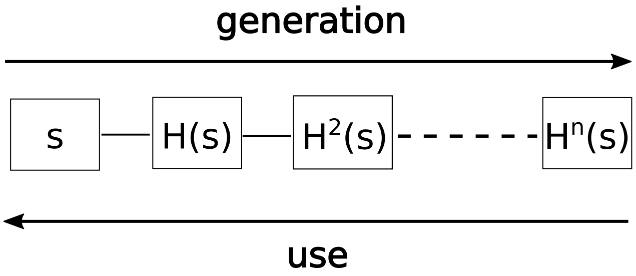


Figure 5.2 – Hash Chain illustration.

We assume that initially `B` receives $H^n(s)$ and is assured of it genuineness (i.e., equal to the result of applying the hash function n times to s). Subsequently when user `A` wants to log-in on `B`, it sends $H^{(n-1)}(s)$ to `B`. `B` simply checks that $H(H^{(n-1)}(s)) = H^n(s)$. This proves that only `A` could have generated $H^{(n-1)}(s)$.

We use the hash-chain mechanism in the following way. We associate a hash-chain for each producer's prefix such that: $H^{(n-i)}(s_p)$ is the hash value corresponding to the forwarding state for the prefix p with sequence number i. Therefore, $H^{(n-i)}(s_p)$ is the **security context** for forwarding state of the prefix p and sequence number i. A router considers an Interest Update for a prefix p valid only if: (1) it carries a hash value $H^{(n-j)}(s_p)$ where j is the corresponding sequence number; (2) the security context for p in the router is $H^{(n-i)}(s_p)$ and $j > i$; (3) the equality $H^{(j-i)}(H^{(n-j)}(s)) = H^{(n-i)}(s_p)$ holds (i.e., the result of applying hash to what is carried in the Interest Update j-i times matches the value in the security context). As we will see in section 5.3.1 and section 5.3.2, in general these conditions will easily hold for Interest Updates generated by a legitimate producer, and they will not hold for Interest Updates forged or replayed by an attacker. Table 5.1 reports the notation we use throughout the chapter.

RS	Registration Server
MP	Mobile Producer
ER	Edge router to which **MP** is connected
p	Prefix owned by **MP**
s_p	Root of the hash-chain for prefix p
n	Length of the hash chain of p
H	Cryptographic hash function
$H^{(n-i)}(s)$	Value of the hash chain for the sequence number i

Table 5.1 – Notation table.

In the following sections, we describe in more details the bootstrap phase and the secure Interest Update validation mechanism we introduced in Section 5.2.

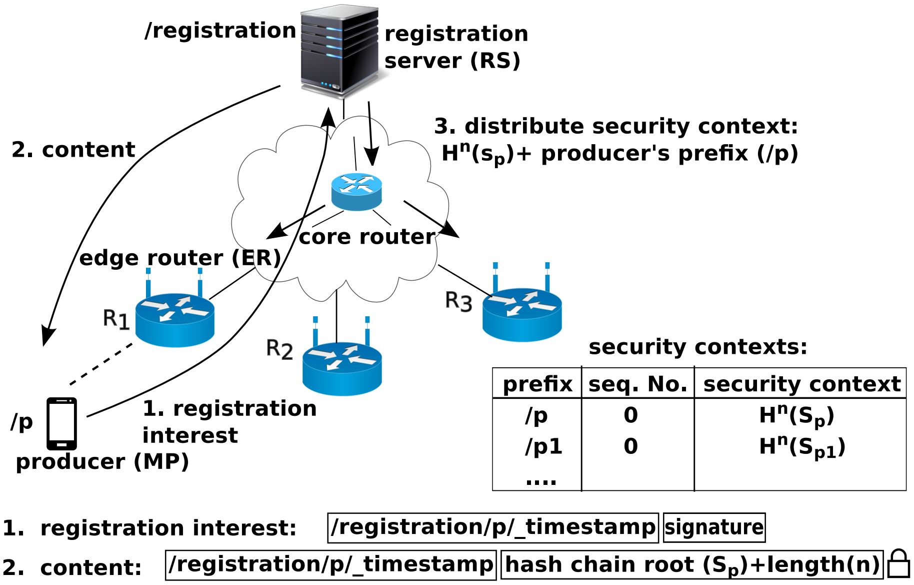
5.3.1 Bootstrap phase

The bootstrap phase and the messages sent (1-3) illustrated in Figure 5.3. The bootstrap starts with **MP** authenticating and proving to **RS** its right to announce its prefix p. To achieve this, (1) **MP** issue an interest to **RS** with **RS**'s name prefix (i.e., /registration) as name prefix and producer's prefix p as a name component. The detailed interest format is shown in the bottom of Figure 5.3. Such interest, we call registration interest, will be signed with the **MP**'s private key and it will carry a fresh timestamp. Once **RS** has verified the signature and checked the freshness of the registration interest, (2) **RS** sends back to **MP** a content containing s_p and n, the root and the length of the hash chain for p. The content payload will be encrypted with **MP**'s public key so that only **MP** can access the root of the chain. The detailed data packet format is shown in the bottom of Figure 5.3. Then, as the third step shown in Figure 5.3, (3) **RS** will distribute p's security context (i.e., the hash value corresponding to the sequence number 0) to the whole network. It is worth noting that while in *MAP-Me* , the forwarding states are permanently stored in FIBs, KITE [13], the forwarding states are not permanently store (i.e., they are temporarily stored in PIT). Therefore in case of KITE, we require that routers that do not have any forwarding state for p will drop the security context. We envision that the security context distribution can be performed through a routing protocol update message.

The protocol ends with **MP** generating the full hash chain and issuing an Interest Update with sequence number 1 to the edge router to which it is connected. This Interest Update is important to prevent a prefix hijacking attack in this step, as will be explained later in Section 5.4.1.

5.3.2 Secure Interest Update Validation

Figure 5.4 shows our Secure Interest Update Validation mechanism. According to the tracing-base protocols, at every mobility event (i.e., when **MP** connects to a different **ER**) **MP** issues a new Interest Update. Our validation mechanism requires that the Interest Update carries a sequence number, monotonically incremented at every release, and the proof of validity. In the following, we describe our mechanism assuming **MP** releases an Interest Update with a newer sequence number j and the corresponding value of the hash-chain $H^{(n-j)}(s)$. We describe the verification steps performed at **ER**, although it has to be noted that every router, core or edge, receiving an Interest Update will perform the same verification steps described in the following.



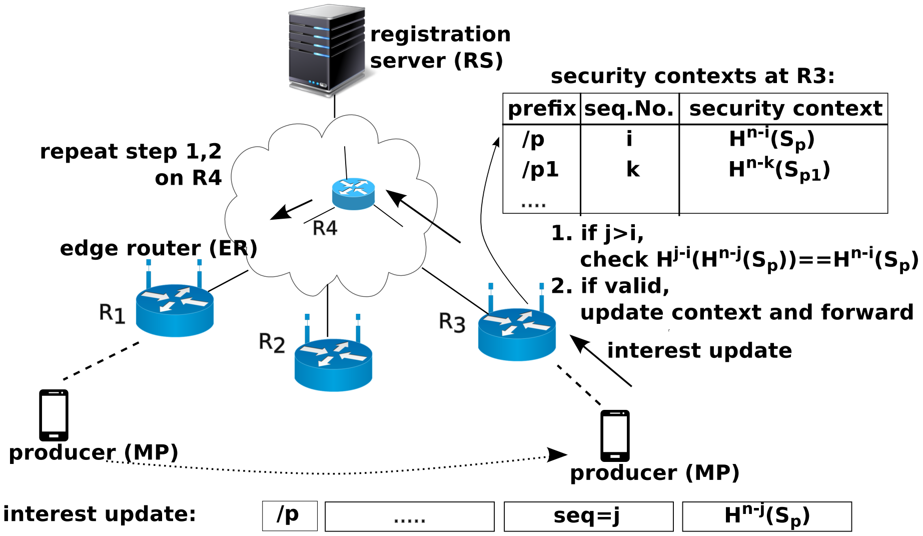
prefix	seq. No.	security context
/p	0	$H^n(S_p)$
/p1	0	$H^n(S_{p1})$
....		

1. registration interest: /registration/p/_timestamp signature
2. content: /registration/p/_timestamp hash chain root (S_p)+length(n)

Figure 5.3 – Bootstrap phase

Upon reception of the Interest Update, **ER** matches the name into its forwarding table to retrieve the current sequence number i of the forwarding state, as well as the related security context, i.e., $H^{(n-i)}(s)$. Then, **ER** verifies that $j > i$ and, if the inequality holds, it extracts $H^{(n-j)}(s)$ from the interest. Finally, **ER** verifies if the Interest Update is legitimate by comparing $H^{(j-i)}(H^{(n-j)}(s))$ to $H^{(n-i)}(s)$. If the two values match, **ER** updates the corresponding forwarding state and the security context to $H^{(n-j)}(s)$, before forwarding the Interest Update to the next router according to the mobility management protocol in place. Note that in Kite, a router might not have the security context for p. In this case we exploit the acknowledge message of KITE to let the anchor to send the corresponding security context. Interest Update validity will be check after receiving the security context. Figure 5.4 shows the verification step performed by **ER**. If the verification succeeds, **ER** forwards the Interest Update to the next (edge or core) router.

In this way, the attacker cannot issue a legitimate Interest Update without knowing a legitimate value in the hash chain for a mobile prefix. An invalid Interest Update will be dropped immediately at the edge router who receives it after performing the above Interest Update Validation procedure.



prefix	seq.No.	security context
/p	i	$H^{n-i}(S_p)$
/p1	k	$H^{n-k}(S_{p1})$
....		

Figure 5.4 – Secure Interest Update Validation

5.3.3 Prevent Replay attack

It is worth noticing that the verification steps described so far do not fully prevent prefix hijacking attacks. It only prevents attacks that do not know legitimate value of the hash chain for the target prefix. However, it does not prevent replayed attacks, where an attacker obtains an old but valid Interest update from a compromised **ER** and replays it through another **ER**. Since there is no guarantee that every router in the network has the most recent version of the forwarding state, outdated routers might accept as fresh the replayed Interest Update.

To solve this problem, we exploit a common property of all the tracing-base protocols: *an Interest Update always hits a router with the most recent version of the forwarding state.* Note that for the *MAP-Me* protocol this is formally proven. For Kite, an Interest Update always arrives at the anchor, which will always have the most recent security context because every Interest update reaches the anchor. This can be proven also for other tracing-based approaches. Therefore, any replayed Interest Update will hit one router with higher sequence number (i.e., the one with the most recent security context). Note that such a hit would not happen for a normal Interest Update whereas it must happen for a replayed Interest Update.

Thus the countermeasure for a replay attack can be designed as follows: if the check $j > i$ fails for a certain Interest Update, it means that such an Interest Update is old and received probably because of a replay attack. All the upstream routers that accepted such old Interest Update have a corrupted forwarding state. To restore the corrupted forwarding state, the "hit" router drops the received old Interest Update and creates a new Interest Update using the local security context. The new Interest Update is then propagated back to the upstream routers, updating and fixing their forwarding state (i.e., fixing the corrupted forwarding states produced by the replay attack). As a consequence, any further old (or replayed) Interest Update received by the **ER** will be dropped also.

5.4 Security Considerations

In the following sections we provide a security discussion regarding the threat model presented in Section 5.2.1. We show how an attacker will not be able to successfully complete the bootstrap nor the secure interest validation phase for a prefix p that does not belong to the attacker. Moreover, we discuss a possible denial of service attack that might exploit the design of our protocol. For that, we also propose a mitigation mechanism.

5.4.1 Preventing Prefix Hijacking attacks

We consider the case in which the attacker tries to pass the registration phase and deploy a security context for the prefix p. The attacker thus needs to generate a valid signature for the registration interest for p. The signature must be calculated with the private key of the producer that owns p. However, since the private key is secretly stored in the producer's device(s) and never transmitted to the network, the adversary cannot generate a new valid registration interest on its own. Its only chance to pass this step is to replay a valid registration interest. Recall that an attacker can compromise an edge router, and by reading the edge router's memory, it can obtain a valid registration interest. If such an interest has already been received by the **RS**, the attacker will not be able to pass the registration phase (the timestamp in the registration interest will reveal the replay attack). If the registration interest has not been received by the **RS** (e.g., due to congestion), the registration interest will be accepted. At this point, the attacker must be able to express a valid Interest Update to get the edge router to update its forwarding state. Because the attacker does not know the producer's private key, it will be unable to decrypt the content carrying the root of the chain for p. In the following we discuss that an attacker cannot generate a valid Interest Update without the root of the chain, thus neither complete the bootstrap phase nor pass the secure interest update validation phase.

To express a valid Interest Update for a prefix p, an attacker must be able to generate $H^{(n-i)}(s)$ such that $i > j$ and $H^{(n-j)}(s)$ is the latest value of the chain released by the genuine producer. In our design, we assume that the hash-chain is generated using a cryptographic hash function (e.g., SHA256) . The security properties of cryptographic hash functions (i.e., Pre-image resistance, Second pre-image resistance and Collision resistance) makes it infeasible to generate $H^{(n-i)}(s)$ without knowing any $H^{(n-k)}(s)$ where $k > i$ [95]. Because the producer releases the values of the chain in the reverse order, it is easy to prove by induction that either the attacker knows the root of the chain or it cannot generate $H^{(n-i)}(s)$.

5.4.2 Denial of Service Attacks

The validity check of Interest Updates might open a door to Denial-of-Service attacks at the edge routers. In particular, consider the case in which an attacker issues a non-legitimate Interest Update for p that hits a router r. The Interest Update carries a sequence number i such that $i \gg j$, and j is the sequence number of the forwarding state for p in r. To be able to detect the Interest Update as non-legitimate, r needs to hash $i - j$ times the security context associated with p. The greater the distance between i and j, the more hashes r will need to calculate. An attacker can use non-legitimate Interest Updates with high sequence numbers and keep the router busy calculating hashes, thus provoking a DoS attack to the other connected producers.

To prevent the above Denial of Service attack, we fix a threshold t so that every router will drop Interest Updates whose sequence number i is grater than $t + j$. However, using a threshold-base approach brings another problem. A routers with an old version j might drop valid Interest Updates because the sequence number i they carry is greater than $t + j$. This might happen if a mobile producer moves frequently between a small subset of edge routers. To avoid this problem we propose to exploit a routing protocol to maintain the security context of the routers loosely synchronized (i.e., limit the maximum difference between sequence numbers of security contexts of any pair of nodes in the network by periodically synchronizing them via routing messages). Every routing update will carry security context of the router generating it, along with the regular routing information. We leave for future work the full design of the mechanism and the evaluation of the overhead introduced by it.

5.5 Evaluation

In this section we evaluate the overhead introduced by our protocol in both routers and mobile producers. We focus our evaluation on the hash-chain verification mechanism because it is expected to be the most computed operation of our protocol (i.e., at every mobility event). In particular, we provide an analysis of (i) computational overhead and (ii) additional storage cost involved in the routers. We leave for future work the evaluation of communication overhead (involved in keeping the sequence number of security contexts in the network loosely synchronized via routing) as well as the overhead of bootstrap phase. As it involves more sophisticated trade-off (between bandwidth overhead and computational overhead) in setting the frequency of synchronization or bootstrap phase invocation.

We compare our hash-chain verification with the signature-based verification adopted in most prefix attestation proposals [56]. We consider the protocol to be the same in both approaches (i.e., both issues an Interest Update that will be verified by each router). In the hash-based verification an Interest Update will carry a hash value while, in the signature-based approach, the Interest Update will be signed with the producer's private key. Results show that our approach in mobile networks can reduce both computational and storage overhead. In particular, the lower computation overhead of our mechanism allows a router to maintain 90% of the original goodput (i.e., with no verification) while the signature approach drops it close to 0%. Recall that here *goodput* is defined as the maximum number of regular interests processed by the router per second excluding Interest Updates. Moreover, our approach reduces by 66% the storage overhead introduced by the most expensive signature based approach [13].

5.5.1 Computational Overhead

To evaluate the computational overhead, we quantify the time required to perform a verification with both hash-based and signature-based approaches. Then, based on a simple analytical model, we derive their impact on overall router goodput as the producer mobility rate increases.

The time required to verify an Interest Update can be characterized as the sum of retrieving the security context for the Interest Update and the verification time. Retrieving the security context only adds a negligible time. In fact, the security context is stored together with the forwarding state in the corresponding table and it can be retrieve during the regular lookup for processing the interest. For the signature verification we assume that the producers' public keys are in the security context. Therefore, the latency for certificate chain traversal to retrieve a public key is not included in the verification delay, which is optimal for signature-based approach. Under the above assumption, the verification delay by both approaches will be dominated by the time to perform either the hash-chain verification or the signature verification.

We evaluate the two verification mechanisms considering the hardware adopted on edge routers. Edge routers are less capable than core routers and so are more sensitive to computational overhead. We use the Cavium Octeon MIPS64 as the reference for a common platform for carrier-grade wireless access routers (LTE and/or WiFi) and we derive the verification time based on the openwrt [96] benchmark result for a MIPS 64 processor [97].

Table 5.2 reports the time required for hash-chain verification against that for signature-based verification. From Table 5.2 we observe that the computational overhead incurred by hash-chain verification is about three orders of magnitude smaller than the computational overhead incurred by using signature verification. It is interesting to note how a single hash can be calculated in a fraction of micro seconds, meaning that a router can apply a hash function to a packet and still process such packet at line rate. This is important to prevent Denial Of Service attacks that exploit the computational overhead. The signature verification cannot be done at line rate, which opens the door to Denial Of Service attacks.

hash chain based		signature based	
SHA256	MD5	RSA	DSA
3μs	0.8μs	4700μs	5710μs

Table 5.2 – Verification delay.

Then, we investigate the impact of this verification delay on edge router's goodput increasing producer's mobility rate. We calculate the edge router's packet goodput from a model similar to the one used in [98]. Considering η as the percentage of Interest Update packets in the total number of packets received by a router, the goodput (in packets/s) is calculated as:

$$goodput = \frac{1-\eta}{\tau_{process} + \eta \times \tau_{verif}} \tag{5.1}$$

where $\tau_{process}$ is the average processing time for a normal packet, τ_{verif} is the verification delay for an Interest Update message. For edge router Cavium Octeon we consider a maximum overall packet throughput of 0.25Mpps, thus $\tau_{process}$= 4μs. For τ_{verif} we apply the number reported in table 5.2.

From equation 5.1 we can compute the edge router's goodput in packet/s. Figure 5.5 shows edge router's goodput performance with increasing number of Interest Update messages from the mobile producer. The amount of Interest Update messages is again represented by the percentage of Interest Update message (i.e, η). The goodput is presented in millions of packets per second (i.e., Mpps).

We see that with a signature-based approach, which relies on RSA or DSA, the goodput performance can be severely impacted by the amount of Interest Update messages. In particular, the goodput drops almost to 0 with only about 5% of Interest Update received. In contrast, with our hash-chain mechanism, which relies on SHA256 or MD5, when receiving the same amount of Interest Update messages, we can maintain 95%-98% of the

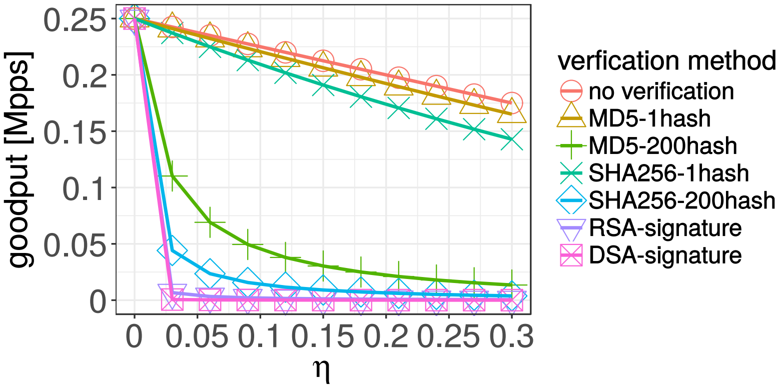


Figure 5.5 – edge router goodput

original goodput (i.e., of that with no verification performed) if only one hash computation is required per Interest Update (as shown in Figure 5.5 as SHA256-1hash and MD5-1hash). Overall, our mechanism achieves 80%, with the slower SHA256, and more than 90%, with the faster MD5, of the original goodput with 1 hash computation per Interest Update. Figure 5.5 also shows that with around 200 hash computations per Interest Update our approach shows comparable performance w.r.t the signature-based approach. Maintaining the context state in the routers updated allows our mechanism to perform the best and to achieve a substantial gain when compared with the signature based approach.

The computational cost at the mobile producer is considered to be negligible. We assume that the producer stores the full chain at the bootstrap phase. Therefore, during any handover it will not need to do any hash computation. To reduce the computational overhead of the hash-chain calculation we can adopt the mechanism proposed by Coppersmith and al. [99]. Such a mechanism provides a computation complexity of $\frac{1}{2}log_2n$ for calculating the full hash chain.

5.5.2 Additional Storage Cost

Every edge and core router needs to maintain a security context for each of its forwarding state. Since our mechanism stores the security context in the same structure containing the forwarding state (e.g., PIT or FIB), the storage cost can be calculated as:

$$Storage\_cost = N_{forwarding\_entry} \times (S_{security\_context} + S_{seq}) \tag{5.2}$$

where $N_{forwarding\_entry}$ is the number of entries in the forwarding structure, $S_{security\_context}$ is the size of the crypto material needed to perform the verification (i.e., either a hash or a public key), S_{seq} is the sequence number corresponding to the forwarding state version. For the hash-based mechanism we assume that $S_{security\_context}$ = 32 bytes while for the signature-based mechanism $S_{security\_context}$ = 256 bytes for both RSA and DSA.

Figure 5.6 shows how the storage cost varies with respect to the number of active mobile producers. For a mobile EPC (Evolved Packet Core) network the number of mobile users is on the order of 1 million. If we consider the worst case scenario for the storage cost, i.e., every router has an entry in the forwarding state structure per mobile user, we can see that store cost is about $50MB$ at each router. Modern router device can easily store such an amount of data.

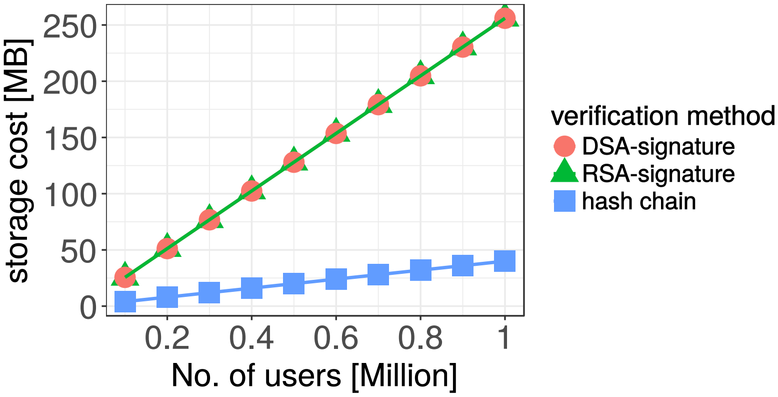


Figure 5.6 – storage cost at each router

To evaluate the storage cost for **MP**, we consider the proposal by Coppersmith and al. [99]. This mechanism requires storage for $log_2 n$ number of hashes, where n is the length of the hash-chain. Therefore, if we assume n equals 1 billion and the size of a single hash is 32 bytes, we only require less than 1KB to store the chain. While this requires more space than storing the single private key for the signature verification approach, we argue that 1KB is negligible overhead for most of the currently available devices (including IoT devices).

5.6 Discussion

In this section, we discuss some of the practical issues related to our prefix attestation protocol. In particular, we focus on two problems: 1) how do we deal with prefix aggregation in the network? 2) why prefix attestation is still needed considering the fact that mobile users' identity is often well authenticated at layer 2 by LTE/Wifi network access control mechanisms? We elaborate on each of the questions and our answers to them in the following sections.

5.6.1 Mobile Prefix Aggregation

So far, in the description of our proposal, we have not considered the possibility of mobile prefix aggregation: we focused on the scenario where routers maintain a separate security context for each mobile prefix. However, in practice, prefix aggregation can be important if we need to scale to a large number of mobile producers in the network. Even though the current trace-based protocols including our proposal of *MAP-Me* and other's KITE proposal do not support prefix aggregation in mobility management, we envision that they will be extended with such a capability in the future to enhance their scalability.

In this section we present our idea on how our prefix attestation protocol can be applied in case mobile prefix aggregation is present. In fact, it only requires that the security contexts for all the aggregated prefixes are stored together with the resulting aggregated prefix in the router and make them distinguished per producer (e.g., each associated to a producer id). The producer id can be the unique name component (or its hash) that is in the producer's name prefix but not shared with other mobile prefixes. Therefore, if the tracing-based protocol is eventually extended to support prefixes aggregation, our prefix attestation protocol will be easily adapted to support prefix aggregation too.

5.6.2 LTE/Wifi Authentication not Enough for Prefix Attestation

At first glance, the following naive approach leveraging the existing LTE/Wifi access control mechanism could also work to authenticate mobile producers into the mobile networks (hence, no need of our prefix attestation protocol): the network first binds a set of prefixes to a user identity (e.g., LTE sim card or Wifi user credentials). Once the user is authenticated at layer 2 (e.g at LTE or Wifi network), the network knows which prefixes the user owns (i.e., due to the bindings) and validating the Interest Update will become trivial by verifying if the name of the Interest Update is owned by the mobile user.

However, we argue that this approach of equating user identity and producer identity is not flexible enough to be used in common scenarios. Recall that we define produce identity as *a mobile device storing a producer identity entitled to publish content under one or more prefixes.* In fact, it is common that a user authenticates many devices to the network using the same user identity, while such devices need to announce different sets of prefixes due to the different applications installed. For example, let us consider the case of a user owning a mobile phone and a laptop. The mobile phone runs a skype application generating content for the prefix /skype/alice, while the laptop does not run the same application. Both of the devices can authenticate to the same network (e.g., by using the user's Wifi credentials) In this case, the laptop should not be entitled to announce the prefix /skype/alice until the skype application is also installed. If this rule is not guaranteed, an attacker can perform attacks described in [16, 17, 18] by installing malware on the laptop that make it announce /skype/alice. We believe that user identity and producer identity should be managed separately and independently to allow flexible protocol design at different layers and support common networking scenarios.

5.7 Conclusion

The ICN communication model offers native support for mobility at the network layer that previous work in the ICN literature have leveraged to define name-based mobility management protocols to handle consumer and producer mobility. Previous work has focused mainly on producer mobility management, more challenging than consumer mobility, and specifically on forwarding mechanisms to guarantee reachability of the name prefix(es) of the producer after each movement.

In chapter 4, we have presented a solution to address producer mobility management in ICN. To complement *MAP-Me* as well as other trace-based proposal in ICN literature which naively manage producer mobility without considering security, in this chapter we complement *MAP-Me* such work by looking at the security implications of producer mobility. We presented a protocol for prefix attestation based on hash-chaining to protect against prefix hijacking attacks that may occur during mobility updates. The protocol targets tracing-based mobility management solutions and it is lightweight and fully distributed. We also proposed countermeasures to a type of replay attack, which is feasible particularly in mobile network but not taken into account by existing signature-based proposals in the literature. Our protocol can run unchanged on different hardware deployed at operational network access (e.g., LTE or WiFi). Initial evaluation results confirm that our protocol introduces minimal computational and storage overhead to secure tracing-based proposals.

We have proposed a preliminary idea to prevent denial of service attack introduced by our prefix attestation protocol. However, it could still mistakenly drops legitimate interest update messages. We leave as future work the full design and the implementation of our prefix attestation mechanism that can prevent Denial of Service attack. Moreover, to keep low computational overhead of our hash chain mechanism, we still need to design a mechanism to keep the sequence number of security contexts in the network loosely synchronized. We envision this can be done by exploiting the routing protocol and we leave it as our future work.

Chapter 6

Congestion Control in Mobile ICN Networks

6.1 Introduction

Since we have addressed producer mobility as well as its associated security at the network layer, we can proceed to investigate ICN's transport layer in mobile environments. While existing congestion control schemes in the ICN literature have focused on exploring its new features such as caching and multi-path, they often assume a simplified scenario of wired networks. In fact, the resulting schemes can suffer from performance degradation when applied in mobile networks. The fundamental reason is that for mobile networks most of the schemes can misinterpret mobility/wireless loss as congestion signals, leading to unnecessary control actions and throughput degradation.

To alleviate the issue of congestion control in mobile ICN networks, in this chapter we propose two mechanisms to facilitate ICN's congestion control: (1) *WLDR*: Wireless Loss Detection and Recovery; and (2) *MLDR*: Mobility Loss Detection and Recovery. The basic idea of MLDR/WLDR is to hide non-congestion related losses from ICN's transport layer. We leverage ICN's in-network processing capability to efficiently detect and recover non-congestion losses at the network layer, such that they become transparent to congestion control. More specifically, we distinguish the nature of losses(i.e. due to wireless or mobility) and manage them separately. For wireless loss, we design MLDR to promptly identify and recover wireless losses at access points. For mobility loss, we design MLDR to prevent losses due to consumer/producer mobility through explicit network notification and dynamic on-the-fly request re-routing. Notice that in the default ICN design (without WLDR/MLDR), to cope with mobility/wireless losses, the baseline technique of retransmission by timer at the receiver is used to recover the loss.

Our approach of leveraging ICN's in-network processing is beneficial for 2 reasons: (1) it allows us to disentangle and separately manage loss of distinct natures. (2) it enables fast loss detection and recovery in the optimal place in the network.

For evaluation, we setup a realistic wireless simulation environment in ndnSIM using Wi-Fi 802.11n connectivity and assess MLDR/WLDR's effectiveness to facilitate congestion control. We compare MLDR/WLDR with consumer-driven alternatives based on timers [75] or on Explicit Loss Notification (ELN) [76].The results show a significant reduction in terms of flow completion time or request satisfaction time, i.e. the time between the first request emission and the corresponding data packet reception at the consumer, which is particularly important in case of latency-sensitive applications. In addition, our proposal provenly removes any dependency on network/application timers that existing ICN solutions rely on and can not set properly.

We will describe our mechanism under the framework of CCN/NDN in this chapter even though it is generally applicable to other ICN architecures as well.

The remainder of the chapter is organized as follows: we introduce our design of WLDR/MLDR in Sec.6.2 and describe its implementation in Sec.6.3. The evaluation is gathered in Sec.6.4, while conclusions are reported in Sec.6.5.

6.2 Design

In this section we present our proposal for *WLDR* (in-network wireless loss detection and recovery, Sec.6.2.1) and *MLDR* (mobility loss detection and recovery, Sec.6.2.2), respectively. While addressing two separate problems, their design shares the same principles. We aim for a solution that

- is *layer-2 agnostic*, applicable to any wireless medium irrespectively of specific access characteristics;
- *differentiates the nature of losses*, to distinguish and separately handle wireless channel losses from buffer overflows due to congestion or from losses due to mobility timeouts;
- leverages *Explicit Loss Notification* (ELN) to decouple the point in the network where detection and recovery of losses are performed;
- leverages *fast in-network loss detection and recovery* in sub-round trip time scale or before the expiration of an Interest timer at the consumer.

Besides reactive detection and recovery, the rationale behind in-network loss management is to make the NDN/CCN data plane robust to losses and insensitive to pending interest table (PIT) timer management, which is a non trivial operation in very lossy environments. Also in-network loss management reduces misguided congestion and flow control decisions at the consumer which ultimately are responsible for reliable transport. Finally, in the design of WLDR/MLDR, we aim to build lightweight mechanisms in terms of signaling overhead, as well as additional state at routers and complexity. Implementation considerations are reported in Sec.6.3.

6.2.1 Wireless Loss Detection and Recovery

Two neighbors are interconnected using adjacencies, called faces in NDN/CCN. An adjacency is a unicast bidirectional channel between two nodes. It may also be established among nodes connected to the same broadcast medium, as in IEEE 802.11 for instance. WLDR is implemented at face level and introduces an additional sequencing on packets to detect losses. Sequentiality is then guaranteed on a per-face basis. In the same way, using multiple wireless faces in parallel, the stream of packets generated by each face is associated with a different sequencing. WLDR is not able to detect losses end-to-end, such a task being the responsibility of the transport protocol. However, by applying WLDR at each hop, WLDR can be simply extended to the multi-hop wireless case.

WLDR introduces new fields in the headers of Interest and Data packets (see Sec. 6.3) to store the sequence numbers used by the algorithm. These values have limited scope on a single wireless link and they may be modified every time a packet traverses a new wireless link implementing WLDR. It is important not to confuse the sequence number used by WLDR with the one that may be present in the Interest/Data names. Next we illustrate WLDR's basic functioning (i.e., loss detection and loss recovery) by walking through an example in Fig. 6.1.

In Fig. 6.1, a consumer is connected to a wireless Access Point (AP here after, with no reference to any specific wireless technology) through a wireless link and sends Interest packets to the AP to request a specific content item. To keep track of Interests sent through a given face, the consumer maintains a counter per-face indicating the sequence number for the *next* Interest to be sent (hence, indicated with *next* in Fig. 6.1, Alg.3). Such a sequence number is also added in the header of every Interest to be sent, then the counter is incremented by one. In Fig. 6.1, *next* is equal to 3, i.e. the consumer will associate the label 3 to the next Interest to be sent and update it to 4.

Loss Detection. The sequence number in the Interest is used by the AP to reconstruct the sequence of packets and so detect potential losses. To verify whether the incoming Interest is the expected one, the AP keeps an *expected* sequence number value. Upon Interest reception, it compares the sequence number in the Interest packet with such expected value. If they coincide, the AP simply updates its expected value (increasing by 1). Coming back

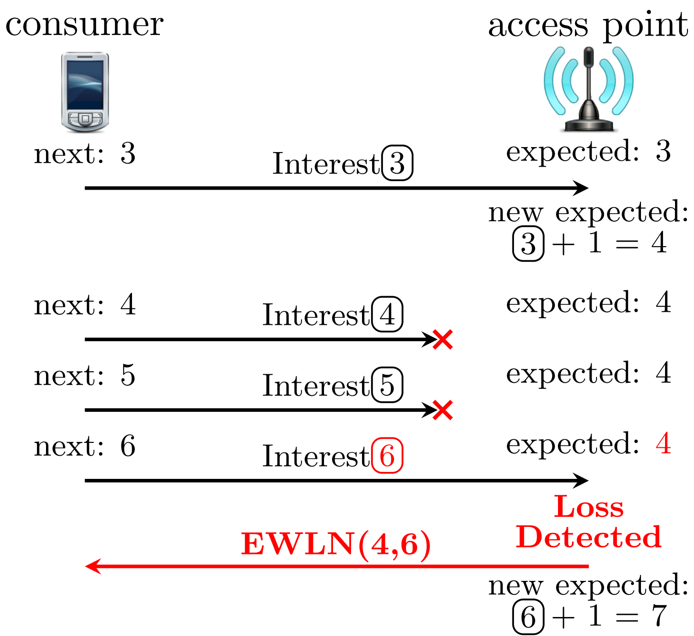


Figure 6.1 – WLDR wireless loss detection

to the example in Fig. 6.1, the expected value at AP is 3 . Upon the reception of an Interest with the same sequence number, the AP updates the expected value to 4.

If the expected value and the Interest sequence number are not the same, the AP detects a loss. This is the case in Fig. 6.1 at reception of the Interest packet with sequence number 6.

Fig. 6.1 describes WLDR between a wireless consumer and an AP. However, WLDR applies to any pair of nodes connected through a wireless link without requiring a distinction between consumer/producer or wireless node/AP. In contrast, previous mechanisms depend on the role of the wireless node. For example, two close algorithms for IP networks are presented by Balakrishnan et al. [100] and Biaz et al [101], the first one working in case of wireless producer only, while the second one working for wireless consumer node only. The only distinction of roles required by WLDR is the one between sender and a receiver node, since WLDR is a directional protocol. The sender is responsible to enumerate the packets and recover losses, while the receiver checks the sequence number in the packet to detect losses and to notify the sender. A node can be a sender and a receiver at the same time and this is the case for a bidirectional link.

In addition, WLDR is implemented at the face level and does not keep any per-flow information, because it does not make any distinction between packets from different applications, nor between Interest and Data packets. As a consequence, mechanism failure at the base station does not lead to connection disruption, as it is the case for alternative proposals in the TCP/IP world (I-TCP [64] or WTCP [102]).

Loss Distinction. The mechanism described above distinguishes wireless channel losses from losses due to mobility by virtue of the sequence number labeling performed at the output face at the sender. Indeed, losses due to mobility are caused by the absence of available output faces to reach the mobile consumer/producer (respectively for Data/Interest packets). Thus, they occur before WLDR labeling. Packets queued in the output buffer may still suffer from drops due to congestion: while managing congestion losses is out of the scope of this chapter and supposed to be handled by congestion control, we observe that WLDR intervenes only at service time before packet transmission. In this way, packet losses due to congestion do not interfere with WLDR mechanism.

Loss Notification. In case of loss, the AP notifies the consumer with an *Explicit Wireless Loss Notification* (EWLN) message. The EWLN contains the current expected sequence number (4 in Fig. 6.1), and the sequence number of the last received Interest (6 in the figure) to notify the loss of the expected packet, namely 4, and packets in between, namely 5. Once EWLN is sent, the AP updates its expected sequence number to the last received Interest plus 1 (7 in the example).

The usage of EWLN packets is in contrast with most of the MAC layer retransmission algorithms, where ACKs are used to track losses. ACKs are used because they are not subject to the communication patterns, since a node can detect a loss using timeouts. The drawback of ACKs is that a node may keep retransmitting packets even if the wireless channel is down, and this is what happens today in the IEEE 802.11 standard. Instead, with EWLN packets a node receives a loss report only if the channel is good enough to transmit some packets. In this way packets are retransmitted only when the channel conditions are good enough, resulting in better performance. Notice that in case an EWLN message gets lost, the baseline case of retransmission by timer at the receiver will happen to eventually recover the loss.

In-network recovery. WLDR is designed to recover the losses in-network, without sending any signal to the application/transport layer running at the consumer side, but a different recovery strategy can be implemented (e.g. explicit loss notification to consumer, see Sec. 6.4.2). Loss recovery is enabled by maintaining a buffer of Interest/Data packets or, depending on the forwarder implementation, by reference to the content store (CS), for data, or the PIT, for interests with no need for copying data. In Fig. 6.1, when the consumer receives the EWLN packet it retransmits all Interests indicated by the EWLN. Interest packets retransmitted by the consumer are sent with a new sequence number to keep the two nodes synchronized (in the example, the AP is expecting packet 7) and to enable future retransmission in case of channel losses. In Fig. 6.1 the new sequence values would be 7 and 8, respectively, for the two lost Interests 4 and 5. The detailed WLDR algorithm is reported in Alg.3,4.

Algorithm 3: WLDR algorithm (sender side)

```
buffer [] ;                                      // Local buffer with sent packets
bufferSize ;                                                // Local buffer size
next ;                                                   // Next sequence number
Function OnSendPacket (packet)
    packet.setSeqLabel(next);
    buffer[next % bufferSize] = packet;
    next++;
    send(packet);
Function OnEWLN (ewlnPkt)
    expectedPkt = ewlnPkt.getExpectedPkt();
    lastReceivedPkt = ewlnPkt.getLastReceivedPkt();
    if ((next - expectedPkt) <= bufferSize) then
        // lost packets are in the buffer
        while (expectedPkt < lastReceivedPkt) do
            lostPkt = buffer[expectedPkt % bufferSize];
            if (lostPkt is not expired) then
                lostPkt.setSequenceLabel(next);
                buffer[next % bufferSize] = lostPkt;
                send(lostPkt);
                expectedPkt++;
                next++;
```

WLDR enhancements

Adjusting Interest/Data lifetime: To avoid retransmissions for packet with expired PIT entry, we use the lifetime field in the Interests and we add an equivalent field in the Data that contains a copy of the Interest lifetime. When the sender node labels a packet (Interest or Data) for the first time, it stores a timestamp so that it can decide when the packet will be timed-out and will not be retransmitted. In case of retransmission, the sender computes the time elapsed from the first packet transmission and the packet is retransmitted only if such time is less then $\alpha \times lifetime$, where $\alpha \in [0, 1]$. With α close to 1 we have more chances to retransmit packets for which the PIT timer is already expired, while with α close to 0 we may not retransmit valuable packets.

Reinitialization of sequence number: A critical component of WLDR is to keep nodes in sync in case of handovers. There are two possible scenarios: *(i)* the mobile node temporarily disconnects from an AP and reconnects to the same one, or *(ii)* the mobile node migrates to a new AP. In the first case there is just a temporary disconnection, so we keep using the same counting sequence. This has the advantage that, if some packets got lost during the disconnection, the two nodes may recover them. Instead, in the second case, we reset all the WLDR state on both the nodes. In this case, the losses due to mobility are handled by MLDR (see Sec.6.2.2).

Algorithm 4: WLDR algorithm (receiver side)

```
expected ;                                    // Expected sequence number
Function OnReceivePacket (packet)
    pktLabel = packet.getSeqLabel();
    if (pktLabel != expected) then
        ewlnPkt.setExpectedPkt(expected);
        ewlnPkt.setLastReceivedPkt(pktLabel);
        send(ewlnPkt);
    expectedLabel = pktLabel + 1;
```

6.2.2 Mobility Loss Detection and Recovery

NDN/CCN name-based connectionless transport significantly simplifies mobility management. However, mobility events may still lead to losses, as a result of Interest expiration in PIT due to temporary unavailability of forwarding output face on the path between consumer and producer. The goal of MLDR is to handle losses due to either consumer or producer mobility, during the time period required by link-layer reconnection and mobility management protocols to update network forwarding state (in case of producer mobility). In the following, we consider separately consumer and producer mobility and introduce MLDR countermeasures accordingly.

Consumer mobility

Consumer mobility is natively supported in NDN/CCN, since after moving and attaching to a new base station, a consumer can reissue lost Interests to retrieve data available from the closest cache. The main problem when a consumer node changes its point of attachment is that the Interests that are already pending in its PIT will never be satisfied due to the symmetric routing property. Such losses due to mobility might be mistakenly treated as congestion signal, which may further affect flow control through, for instance, a window or rate reduction.

Loss detection. Interest losses are typically detected by means of PIT timer expiration, with the timer being set equal to the Interest lifetime. Besides known timer setting difficulties, waiting for timer expiration implies retransmission delays and negatively affects congestion control behavior, regardless of the specific rate control methods used at the consumer. In MLDR, we base consumer mobility loss detection on local "face up/down" signaling, also distinguishing them from losses of another nature. In practice, this is a task of link-layer technologies and the network layer just receives the signal from them. For MLDR to benefit from such signals, the link-layer detection of "face up/down" should be as

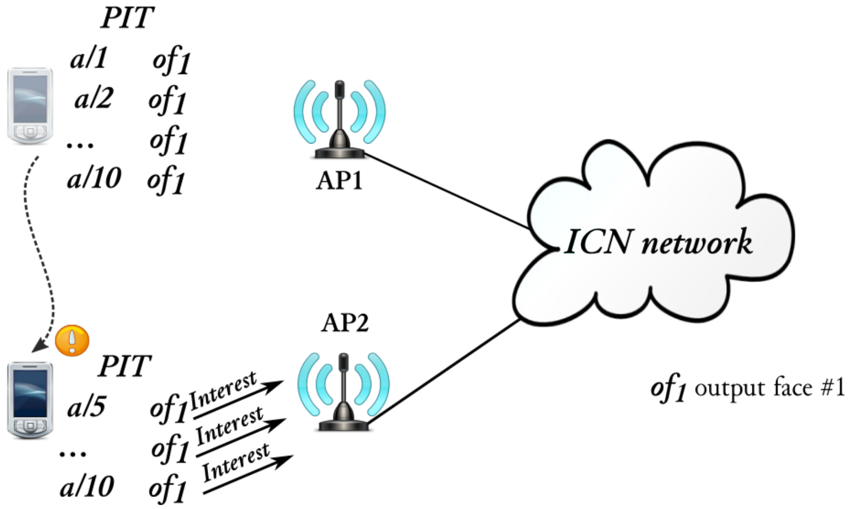


Figure 6.2 – On detection of consumer mobility: for the case where a wireless face first goes down and there is no alternative face available, and later in time the same face goes up again.

agile as possible. Indeed, upon change of AP by the consumer, the corresponding face goes down until the connection with another AP is successfully established. A signal of "face down" triggers in MLDR a PIT lookup in order to find all entries associated with pending interests forwarded through such a face.

Loss notification and recovery. For each of the PIT entries involving the face going down as an output face, a *Notify and Retransmit* procedure is initiated. Its goal is twofold: *(i)* to inform the transport layer of the mobility event in order to prevent unnecessary interest rate reduction upon PIT timer expiration and *(ii)* to trigger fast interest retransmission, on another available face, if any, or later in time on a wireless face going up again when consumer. reconnects to the network. To this aim, MLDR first checks whether alternative output faces are available. If so, it retransmits the interests. Otherwise, if no alternative face is available, MLDR adds a special mobility flag, denoted as M to the interest and sends it back to the application to inform it about the mobility event. On a "face up" signal, MLDR retransmits all pending Interests that have no output face. It is important to remark that PIT entries are not removed by MLDR, rather updated in the implementation as far as concerns the pointers to the output faces. In the presence of multiple output faces corresponding to a PIT entry, only the face that is down is removed.

An example of consumer mobility detection is illustrated in Fig. 6.2. The consumer sends Interests $a/1$ to $a/10$, while connected to $AP1$. The first four Interests are satisfied before its disconnection from $AP1$. When the consumer connects to $AP2$, $a/5$ to $a/10$ Interests are still pending in its PIT, so a retransmission to $AP2$ can be triggered.

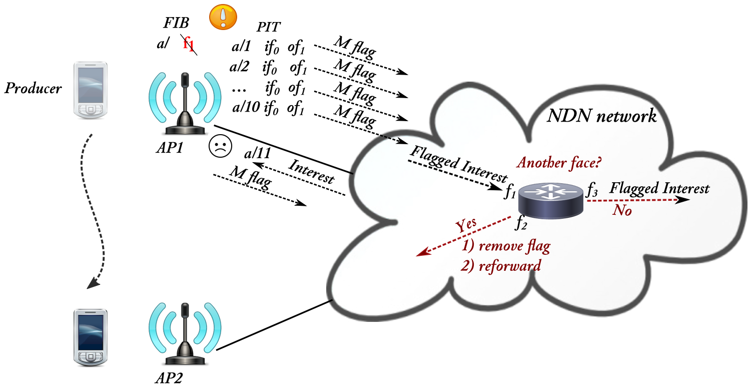


Figure 6.3 – On detection of producer mobility: when a face down signal is received at AP1, loss detected as in the second case. Since no alternative faces are available, interests are all M-flagged and forwarded back.

Producer mobility

We now address producer mobility, namely the case where a change of AP for the producer breaks the path between consumer and producer with consequent loss of Interest and Data packets in both directions. MLDR's objective is to recover such losses during the time period required by the mobility management protocol to update network forwarding state according to new producer location. To do so, MLDR notifies in-path routers about the mobility event, to enable a retransmission of the Interests over alternative paths within PIT entry/application lifetime. Indeed, the proposed solution is a generalization of MLDR detection and notify plus retransmit procedure described above.

Loss detection. A mobility event detection occurs in two cases: first, for an Interest arriving at the old AP where it cannot be forwarded anymore and, second, as in the consumer mobility case, upon "face down" signaling at the same AP.

Loss notification and recovery. In the case where the AP cannot forward an incoming Interest, a flag M is set in the Interest packet and the latter is sent back on the incoming face. A "face down" signal at the AP triggers MLDR detection procedure as previously described and the same procedure as for consumer mobility applies with possible retransmission over alternative available faces. Otherwise, the $M-$flagged Interests will be generated and sent through the corresponding incoming faces (whose pointers are stored in PIT entries) to propagate back to the consumer an explicit mobility notification. Corresponding entries are then removed from the PIT.

An example is illustrated in Figure 6.3. When a face of_1 goes down, the pending Interests $a/1$ to $a/10$ will be flagged and sent back on their incoming face(s). If the AP is still receiving the Interests for this particular content, it will flag them and forward back.

Propagation of $M-$flagged Interests may give rise to retransmission on other available faces, e.g., those dynamically created by mobility management protocols to reconnect the producer at the new location. Upon reception of an $M-$flagged Interest at a given network node, a decision about Interest re-forwarding should be taken. The Interest is re-forwarded in the case where the face from which the flagged Interest arrived is a unique output face for the corresponding PIT entry. If this face is not unique, the flagged Interest is rejected and the corresponding output face is just removed from the PIT entry. If the flagged Interest can be re-forwarded,a FIB lookup is performed to find another face to use according to the node forwarding strategy, which can support multipath or not, independently from our scheme. Note that multipath does not require any change to MLDR. The M flag is then removed from the packet and the Interest is re-forwarded through the selected face.

If no other face is available, the $M-$flagged Interest is forwarded to the list of corresponding incoming faces. The PIT entry is then removed. It is easy to see that if no nodes of the path have alternative faces to re-forward the $M-$flagged Interest, it will finally arrive to the consumer to notify it about the mobility loss. As for wireless channel losses, unnecessary interest rate reduction may be prevented by the explicit notification of mobility loss to consumer. It is important to observe that FIB entries are not altered, in fact, MLDR never removes a face from the corresponding FIB. The algorithm in case of producer mobility is detailed in Alg. 5.

MLDR Enhancements

Adjusting Interest lifetime: One of the operations that each node (consumer included) should perform before re-forwarding an $M-$flagged Interest, is to adjust the lifetime value carried by the re-forwarded Interest to the residual time left until the PIT entry expiration. In addition, as for WLDR in case of retransmissions, the decision about re-forwarding may be based on this value. For example, only if residual time is large enough the Interest is re-forwarded, otherwise, it will be sent back to downstream nodes in the case where there are no alternative faces. Note that we do not modify the PIT timers.

Preventing retransmission loops: To avoid loops due to re-forwarding an $M-$flagged Interest over the same output face, a list of faces used for retransmission is added to the subset of PIT entries affected by a mobility loss notification. Every time an $M-$flagged Interest is forwarded, the corresponding output face is added to the list. Only if such a face has not yet been used for retransmission of an Interest with the same name during PIT entry lifetime, the Interest is effectively re-forwarded. Otherwise it is further propagated with the M flag in the direction of the consumer.

RTT (Round Trip Time) reduction: Such enhancement to baseline MLDR only applies to the class of congestion controllers at the consumer side leveraging RTT monitoring. Indeed, in-network retransmissions may introduce an additional delay affecting

Algorithm 5: MLDR algorithm (Producer mobility)

```
Function OnProducerWirelessFaceDown (face)
    while (PIT.hasNextEntry()) do
        reforwardInterests(face,PIT.getNextEntry());
Function OnMobilityFlag (interest,face)
    interest.unsetMobilityFlag();
    reforwardInterests(face,PIT.match(interest));
Function reforwardInterests (face,entry)
    if (face $\in$ entry.outFaces() &
                |entry.outFaces()| = 1) then
        interest = entry.getInterest();
        nextHopFaces = FIB.match(interest);
        forall (outFace $\in$ nextHopFaces) do
            if (outFace $\notin$ entry.InFaces() &
            outFace $\notin$ entry.usedForRetransmission()) then
                residualTime = entry.expiration() - now;
                interest.setLifetime(residualTime);
                outFace.send(interest);
                entry.addUsedForRetransmission(outFace);
                return;
        interest.setMobilityFlag();
        forall (inFace $\in$ entry.inFaces()) do
            inFace.send(interest);
        PIT.remove(entry);
```

RTT monitoring at the consumer which may be misinterpreted as due to congestion. To avoid this, we aim at correcting RTT estimation at the consumer by removing the RTT component due to retransmission. This scheme updates the PIT entry corresponding to an Interest to be reforwarded with a retransmission timestamp. Similar techniques have been proposed in [102]. Thus, we need two PIT timestamps: the original sending time and the retransmission time. Note that in case of multiple retransmissions of an M-flagged Interest, we record only the latest retransmission timestamp. In this way, the difference between the two timestamps indicates the overall retransmission time for the Interest. Upon reception of Data matching the retransmitted Interest, the difference between these two timestamps is computed and stored in the Data packet. Thus, this difference (equal to the time spent on loss recovery) can be removed in RTT computation at the consumer side. The RTT reduction can be important for modern delay-based congestion control protocols. This RTT reduction scheme may also be used with WLDR to explicitly notify the application about the additional delay introduced by the retransmission process.

General comments: Before the performance evaluation in Sec. 6.4, we observe that immediate local retransmission at consumer side may already improve Interest satisfaction time w.r.t. timer-based retransmissions, but the latency reduction gains are even more important in the presence of in-network retransmissions. In both cases, MLDR enables retransmission at sub-RTT scale, which is unlike any approach based on timer expiration at consumer side or explicit notification and retransmission at the consumer. It is important to observe that even if other nodes than consumer/producer are mobile, MLDR can still deal with that, under more complex mobility management. Finally, from the security point of view, we note that MLDR does not modify any ICN data plane information that could introduce opportunities for attacks.

6.3 Implementation

In this section we describe the additional state required in each node by WLDR and MLDR as well as the additional header fields that we use in Interest/Data packets.

WLDR. The ICN forwarder of a node running WLDR (a WLDR node hereinafter) needs to locally store two values per face: the next sequence number, `next_seqno` and the expected sequence number, `expected_seqno`. Both of them are integers and in 4 bytes. Such values are required for packet labeling at the sender and loss detection at the receiver. In addition, a WLDR node keeps track of the sequence of sent packets and temporarily stores them (or stores a pointer to their copy in PIT/CS) in a circular buffer, one per output face: each packet is hence stored at position `seqno % buffer_size`. The amount of additional state required by WLDR depends on the size of this buffer. A too small buffer may reduce chances to recover losses, because of packets being overwritten too frequently. In order to dimension such a buffer, we consider a 'bandwidth delay product' rule where delay stands for PIT timer (logically the content lifetime which however may vary according to the considered application). In the current implementation this results in a buffer of 8192 packets considering 1sec PIT timer and a Wi-Fi link at 100Mbps, which is close to the number of Data packets (of 1500 bytes) that can be sent in a second on this link. This guarantees that we almost never overwrite any useful packet, for a cost in terms of memory of around 11MB. However, if we record only the pointers to the PIT/CS entries where these packets are already stored, the considered buffer size can be reduced to 64kB.

The amount of states required by WLDR depends also on the number of local faces. As a mobile host, the number of lcoal faces is limited to its wireless interfaces (e.g., Wifi and LTE). So as a mobile host, the amount of states is about 64*2=128KB. For an AP (access point), the number of faces depends on the type of wireless access (e.g., Wifi or LTE interfaces). For a Wifi AP, the number of connected stations is in the order of tens and hence the total states are about 640KB. As an LTE AP (i.e., EnodeB), the number of connected mobile devices is in the order of thousands, which means 64MB of states.

In terms of computational complexity, WLDR only adds constant and negligible delay to each packet forwarding operation.

MLDR. MLDR does not require substantial modifications to the existing data structures or packet format. However, additional information has to be stored by each router to prevent retransmission loops of forwarded Interests and perform the RTT reduction (see Sec.6.2.2). More precisely, a list of faces used for retransmission is added to PIT entries affected by a mobility loss notification. To perform the RTT reduction signaling, two additional timestamps need to be stored in the PIT entries: the time of the original sending of an Interest and the moment of its retransmission (if any).

In terms of computational complexity, the basic implementation of the *OnProducerWirelessFaceDown* function (see Alg. 5) requires a linear scan of the PIT on the AP node in case of producer mobility. In order to reduce such complexity, we store the list of pointers to associated PIT entries at each output face. With this data structure we can obtain the required PIT entries in constant time, with no iteration over the entire PIT. The complexity involved to lookup an $M-$flagged interest is no more than that of a standard Interest lookup.

Packet format. WLDR and MLDR introduce four new fields in the packets: the *sequence number*, the *Data lifetime*, the *mobility flag*, and the *RTT reduction.* The sequence number is an integer value that is introduced both in Interest and Data. The Data lifetime is the equivalent of the Interest lifetime, and is specific to Data (Interests have this field by default). The mobility flag, that requires a single bit, and the RTT reduction, which is an other integer value, are field required only on the Interests.

EWLN packet: WLDR also introduces a new signaling message, denoted as EWLN (Explicit Wireless Loss Notification) which carries: (i) a flag specifying that the packet is an EWLN, (ii) the expected sequence number at the receiver when the loss is detected and (iii) the sequence number of the last received packet. It is worth observing that such signaling message has a one hop validity and it is discarded by the receiver (e.g. the AP or the station) after having triggered either a retransmission of the missing packet(s) or the creation of explicit notification message(s) carrying the name of the missing packet(s) for further propagation.

6.4 Evaluation

As shown in [69], different packet loss models can affect the effectiveness of the solutions to improve congestion control in wireless environment. Therefore, it is important to evaluate our mechanisms under as realistic wireless environment as possible. To this end, we setup a realistic wireless simulation environment in ns3 2.24/ndnSIM 2.1 using IEEE 802.11n access, as further detailed below, to assess WLDR/MLDR's performance. We assume that the congestion control is built into the transport protocol. Thus, with no loss of generality, we implemented a receiver-driven window-based congestion control scheme based on RAAQM[103] at the consumers and used it in our evaluation.

6.4.1 ICN over IEEE 802.11

We assume all nodes are connected to the same broadcast medium shared in infrastructure mode, namely IEEE 802.11n on 5GHz frequencies, with a single base channel of 40MHz with short guard intervals (SGI), using a single antenna at either the AP and the wireless nodes (denoted as stations).

Channel characteristics and contention: The PHY rate adaptation is minstrel [87] for High Throughput (HT) rates, i.e. MCS (Modulation and Coding Scheme) from 0 to 7 (corresponding to Data rates from 15Mbps to 150Mbps). 802.11 frame aggregation is also enabled with a maximum A-MSDU (Aggregated Mac Service Data Unit) size of 7935 Bytes and A-MPDU (Aggregated Mac Protocol Data Unit) maximum size of 64kB with block ack which enable high application throughputs. When multiple STAs have a face established with the same AP, multiple access is managed by 802.11 EDCF which implies transmission latency and bandwidth sharing among active stations. A face between a STA and the AP is characterized by a time varying capacity that depends on a number of factors like radio conditions, PHY rate selection, medium sharing. In this work, we assume a small cellular Wi-Fi deployment managed by a single entity that can engineer and manage radio planning and 802.11 tuning.

Coverage and mobility: In our simulations each AP operates with a maximum power of 40mW (16dBm), which enables a maximum radio range of 120 meters in outdoor. STAs are assumed to move in a fully covered geographic area performing handover from one cell to another using the Hysteresis handoff algorithm described in [104] to perform handovers among the Wi-Fi cells. The hysteresis handoff has been shown to give the best performance in terms of handover latency. See [105] for more details about 802.11n parametrization.

6.4.2 WLDR Evaluation

We first evaluate WLDR in the scenario illustrated in Fig. 6.4, where one consumer and one producer are connected by means of a 802.11n to a wired network represented by $AP1$, $AP2$ and one intermediate router. During the simulations, these two nodes move back and forth from the two APs as indicated by the arrows. We use mobile nodes in order to test our algorithm with different signal conditions that are variable according to the distance between the STA and the related AP (from 0 to 80 m). We let the speed vary between 3km/h to 50km/h. The STAs remain connected to the same AP (no handovers). The propagation delay of the wired links is set to 1ms, while the link capacity changes according to the simulation. The propagation delay of the wireless links depends on the distance between STA and AP. The workload consists in 10 parallel flows of 50,000 packets ('flow' here stands for retrieval of a content item). Intermediate caching is disabled to allow the observation of wireless losses at both ends.

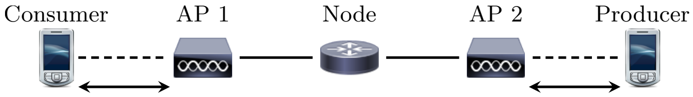
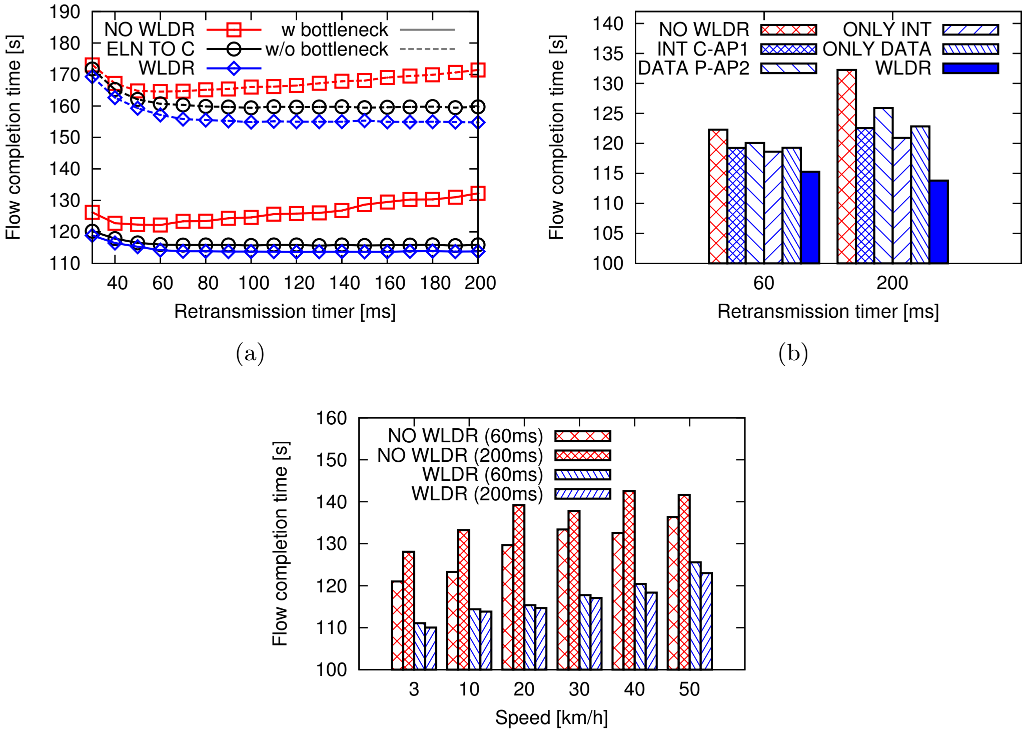


Figure 6.4 – WLDR test topology.

Fig 6.5(a) shows the average flow duration time for different values of Interest retransmission timer (set equal to PIT timer), when the mobile nodes move at 10km/h. In the figures we compare three alternatives: *NO WLDR* indicating the simulations without WLDR, rather with consumer timer-based retransmissions, *WLDR* and finally, *ELN TO C* indicating the solution leveraging Explicit Loss Notification (ELN) messages to the consumer every time a wireless channel loss is detected. In the latter case, we use WLDR detection, but instead to recover the losses in the network, we notify the consumer who immediately retransmits the Interest, without waiting for the timeout nor decreasing the congestion window.

In the absence of WLDR, one can observe a significant dependency of flow completion time on retransmission/PIT timer. Indeed, if the timer value is too large, waiting for a timeout to detect a loss is too costly. On the contrary, if the timer is too small (w.r.t. the expected average round trip time, which is of 50ms), unnecessary timer expirations cause Data discard even in absence of losses.

If WLDR in-network detection considerably improves flow completion time, its in-network recovery also enhances overall performance when compared against *ELN TO C* solution. In fact, the recovery time for *ELN TO C* depends on end-to-end network latency, while WLDR recovery time only on wireless hop latency. Hence, if we create a bottleneck in the wired part of the network (i.e., the wired link between AP1 and node, and the wired link between node and AP2 in Figure 6.4) by modifying wired link capacities from 300Mbps to 60Mbps, the inefficiency gap of *ELN TO C* over WLDR will increase. This has been demonstrated in Figure 6.5(a), where we compare 3 schemes NO WLDR, *ELN TO C* and WLDR under the cases of with or without wired link being the bottleneck. As we can see that in case of wired link being the bottleneck, the gap between WLDR and *ELN TO C* (the 2 lines at bottom) is small. Conversely, in case of wireless link being the bottleneck, we see that the gap between WLDR and *ELN TO C* (the 2nd and 3rd line on the top) becomes more evident.



(a)

(b)

(c)

Figure 6.5 – (a) Flow completion time with and without bottleneck; (b) Flow completion time with WLDR partially activated; (c) Flow completion time for different speeds.

We now break down WLDR gains into its components, namely detection and recovery of Interest rather then Data packets and WLDR at consumer side (between consumer and $AP1$) rather than at producer side (between producer and $AP2$). To this purpose, we enable WLDR only partially in the simulation in order to quantify gains due to each component. Results are reported in Fig. 6.5(b). The speed of the moving nodes is set to 10km/h and we use two values for Interest retransmission/PIT timer: 60 ms (best observed value without WLDR in the previous simulation) and 200ms. The bottleneck is in the wireless part of the network. We show the flow duration associated with 6 cases: (i) WLDR is not active (*NO WLDR*), (ii) Interest recovery between consumer and AP1 only (*INT C-AP1*), (iii) Data recovery between the producer and AP2 only (*DATA P-AP2*), (iv) Interest recovery everywhere (*ONLY INT*), (v) Data recovery everywhere (*ONLY DATA*), (vi) WLDR is fully activated (*WLDR*).

As expected, the timer value has no impact of WLDR, while it visibly affects the performance for *NO WLDR*. Comparing *ONLY INT* and *ONLY DATA*, one can observe that recovering Interest is more advantageous than recovering Data, especially when they are recovered at consumer side. Intuitively, this allows detection and recovery of losses on the first hop, thus significantly reducing recovery time w.r.t. timer-based or ELN-based consumer retransmission.

Data packet recovery is also important, as they are bigger than Interests in size, hence more prone to losses. The benefits for in-network Data recovery are clearly more important when performing it at producer side on the first hop for the Data packet. It is also worth noticing that the difference between the *DATA P-AP2* and *ONLY DATA* is slightly higher in the case of the timer set to 200ms which allow for more retransmissions of Data before timer expiration. We can conclude that WLDR is insensitive to retransmission/PIT timer value provided that the timer is higher than the average round trip time, so accomodating in-network (possibly more than one tentative) retransmissions.

Finally, Figure 6.5(c) shows the flow completion time versus mobile nodes speed for 60ms and 200ms timer values, with the bottleneck still in the wireless network. WLDR always outperforms the case with consumer retransmissions (from 7.22% up to 12% for 60ms timer, between 13% and 18% for 200ms timer).

6.4.3 MLDR Evaluation

In this section we quantify the effectiveness of MLDR scheme by analyzing consumer and producer mobility separately (we consider them jointly in Sec.6.4.4) The topology employed is reported in Fig. 6.6 and consists of the root node acting as consumer or producer in case of producer or consumer mobility respectively, of 6 network routers and 6 IEEE 802.11n Access Points(AP_1 to AP_6) at 50m of distance each other. All wired links have a constant propagation delay of 1ms. In the simulations the STA moves linearly across the APs, as described by the dashed arrow in Fig. 6.6. For the producer mobility, the examples of the directions for in-network reforwarding are illustrated by the dashed blue and green arrows.

To characterize MLDR behavior under a generic mobility management protocol, we implemented an ideal global routing scheme that immediately updates the FIBs of each node as soon as producer mobility is detected (i.e. producer is associated to a new AP). Under real mobility management protocols, the time for updating network forwarding state would be longer, hence the higher the gains due to MLDR w.r.t. the case under study.

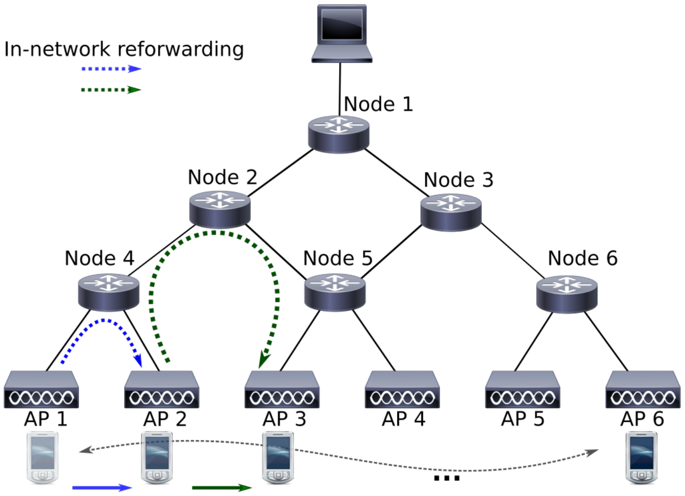
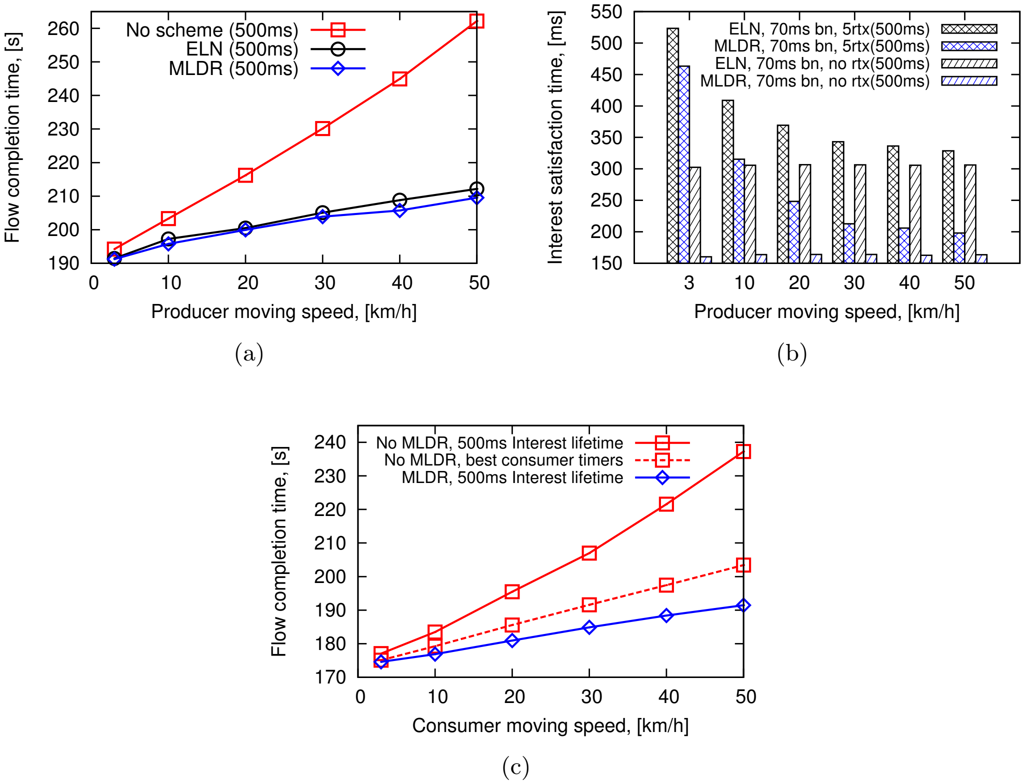


Figure 6.6 – MLDR test topology.

We start from **Producer Mobility**: the producer moves between the APs at different speeds (from 3km/h to 50km/h). We set an Interest lifetime of 500ms, which is an order of magnitude bigger than the average round trip time, here essentially determined by the wireless hop. In all scenarios, the consumer requests 300k packets.

We compare three approaches: (i) the baseline with loss detection and recovery performed at consumer side based on timer expiration (ii) the case with in-network loss detection, Explicit Loss Notification (ELN) to the consumer (iii) MLDR, where either detection and recovery are performed in-network. The ELN scheme exploits $\boldsymbol{M}-$flagged Interests of MLDR, in this case sent directly to the consumer with no interception by in-network routers.

Fig. 6.7(a) reports flow completion time (or download time) as a function of producer moving speed. MLDR gains in terms of loss detection are striking and higher the speed, higher this gain. Indeed, the higher the speed, the higher the number of performed handovers corresponding to the number of times MLDR is in action. Overall, in this scenario, the number of re-forwarded Interests grows with the speed value to a maximum value of



(a)

(b)

(c)

Figure 6.7 – (a) Flow completion time as a function of producer moving speed; (b) Interest satisfaction time as a function of producer moving speed; (c) Flow completion time as a function of consumer moving speed

0.25% of total traffic at 50km/h.

If the benefits of in-network loss detection are significant, the additional gain of MLDR over ELN due to in-network retransmission is much smaller. This can be easily explained as the additional latency introduced to notify the consumer and let him retransmit the Interest packets is negligible w.r.t. the overall round trip time, mainly affected by wireless hop delay.

To better understand where the difference between these two schemes plays a role, we increase the propagation delay of the links between *Node* 1, *Node* 2, *Node* 3 to 70ms and compute for ELN and MLDR solutions the average "*Interest satisfaction time (IST)*", i.e., the interval of time between the first transmission of an Interest packet and the reception of the corresponding Data packet.

IST is measured at the consumer and takes into account all performed retransmissions. The average values of IST for the retransmitted packets are presented in Figure 6.7(b).

We show the results for the cases with (indicated with *5rtx*) and without (labeled *no rtx*) additional retransmissions by timer performed by the consumer. In case of retransmissions the consumer can issue the same Interest up to 5 times in case of timer expiration, as in all the other simulations. In the no retransmission setting, the consumer retransmits an Interest only on reception of a loss notification.

In the absence of further retransmissions, we observe that MLDR and ELN show a constant retransmission time as a function of moving speed, with MLDR considerably outperforming ELN by means of much smaller IST (almost half of ELN's IST). The performance gap remains in the case where additional consumer retransmissions are allowed, also increasing as moving speed grows. MLDR's better performance here is a consequence of the increasing amount of re-forwarded Interests that have the effect of reducing the number of timeouts/required retransmission for MLDR.

From the presented results we can conclude that, in terms of IST, MLDR always outperforms ELN by virtue of the additional delay that the $M-$flagged Interests experience to reach the consumer and to be retrnasmitted there, rather then being intercepted and immediately re-forwarded by the routers of the path, as is the case for MLDR.

Consumer mobility loss recovery: We now move to the consumer mobility case. We consider the same network topology in Fig. 6.6, with the producer placed at the root. The consumer moves between the APs at different speeds (from 3kmh to 50km/h).

The results of this scenario are presented in Fig. 6.7(c). Here, retransmissions can only be performed at the consumer, hence we compare the performance of MLDR against consumer-driven retransmission using the best timer values, i.e., the values associated with each moving speed that give us the best performance as obtained by a set of simulations with different timer values, as we did for WLDR in Fig. 6.5(a). Here we tested timers from 50ms to 500ms.

In Fig. 6.7(c), we observe again the significant improvement in terms of flow completion time when MLDR is activated. This is both due to earlier detection and recovery. Indeed, MLDR performs better than consumer retransmissions under either 500ms timer value, either the best consumer timers, by virtue of its quick reaction to mobility events.

6.4.4 Joint WLDR-MLDR Evaluation

In this section we analyze the performance of the two proposed algorithms combined, in a more realistic scenario. We use the topology in Fig. 6.8, where APs are positioned in a grid of 6 by 6 nodes at a distance of 80m each other. Each edge router in the lower layer of the fat tree is connected to 3 APs. We run simulations with 10 mobile nodes (5 acting

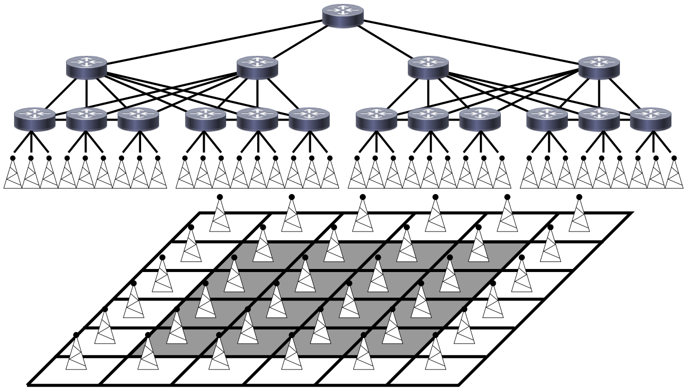


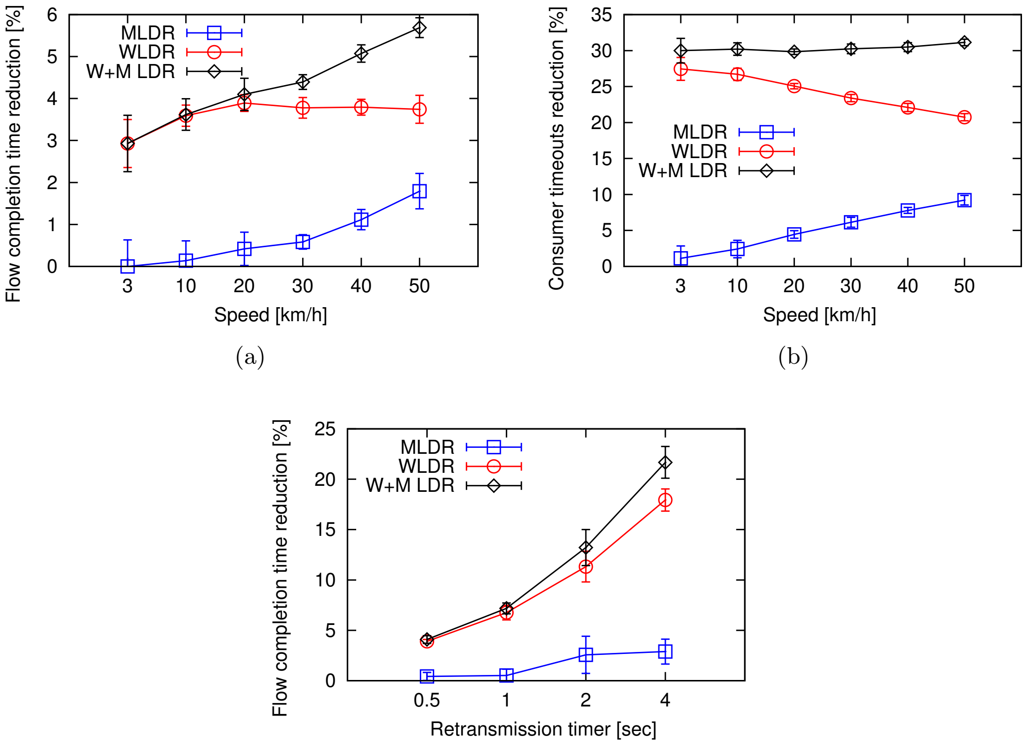
Figure 6.8 – Edge network topology.

as consumers, 5 as producers). The mobility is simulated using a random waypoint model. Micro-mobility is assumed as in current radio mobile networks as LTE. The mobile nodes move in the area indicated by the gray squares in Fig. 6.8. We put extra APs outside the moving area to guarantee homogeneous radio coverage: each STA can sense 9 APs from each point of the simulation area. A consumer retrieves a file composed of 100k chunks from a single producer. We run each simulation 200 times, with the nodes starting at different positions.

Fig. 6.9(a) shows the average and the standard deviation of the flow completion time reduction (in percentage) that we obtain running our proposals with respect to the results obtained recovering losses at the consumer on timer expiration. In these simulations we set the retransmission timer (or Interest lifetime), and so the PIT timer, equal to 500ms. In the figure *MLDR* and *WLDR* indicate the gain that we achieve using only one of the two proposed algorithms, while *W+M LDR* denotes the gain when both WLDR and MLDR are used.

It is easy to see that MLDR benefits increase when the speed of the mobile nodes is higher. This is due to the larger number of handovers at high speed during the simulations, hence of opportunities for MLDR to quickly detect and recover packets that would have been lost otherwise. For instance, when the mobile nodes move at 3km/h on average 17 handovers per flow occur (considering the sum of the handovers performed by the consumer and the producer), while at 50km/h the average number of handovers is more than 200. WLDR, instead, is more effective at low speeds, when the majority of the losses are due to the wireless channel. Finally, we can see that by combining the two algorithms the resulting gain is sum of the gains brought by each one separately. In this setting we achieve a maximum gain of almost 6% when the mobile nodes move at 50km/h.

Fig. 6.9(b) reports the average and the standard deviation of the reduction of the timeouts registered at the consumer when we enable our mechanisms. The simulation setting



(a)

(b)

(c)

Figure 6.9 – (a) Flow duration for different speeds, (b) Timeouts reduction at the consumer, (c) Flow duration for different retransmission timeouts.

is the same as the one in Fig. 6.9(a). This figure confirms our conclusions: WLDR is more effective a lower speeds, when most of the losses are due to the wireless channel, while MLDR becomes more and more effective when there is an increase in the number of mobility events. Enabling both WLDR and MLDR we are able to reduce the number of timeouts at the receiver by more than 30%.

In Fig. 6.9(c) we show again the flow completion time reduction, but as a function of the retransmission timer. In this set of simulations we set the node speed to 20km/h. Increasing the retransmission timer from 500ms to 4sec the flow completion time can be reduced by more than 20% combining WLDR and MLDR. This is due to the fact that using in-network retransmissions we remove the dependency on network/application timers that affect the network performance and are really difficult to tune correctly.

6.5 Conclusions

ICN with hop-by-hop forwarding transport model offers an opportunity to rethink congestion control over wireless mobile networks beyond the limitations of traditional connection-based approaches, by leveraging in-network control capabilities. Quite some attention has been devoted to congestion control design in the ICN community, but very little to the case of wireless mobile environments, where the distinction of the nature of loss events and the capability to achieve prompt recovery are key factors for effective rate and congestion control.

In this chapter, we analyzed the potential for improvement of in-network loss detection and recovery and proposed two solutions, WLDR and MLDR, respectively tackling wireless channel losses and losses due to mobility events. WLDR-MLDR follow by the same design principles: they consist of a link-layer agnostic, purely distributed approach decoupling in space and in time loss detection and recovery operations by exploiting Explicit Loss Notification messages. Fast recovery at sub-round trip time scale is achieved in the network once the information about loss detection has reached the first potential retransmission point. The performance evaluation carried out by means of simulations shows significant benefits in terms of reduction of flow completion time or per-packet request satisfaction time over consumer-based solutions. The other advantage over state of the art solutions is that WLDR/MLDR remove the dependency on network/application timers.

Note that the services provided by WLDR/MLDR may not be necessary for all types of applications. For instance, for delay-sensitive applications (e.g., teleconferencing) dropping lost packets can be preferable to recovering them in the network due to the delay incurred by the recovery operations. Therefore, it is useful to have the flexibility of enabling/disabling WLDR/MLDR based on the needs from applications. To this aim, we plan as a future work to migrate the design and implementation of WLDR/MLDR from network layer to ICN's strategy layer to have the needed flexibility. It will be interesting to see how well WLDR/MLDR can perform under different traffic mixes (i.e., mix between throughput-sensitive traffic requiring WLDR/MLDR and delay-sensitive traffic not requiring WLDR/MLDR). Furthermore, we plan to carry out larger scale experimentation in realistic indoor/outdoor Wi-Fi environments and extend the analysis to other wireless access technologies in 5G context.

Chapter 7

Conclusion

With the proliferation of mobile computing devices and advances in wireless access technology, mobility has become a required component for Internet communications. However, despite the numerous efforts devoted to enabling mobility within IP networks in the past decades, the resulting set of mechanisms are mostly relying on anchors, and hence are inefficient, complex and access-dependent (e.g., mobility management in 3G/4G).

In this context, ICN has been proposed as a promising future internet architecture that offers a paradigm shift from host-centric to data-centric architecture and brings a number of promising benefits w.r.t the current IP-based Internet. While one of the recognized strengths of ICN is superior mobility support, several architectural challenges remain to be solved to make ICN a success on Internet mobility support. This thesis proposed several steps towards addressing these challenges. In particular, the thesis has focused on 3 challenges from different ICN architectural aspects.

First, we addressed the challenge of producer mobility. To that end, *MAP-Me* , an anchor-less protocol managing intra-AS producer mobility, even in the presence of latency-sensitive traffic, has been designed, implemented and evaluated. *MAP-Me* is simple and only leverages the ICN forwarding plane and reactive notifications sent to the network to manage mobility. We used ndnSIM 2.1 with 802.11n Wifi access networks for evaluation. Extensive simulations across a variety of topologies, mobility patterns, and radio models demonstrated that *MAP-Me* improves user performance in terms of handoff latency, packet loss, and path stretch while retaining low network overheads w.r.t existing proposals, including anchor-based and trace-based approaches, a global routing approach, as well as a reference implementation for our *MAP-Me*approach. The reported results confirmed our initial objectives and showed that *MAP-Me* optimally offloads from the infrastructure the communications that are local. We open-sourced our simulation framework, as it will be potentially useful for future research in developing new mobility solutions in ICN.

Further, we complemented *MAP-Me* , as well as existing proposals regarding producer mobility management, by investigating security implications of producer mobility. We presented a protocol for prefix attestation based on hash-chaining to protect against *prefix hijacking attacks* that may occur during mobility updates. The protocol secures *MAP-Me* as well as trace-based mobility management solutions that can be promising in the context of producer mobility management of ICN. The protocol is lightweight and fully distributed. We also proposed countermeasures to replay-based prefix hijacking attack, which is not taken into account by existing signature-based proposals in the literature. Our protocol can run unchanged on different hardware deployed at operational network access (e.g., LTE or WiFi). Initial evaluation results confirmed that our protocol introduces minimal computational and storage overhead.

Finally, at the ICN transport layer we investigated the issue with congestion control in a mobile environment. Quite some attention has been devoted to congestion control design in the ICN community, but very little to the case of wireless mobile environments, where non-congestion related loss (i.e, due to wireless/mobility) can have adverse effects on receiver-driven congestion control of ICN. To address such problems, we analyze the potential of ICN's in-network processing capability and proposed two solutions, WLDR and MLDR, tackling wireless channel losses and mobility losses, respectively. WLDR/MLDR follow the same design principles: they consist of a link-layer agnostic, purely distributed approach, decoupling in space and in time loss detection and recovery operations by exploiting Explicit Loss Notification messages. Fast recovery at sub-round trip time scale is achieved in the network once the information about loss detection has reached the first potential retransmission point. The performance evaluation carried out by means of ndnSIM based simulations using 802.11n wifi and a fat-tree topology showed significant benefits in terms of reduction of flow completion time ($> 20\%$ gain) and per-packet request satisfaction time over consumer-based solutions. The other advantage of WLDR/MLDR over existing methods in literature to cope with mobile ICN network is the removal of the dependency on network/application timers.

Chapter 8

Future Work

The work done in this thesis has explored several important architectural challenges in ICN that need to be addressed before it can fully support mobility. However, many aspects still need further investigations and some other architectural challenges have not been touched upon yet due to lack of time during the thesis. We discuss several of the most relevant ones below:

In the work of *MAP-Me*, while the proposed solution can efficiently address producer mobility at micro-mobility level (i.e., intra-AS mobility), a solution to manage mobility at macro-mobility level (i.e., inter-AS roaming) is still lacking. Since macro-mobility happens at a much lower frequency, the handoff latency or packet loss performance can be of secondary consideration. Therefore, we envision that at macro-mobility level, a resolution-based mobility scheme (e.g., DNS-like) can be used to complement *MAP-Me*'s design. In other words, some mapping between producer's original prefix and the new one obtained after inter-AS roaming must be recorded and queried at the resolution server to enable macro-mobility of the producer. Still, this name prefix translation leads to caching performance degradation. The design of a resolution-based scheme that can avoid such caching degradation is a non-trivial technical problem for future work.

Another limitation of the current *MAP-Me* design is the non-support of multihoming for a mobile producer. This is because one Interest Update of *MAP-Me* will replace all paths to multi-homed interfaces with one unique path pointing to the location where the IU message is originated. To solve this problem, a potential direction is to keep the paths to muliti-homed interfaces separate and only allow Interest Update to modify the path belonging to the same interface. For instance, if producer is connected through both Wifi and LTE, we can associate each output face in the FIB with a specific interface id (i.e., Wifi or LTE). And IU/IN message carries also the right interface id when sent through an interface (i.e, wifi or LTE). When the router receives IU/IN, it only updates and forwards the IU through the face with the matching interface id. In this way, we can keep multi-homing capability in *MAP-Me*. However, this requires additional states in the FIB and knowing interfaces used by a producer a priori. Further investigation is needed to enable multi-homing in *MAP-Me* .

Also, we plan as a future work to test *MAP-Me* on real machines or using emulated testbeds (e.g., vICN [106]). This is important to discover design issues we may not be able to see with ns3-based simulations, e.g., packet losses due to non-instant FIB update operations induced by *MAP-Me*.

The *prefix attestation protocol* that we propose to secure trace-based producer mobility protocols and *MAP-Me* also needs further refinement and investigation. In particular, regarding the security context, we still need to define a concrete mechanism that exploits routing protocol to periodically update security context with a fine-grained time interval to avoid the overwhelming complexity incurred by lots of hash computations as well as high communication overhead potentially incurred by the periodic update operations. Moreover, we leave as future work the design and the implementation of a mechanism for preventing Denial of Service attacks as explained in chapter 5. Apart from these, we also need to investigate other types of attacks other than prefix hijacking that are feasible for producer mobility and design countermeasures accordingly.

Last but not least, in our attempt to define the WLDR/MLDR mechanism to improve congestion control in mobile environments, currently we still lack the flexibility to enable/disable WLDR/MLDR based on application needs. Note that the services of WLDR/MLDR may not be necessary for all types of applications. For instance, for delay-sensitive applications (e.g., teleconferencing), dropping lost packets can be preferable to recovering them in the network due to the delay incurred by the recovery operations. Therefore, it is useful to have the flexibility of enabling/disabling WLDR/MLDR based on applications needs. To this aim, we plan as a future work to migrate the design and implementation of WLDR/MLDR from network layer to ICN's strategy layer to have the needed flexibility. It will be interesting to investigate how well WLDR/MLDR can perform under different traffic mixes between throughput-sensitive traffic and delay-sensitive traffic. Furthermore, we plan to carry out a larger scale experimentation in realistic indoor/outdoor Wi-Fi environments and extend the analysis to other wireless access technologies in 5G context.

Bibliography

[1] Cisco Visual Networking Index: Forecast and Methodology, 2016–2021. `https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf`, Jun 2017. Online; accessed 13 November 2017.

[2] Ekram Hossain and Monowar Hasan. 5g cellular: key enabling technologies and research challenges. *IEEE Instrumentation & Measurement Magazine*, 18(3):11–21, 2015.

[3] Naga Bhushan, Junyi Li, Durga Malladi, Rob Gilmore, Dean Brenner, Aleksandar Damnjanovic, Ravi Sukhavasi, Chirag Patel, and Stefan Geirhofer. Network densification: the dominant theme for wireless evolution into 5g. *IEEE Communications Magazine*, 52(2):82–89, 2014.

[4] Patrick Kwadwo Agyapong, Mikio Iwamura, Dirk Staehle, Wolfgang Kiess, and Anass Benjebbour. Design considerations for a 5g network architecture. *IEEE Communications Magazine*, 52(11):65–75, 2014.

[5] Van Jacobson, Diana K Smetters, James D Thornton, Michael F Plass, Nicholas H Briggs, and Rebecca L Braynard. Networking named content. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 1–12. ACM, 2009.

[6] D. Le, X. Fu, and D. Hogrefe. A review of mobility support paradigms for the internet. *IEEE Communications Surveys Tutorials*, 8(1):38–51, First 2006.

[7] Jürgen Hofmann, Vlora Rexhepi-van der Pol, Guillaume Sébire, and Sergio Parolari. 3gpp release 8. *GSM/EDGE: Evolution and Performance*, pages 63–99, 2011.

[8] Yuanjie Li, Zengwen Yuan, and Chunyi Peng. A control-plane perspective on reducing data access latency in lte networks. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 56–69. ACM, 2017.

[9] Zafar Ayyub Qazi, Melvin Walls, Aurojit Panda, Vyas Sekar, Sylvia Ratnasamy, and Scott Shenker. A high performance packet core for next generation cellular networks. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 348–361. ACM, 2017.

[10] Damon Wischik, Costin Raiciu, Adam Greenhalgh, and Mark Handley. Design, implementation and evaluation of congestion control for multipath tcp. In *NSDI*, volume 11, pages 8–8, 2011.

[11] Gareth Tyson, Nishanth Sastry, Ivica Rimac, Ruben Cuevas, and Andreas Mauthe. A survey of mobility in information-centric networks: Challenges and research directions. In *Proc. NoM*, pages 1–6, New-York (USA), 2012.

[12] Yu Zhang, Alexander Afanasyev, Jeff Burke, and Lixia Zhang. A survey of mobility support in named data networking. In *Proc. of IEEE INFOCOM NOM*, 2016.

[13] Yu Zhang, Hongli Zhang, and Lixia Zhang. Kite: A mobility support scheme for ndn. In *Proc. of ACM ICN Poster*, 2014.

[14] Hitesh Ballani, Paul Francis, and Xinyang Zhang. A study of prefix hijacking and interception in the internet. In *ACM SIGCOMM Computer Communication Review*, volume 37, pages 265–276. ACM, 2007.

[15] Mohammad Al-Shurman, Seong-Moo Yoo, and Seungjin Park. Black hole attack in mobile ad hoc networks. In *Proceedings of the 42nd annual Southeast regional conference*, pages 96–97. ACM, 2004.

[16] Mauro Conti, Paolo Gasti, and Marco Teoli. A lightweight mechanism for detection of cache pollution attacks in named data networking. *Computer Networks*, 57(16):3178–3191, 2013.

[17] Cesar Ghali, Gene Tsudik, and Ersin Uzun. Needle in a haystack: Mitigating content poisoning in named-data networking. In *Proceedings of NDSS Workshop on Security of Emerging Networking Technologies (SENT)*, 2014.

[18] Moreno Ambrosin, Alberto Compagno, Mauro Conti, Cesar Ghali, and Gene Tsudik. Security and privacy analysis of nsf future internet architectures. *arXiv preprint arXiv:1610.00355*, 2016.

[19] Yu Zhang, Alexander Afanasyev, Jeff Burke, and Lixia Zhang. A survey of mobility support in named data networking. In *Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on*, pages 83–88. IEEE, 2016.

[20] George Xylomenos, Christopher N Ververidis, Vasilios A Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V Katsaros, and George C Polyzos. A

survey of information-centric networking research. *IEEE Communications Surveys & Tutorials*, 16(2):1024–1049, 2014.

[21] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, et al. Named data networking. *ACM SIGCOMM Computer Communication Review*, 44(3):66–73, 2014.

[22] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica. A data-oriented (and beyond) network architecture. In *ACM SIGCOMM Computer Communication Review*, volume 37, pages 181–192, 2007.

[23] Bengt Ahlgren, Matteo D'Ambrosio, Marco Marchisio, Ian Marsh, Christian Dannewitz, Börje Ohlman, Kostas Pentikousis, Ove Strandberg, René Rembarz, and Vinicio Vercellone. Design considerations for a network of information. In *Proc. CoNEXT*, page 66, 2008.

[24] Gareth Tyson, Andreas Mauthe, Sebastian Kaune, Paul Grace, and Thomas Plagemann. Juno: An adaptive delivery-centric middleware. In *Proc. CCNC*, pages 587–591, 2012.

[25] Nikos Fotiou, Pekka Nikander, Dirk Trossen, George C Polyzos, et al. Developing information networking further: From psirp to pursuit. In *Broadnets*, pages 1–13, 2010.

[26] Ivan Seskar, Kiran Nagaraja, Sam Nelson, and Dipankar Raychaudhuri. Mobilityfirst future internet architecture project. In *Proc. AINTEC*, pages 1–3, 2011.

[27] Zhenkai Zhu, Ryuji Wakikawa, and Lixia Zhang. A survey of mobility support in the internet. *RFC 6301*, March 2011.

[28] Gareth Tyson, Nishanth Sastry, Ruben Cuevas, Ivica Rimac, and Andreas Mauthe. A survey of mobility in information-centric networks. *Communications of the ACM*, 56(12):90–98, 2013.

[29] Bohao Feng, Huachun Zhou, and Qi Xu. Mobility support in named data networking: a survey. *EURASIP Journal on Wireless Communications and Networking*, 2016(1):220, 2016.

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

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

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

[33] Do-hyung Kim, Jong-hwan Kim, Yu-sung Kim, Hyun-soo Yoon, and Ikjun Yeom. Mobility support in content centric networks. In *Proc. of ACM ICN 2012.*

[34] Alexander Afanasyev, Cheng Yi, Lan Wang, Beichuan Zhang, and Lixia Zhang. SNAMP: Secure namespace mapping to scale NDN forwarding. In *Proc. Computer Communication Workshops*, pages 281–286, 2015.

[35] Do-hyung Kim, Jong-hwan Kim, Yu-sung Kim, Hyun-soo Yoon, and Ikjun Yeom. End-to-end mobility support in content centric networks. *International Journal of Communication Systems*, 28(6):1151–1167, 2015.

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

[37] L. Wang, O. Waltari, and J. Kangasharju. Mobiccn: Mobility support with greedy routing in content-centric networks. In *Proc. of IEEE GLOBECOM*, 2013.

[38] Dookyoon Han, Munyoung Lee, Kideok Cho, T. Kwon, and Yanghee Choi. Publisher mobility support in content centric networks. In *Proc. of ICOIN*, 2014.

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

[40] Rao Ying, Luo Hongbin, Gao Deyun, Zhou Huachun, and Zhang Hongke. Lbma: A novel locator based mobility support approach in named data networking. *China Communications*, 11(4):111–120, 2014.

[41] Yo Nishiyama, Masanori Ishino, Yuki Koizumi, Toru Hasegawa, Kohei Sugiyama, and Atsushi Tagami. Proposal on routing-based mobility architecture for ICN-based cellular networks. In *Proc. Computer Communication Workshops*, pages 467–472, 2016.

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

[43] George Xylomenos, Xenofon Vasilakos, Christos Tsilopoulos, Vasilios A Siris, and George C Polyzos. Caching and mobility support in a publish-subscribe internet architecture. *IEEE Communications Magazine*, 50(7), 2012.

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

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

[46] Jordan Augé, Giovanna Carofiglio, Giulio Grassi, Luca Muscariello, Giovanni Pau, and Xuan Zeng. Map-me: Managing anchor-less producer mobility in information-centric networks. *arXiv preprint arXiv:1611.06785*, 2016.

[47] Aytac Azgin, Ravishankar Ravindran, and Guoqiang Wang. A scalable mobility-centric architecture for named data networking. *arXiv preprint arXiv:1406.7049*, 2014.

[48] Frederik Hermans, Edith Ngai, and Per Gunningberg. Global source mobility in the content-centric networking architecture. In *Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications*, pages 13–18. ACM, 2012.

[49] Do-hyung Kim, Jong-hwan Kim, Yu-sung Kim, Hyun-soo Yoon, and Ikjun Yeom. Mobility support in content centric networks. In *Proceedings of the second edition of the ICN workshop on Information-centric networking*, pages 13–18. ACM, 2012.

[50] Dawei Li and Mooi Choo Cuah. Scom: A scalable content centric network architecture with mobility support. In *Mobile Ad-hoc and Sensor Networks (MSN), 2013 IEEE Ninth International Conference on*, pages 25–32. IEEE, 2013.

[51] Jihoon Lee, Sungrae Cho, and Daeyoub Kim. Device mobility management in content-centric networking. *IEEE Communications Magazine*, 50(12), 2012.

[52] Liang Wang, Otto Waltari, and Jussi Kangasharju. Mobiccn: Mobility support with greedy routing in content-centric networks. In *Proc. of GLOBECOM*, 2013.

[53] Moni Naor and Kobbi Nissim. Certificate revocation and certificate update. *IEEE Journal on selected areas in communications*, 18(4):561–570, 2000.

[54] Andrew T Campbell, Javier Gomez, Sanghyo Kim, András Gergely Valkó, Chieh-Yih Wan, and Zoltán R Turányi. Design, implementation, and evaluation of cellular ip. *IEEE personal communications*, 7(4):42–49, 2000.

[55] Subir Das, Archan Misra, and Prathima Agrawal. Telemip: telecommunications-enhanced mobile ip architecture for fast intradomain mobility. *IEEE Personal Communications*, 7(4):50–58, 2000.

[56] K. Butler, T. R. Farley, P. McDaniel, and J. Rexford. A survey of bgp security issues and solutions. *Proceedings of the IEEE*, 98(1):100–122, Jan 2010.

[57] Geoff Huston, Mattia Rossi, and Grenville Armitage. Securing bgp - a literature survey. *IEEE Communications Surveys & Tutorials*, 13(2):199–222, 2011.

[58] Stephen Kent, Charles Lynn, and Karen Seo. Secure border gateway protocol (s-bgp). *IEEE Journal on Selected areas in Communications*, 18(4):582–592, 2000.

[59] Russ White. Securing bgp through secure origin bgp (sobgp). *Business Communications Review*, 33(5):47–53, 2003.

[60] Paul C van Oorschot, Tao Wan, and Evangelos Kranakis. On interdomain routing security and pretty secure bgp (psbgp). *ACM Transactions on Information and System Security (TISSEC)*, 10(3):11, 2007.

[61] Sandra Murphy and Madelyn Badger. Ospf with digital signatures. rfc 2154. 1997.

[62] Tao Wan, Evangelos Kranakis, and Paul C van Oorschot. S-rip: A secure distance vector routing protocol. In *International Conference on Applied Cryptography and Network Security*, pages 103–119. Springer, 2004.

[63] Hari Balakrishnan, Venkata N Padmanabhan, Srinivasan Seshan, and Randy H Katz. A comparison of mechanisms for improving tcp performance over wireless links. *IEEE/ACM transactions on networking*, 5(6):756–769, 1997.

[64] A. Bakre and B.R. Badrinath. I-TCP: indirect TCP for mobile hosts. In *Proc. of IEEE ICDCS*, 1995.

[65] K. Brown and S. Singh. A network architecture for mobile computing. In *Proc. of IEEE INFOCOM*, 1996.

[66] S. Kopparty, S. V. Krishnamurthy, M. Faloutsos, and S. K. Tripathi. Split TCP for mobile ad hoc networks. In *Proc. of IEEE GLOBECOM*, 2002.

[67] Christina Parsa and J. J. Garcia-Luna-Aceves. Improving TCP Performance over Wireless Networks at the Link Layer. *Mob. Netw. Appl.*, 5(1), Mar 2000.

[68] KwangSik Shin, Jinhyuk Kim, and Sang Bang Choi. Loss Recovery Scheme for TCP Using MAC MIB over Wireless Access Networks. *Communications Letters, IEEE*, 15(10), 2011.

[69] Hari Balakrishnan, Srinivasan Seshan, Elan Amir, and Randy H. Katz. Improving TCP/IP Performance over Wireless Networks. In *Proc. of ACM MOBICOM*, 1995.

[70] Antonio DeSimone, Mooi Choo Chuah, and On-Ching Yue. Throughput performance of transport-layer protocols over wireless lans. In *Global Telecommunications Conference, 1993, including a Communications Theory Mini-Conference. Technical Program Conference Record, IEEE in Houston. GLOBECOM'93., IEEE*, pages 542–549. IEEE, 1993.

[71] S. Biaz and N.F. Vaidya. Distinguishing congestion losses from wireless transmission losses: a negative result. In *Computer Communications and Networks, 1998.*, Oct 1998.

[72] Y. Tobe, Y. Tamura, A. Molano, S. Ghosh, and H. Tokuda. Achieving moderate fairness for UDP flows by path-status classification. In *Proc. of IEEE LCN*, 2000.

[73] Saverio Mascolo, Claudio Casetti, Mario Gerla, M. Y. Sanadidi, and Ren Wang. TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links. In *Proc. of ACM MOBICOM*, 2001.

[74] Song Cen, P.C. Cosman, and G.M. Voelker. End-to-end differentiation of congestion and wireless losses. *Networking, IEEE/ACM Transactions on*, 11(5), Oct 2003.

[75] Marica Amadeo, Antonella Molinaro, Claudia Campolo, Manolis Sifalakis, and Christian F. Tschudin. Transport layer design for named data wireless networking. In *Prof. of IEEE INFOCOM NOM*, 2014.

[76] Longzhe Han, Seung-Seok Kang, Hyogon Kim, and H.P. In. Adaptive Retransmission Scheme for Video Streaming over Content-Centric Wireless Networks. *Communications Letters, IEEE*, 17(6), June 2013.

[77] Klaus Schneider, Cheng Yi, Beichuan Zhang, and Lixia Zhang. A practical congestion control scheme for named data networking. In *Proceedings of the 2016 conference on 3rd ACM Conference on Information-Centric Networking*, pages 21–30. ACM, 2016.

[78] Spyridon Mastorakis, Alexander Afanasyev, Ilya Moiseenko, and Lixia Zhang. ndnSIM 2: An updated NDN simulator for NS-3. Technical Report NDN-0028, Revision 2, NDN, Nov 2016.

[79] Charles Perkins, IP Mobility Support Network Working Group, et al. Rfc 2002. *IP Mobility Support*, 70, 1996.

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

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

[82] Jordan Augé, Giovanna Carofiglio, Giulio Grassi, Luca Muscariello, Giovanni Pau, and Xuan Zeng. MAP-Me: Managing Anchor-less Producer Mobility in Content-Centric Networks. Technical report, `https://mapme-tnsm17.github.io/`, 2016.

[83] Lan Wang, AKMM Hoque, Cheng Yi, Adam Alyyan, and Beichuan Zhang. Ospfn: An ospf based routing protocol for named data networking. 2012.

[84] AKM Hoque, Syed Obaid Amin, Adam Alyyan, Beichuan Zhang, Lixia Zhang, and Lan Wang. Nlsr: named-data link state routing protocol. In *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, pages 15–20, 2013.

[85] Pierre Francois, Clarence Filsfils, John Evans, and Olivier Bonaventure. Achieving sub-second igp convergence in large ip networks. *SIGCOMM Comput. Commun. Rev.*, 35(3):35–44, Jul 2005.

[86] Ndnsim simulator. `http://ndnsim.net`.

[87] Andrew Mcgregor and Derek Smithies. Rate adaptation for 802.11 wireless networks: Minstrel. In *Submitted to ACM SIGCOMM 2010, http://blog.cerowrt.org/papers/minstrel-sigcomm-final.pdf.*

[88] wiki. Random waypoint model.

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

[90] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura 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.

[91] Pierre Francois, Clarence Filsfils, John Evans, and Olivier Bonaventure. Achieving sub-second igp convergence in large ip networks. *ACM SIGCOMM Computer Communication Review*, 35(3):35–44, 2005.

[92] Jeffrey Cichonski, Joshua M Franklin, and Michael Bartock. Guide to lte security. *DRAFT NIST Special Publication 800-187*, 2016.

[93] Daksha Bhasker. 4g lte security for mobile network operators. *Cyber Secur. Inf. Sys. Inf. Anal. Cent.(CSIAC)*, 1(4):20–29, 2013.

[94] Leslie Lamport. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–772, 1981.

[95] Bart Preneel. Cryptographic hash functions. *Transactions on Emerging Telecommunications Technologies*, 5(4):431–448, 1994.

[96] openwrt benchmark result. `https://wiki.openwrt.org/doc/howto/benchmark.openssl`, 2017.

[97] OCTEON III CN7020. `http://www.cavium.com/new/Table.html\#Octeonplus`, 2016.

[98] Cesar Ghali, Marc A. Schlosberg, Gene Tsudik, and Christopher A. Wood. Interest-based access control for content centric networks (extended version). *CoRR*, abs/1505.06258, 2015.

[99] Don Coppersmith and Markus Jakobsson. Almost optimal hash sequence traversal. In *International Conference on Financial Cryptography*, pages 102–119. Springer, 2002.

[100] Hari Balakrishnan and Randy H Katz. Explicit loss notification and wireless web performance. In *Proc. of IEEE GLOBECOM Internet Mini-Conference*, 1998.

[101] S. Biaz and N.F. Vaidya. Discriminating congestion losses from wireless losses using inter-arrival times at the receiver. In *Proc. of IEEE ASSET'99*, 1999.

[102] K. Ratnam and I. Matta. WTCP: an efficient mechanism for improving TCP performance over wireless links. In *Proc of IEEE ISCC*, 1998.

[103] G. Carofiglio, M. Gallo, L. Muscariello, M. Papalini, and Sen Wang. Optimal Multipath Congestion Control and Request Forwarding in Information-Centric Networks. In *Proc. of IEEE ICNP*, 2013.

[104] Vivek Mhatre and Konstantina Papagiannaki. Using smart triggers for improved user performance in 802.11 wireless networks. In *Proc. of ACM MobiSys*, 2006.

[105] Eldad Perahia and Robert Stacey. *Next Generation Wireless LANs 802.11n and 802.11ac*. Cambridge University Press, 2 edition, 2013.

[106] Mauro Sardara, Luca Muscariello, Jordan Augé, Marcel Enguehard, Alberto Compagno, and Giovanna Carofiglio. Virtualized icn (vicn): towards a unified network virtualization framework for icn experimentation. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*, pages 109–115. ACM, 2017.

[107] Noor Abani, Torsten Braun, and Mario Gerla. Proactive caching with mobility prediction under uncertainty in information-centric networks. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*, pages 88–97. ACM, 2017.

[108] Srinivasan Keshav and Samuel P Morgan. Smart retransmission: Performance with overload and random losses. In *INFOCOM'97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE*, volume 3, pages 1131–1138. IEEE, 1997.

[109] Dookyoon Han, Munyoung Lee, Kideok Cho, T. Kwon, and Y. Choi. Publisher mobility support in content centric networks. In *The International Conference on Information Networking 2014 (ICOIN2014)*, pages 214–219, Feb 2014.

[110] A. T. Campbell, J. Gomez, S. Kim, A. G. Valko, Chieh-Yih Wan, and Z. R. Turanyi. Design, implementation, and evaluation of cellular ip. *IEEE Personal Communications*, 7(4):42–49, Aug 2000.

[111] Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Borje Ohlman. A survey of information-centric networking. *IEEE Communications Magazine*, 50(7), 2012.

[112] B.S. Bakshi, P. Krishna, N.H. Vaidya, and D.K. Pradhan. Improving performance of tcp over wireless networks. In *Distributed Computing Systems, 1997., Proceedings of the 17th International Conference on*, May 1997.

[113] Thomas Bonald, Alexandre Proutière, and James W Roberts. Statistical performance guarantees for streaming flows using expedited forwarding. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 1104–1112, 2001.

[114] Peter Boothe, James Hiebert, and Randy Bush. How prevalent is prefix hijacking on the internet. *NANOG36 Talk, February*, 2006.

[115] Giovanna Carofiglio, Luca Muscariello, Michele Papalini, Natalya Rozhnova, and Xuan Zeng. Leveraging icn in-network control for loss detection and recovery in wireless mobile networks. In *ICN*, pages 50–59, 2016.

[116] Giovanna Carofiglio, Luca Muscariello, ichele Papalini, Natalya Rozhnova, and Xuan Zeng. Leveraging icn in-network control for loss detection and recovery in wireless mobile networks. In *ACM SIGCOMM ICN'2016*, Kyoto, Japan, Sept. 2016.

[117] Yuh-Shyan Chen, Chih-Shun Hsu, and De-Yi Huang. A pipe-assisted mobility management in named data networking networks. In *Network Operations and Management Symposium (APNOMS), 2014 16th Asia-Pacific*, pages 1–4, 2014.

[118] Alberto Compagno, Xuan Zeng, Luca Muscariello, Giovanna Carofiglio, and Jordan Augé. Secure producer mobility in information-centric network. In *Proc. ACM ICN*, Berlin (DE), Sep 2017.

[119] Truong-Xuan Do and Younghan Kim. Optimal provider mobility in large-scale named-data networking. *KSII Transactions on Internet & Information Systems*, 9(10), 2015.

[120] M. Mosko. CCNx Messages in TLV Format. Internet-Draft draft-irtf-icnrg-ccnxmessages-04, PARC, Inc., 2017.

[121] D Eastlake 3rd and Tony Hansen. Us secure hash algorithms (sha and sha-based hmac and hkdf). Technical report, 2011.

[122] Romano Fantacci, Francesco Chiti, and Leonardo Maccari. Fast distributed bidirectional authentication for wireless sensor networks. *Security and Communication Networks*, 1(1):17–24, 2008.

[123] Dan Forsberg, Yoshihiro Ohba, Basavaraj Patil, Hannes Tschofenig, and Alper Yegin. Protocol for carrying authentication for network access (pana). Technical report, 2008.

[124] B. Francis, V. Narasimhan, A. Nayak, and I. Stojmenovic. Techniques for enhancing tcp performance in wireless networks. In *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*, June 2012.

[125] Christine Fricker, Philippe Robert, James Roberts, and Nada Sbihi. Impact of traffic mix on caching performance in a content-centric network. In *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pages 310–315. IEEE, 2012.

[126] Martin Gaedke, Johannes Meinecke, and Martin Nussbaumer. A modeling approach to federated identity and access management. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 1156–1157. ACM, 2005.

[127] Zhaoyu Gao, Arun Venkataramani, James F Kurose, and Simon Heimlicher. Towards a quantitative comparison of location-independent network architectures. *ACM SIGCOMM Computer Communication Review*, 44(4):259–270, 2015.

[128] Cesar Ghali, Gene Tsudik, Christopher A Wood, and Edmund Yeh. Practical accounting in content-centric networking. In *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*, pages 436–444. IEEE, 2016.

[129] Li Gong. Increasing availability and security of an authentication service. *IEEE Journal on Selected Areas in Communications*, 11(5):657–662, 1993.

[130] Ralf Hauser, Tony Przygienda, and Gene Tsudik. Reducing the cost of security in link-state routing. In *Network and Distributed System Security, 1997. Proceedings., 1997 Symposium on*, pages 93–99. IEEE, 1997.

[131] Yih-Chun Hu, David B Johnson, and Adrian Perrig. Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks. *Ad hoc networks*, 1(1):175–192, 2003.

[132] Yih-Chun Hu, Markus Jakobsson, and Adrian Perrig. Efficient constructions for one-way hash chains. In *International Conference on Applied Cryptography and Network Security*, pages 423–441. Springer, 2005.

[133] Esa Hyytiä and Jorma Virtamo. Random waypoint mobility model in cellular networks. *Wireless Networks*, 13(2):177–188, Apr 2007.

[134] G. Carofiglio, M. Gallo, L. Muscariello, M. Papalini, and Sen Wang. Optimal Multipath Congestion Control and Request Forwarding in Information-Centric Networks. In *Proc. of IEEE ICNP*, 2013.

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

[136] Hendrik Schulze and Klaus Mochalski. Ipoque internet study. Technical report, tech. rep., ipoque GmbH, 2009.

[137] Mohit Lad, Daniel Massey, Dan Pei, Yiguo Wu, Beichuan Zhang, and Lixia Zhang. Phas: A prefix hijack alert system. In *USENIX Security symposium*, volume 1, page 3, 2006.

[138] Jihoon Lee and Daeyoub Kim. Partial path extension scheme for mobile content source in content-centric networking (ccn). *EURASIP Journal on Wireless Communications and Networking*, 2015(1):212, 2015.

[139] open ssl wiki. `https://wiki.openssl.org/index.php/Libcrypto_API`.

[140] Rafa M Lopez, Ashutosh Dutta, Yoshihiro Ohba, Henning Schulzrinne, and Antonio F Gomez Skarmeta. Network-layer assisted mechanism to optimize authentication delay during handoff in 802.11 networks. In *Mobile and Ubiquitous Systems: Networking & Services, 2007. MobiQuitous 2007. Fourth Annual International Conference on*, pages 1–8. IEEE, 2007.

[141] Eve Maler and Drummond Reed. The venn of identity: Options and issues in federated identity management. *IEEE Security & Privacy*, 6(2), 2008.

[142] Kazuhisa Matsuzono, Hitoshi Asaeda, and Thierry Turletti. Low latency low loss streaming using in-network coding and caching. In *IEEE INFOCOM*, 2017.

[143] Patrick McDaniel, William Aiello, Kevin Butler, and John Ioannidis. Origin authentication in interdomain routing. *Computer Networks*, 50(16):2953–2980, 2006.

[144] Mehta Miten and Vaidya Nitin. Delayed duplicate-acknowledgements: A proposal to improve performance of tcp on wireless links. Technical report, College Station, TX, USA, 1998.

[145] G. Grassi, D. Pesavento, G. Pau, L. Zhang, and S. Fdida. Navigo: Interest forwarding by geolocations in vehicular named data networking. *CoRR*, abs/1503.01713, 2015.

[146] V. Jacobson et al. L.Zhang, 2010. NSF Named-Data Networking (NDN) project `http://named-data.net`.

[147] G. Carofiglio, M. Gallo, L. Muscariello, and L. Papalini. Multipath congestion control in content-centric networks. In *Proc. of IEEE INFOCOM NOMEN*, 2013.

[148] Supporting evolved packet core for one million mobile subscribers with four intel xeon processor-based servers. `https://networkbuilders.intel.com/docs/MESH_Group_Intel_EPC_TB_FINAL.pdf`.

[149] William Pugh. Skip lists: a probabilistic alternative to balanced trees. *Communications of the ACM*, 33(6):668–676, 1990.

[150] Jian Qiu, Lixin Gao, Supranamaya Ranjan, and Antonio Nucci. Detecting bogus bgp route information: Going beyond prefix hijacking. In *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*, pages 381–390. IEEE, 2007.

[151] G. Xylomenos, X. Vasilakos, C. Tsilopoulos, V. A. Siris, and G.C. Polyzos. Caching and mobility support in a publish-subscribe internet architecture. *IEEE Communications Magazine*, 50(7):52–58, 2012.

[152] M. Shand and S. Bryant. A Framework for Loop-Free Convergence. RFC 5715 (Informational), Jan 2010.

[153] C. Perkins. IP Mobility Support for IPv4, Revised. RFC 5944 (Proposed Standard), Nov 2010.

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

[155] JW Roberts. Engineering for quality of service. *Self-Similar Network Traffic and Performance Evaluation*, 401420, 2000.

[156] Jacques Samain, Giovanna Carofiglio, Luca Muscariello, Michele Papalini, Mauro Sardara, Michele Tortelli, and Dario Rossi. Dynamic adaptive video streaming: Towards a systematic comparison of icn and tcp/ip. *IEEE Transactions on Multimedia*, 2017.

[157] N.K.G. Samaraweera. Non-congestion packet loss detection for tcp error recovery using wireless links. *Communications, IEE Proceedings-*, 146(4), Aug 1999.

[158] Jan Seedorf, Bilal Gill, Dirk Kutscher, Benjamin Schiller, and Dirk Kohlweyer. Demo overview: fully decentralised authentication scheme for icn in disaster scenarios. In *Proceedings of the 1st international conference on Information-centric networking*, pages 191–192. ACM, 2014.

[159] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

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

[161] Jatinder Pal Singh. Authentication on the edge: Distributed authentication for a global open wi-fi network. *Online] http://www. standford. edu/'jatinder/academic/publications/year/2007/mobilcom2007*, pages 1–12, 2007.

[162] JY Hon So and Jidong Wang. Micro-hip a hip-based micro-mobility solution. In *Communications Workshops, 2008. ICC Workshops' 08. IEEE International Conference on*, pages 430–435. IEEE, 2008.

[163] Neil Spring, Ratul Mahajan, and David Wetherall. Measuring isp topologies with rocketfuel. In *ACM SIGCOMM Computer Communication Review*, volume 32, pages 133–145. ACM, 2002.

[164] G. Tyson, N. Sastry, I. Rimac, R. Cuevas, and A. Mauthe. A survey of mobility in information-centric networks: Challenges and research directions. In *Proc. of NOM*, NoM '12, New York, NY, USA, 2012. ACM.

[165] TODO. Todo. *TODO*, TODO.

[166] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard. Networking named content. In *Proc. of ACM CoNEXT '09*.

[167] Hao Yang, Haiyun Luo, Fan Ye, Songwu Lu, and Lixia Zhang. Security in mobile ad hoc networks: challenges and solutions. *IEEE wireless communications*, 11(1):38–47, 2004.

[168] R. Yavatkar and N. Bhagawat. Improving end-to-end performance of tcp over mobile internetworks. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*, Dec 1994.

[169] Yingdi Yu, Alexander Afanasyev, David Clark, Van Jacobson, Lixia Zhang, et al. Schematizing trust in named data networking. In *Proceedings of the 2nd International Conference on Information-Centric Networking*, pages 177–186. ACM, 2015.

[170] Zheng Zhang, Ying Zhang, Y Charlie Hu, Z Morley Mao, and Randy Bush. ispy: Detecting ip prefix hijacking on my own. *IEEE/ACM Transactions on Networking (TON)*, 18(6):1815–1828, 2010.

[171] Changxi Zheng, Lusheng Ji, Dan Pei, Jia Wang, and Paul Francis. A light-weight distributed scheme for detecting ip prefix hijacks in real-time. In *ACM SIGCOMM Computer Communication Review*, volume 37, pages 277–288. ACM, 2007.

[172] Zhenkai Zhu, Sen Wang, Xu Yang, Van Jacobson, and Lixia Zhang. Act: audio conference tool over named data networking. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, pages 68–73. ACM, 2011.

[173] NS-3. NS-3 simulator, https://www.nsnam.org/.