Intelligent Seamless Handover in Next Generation Networks

Mohammed Al-Khalidi, *Senior Member, IEEE,* Rabab Al-Zaidi, *Member, IEEE,* Nikolaos Thomos, *Senior Member, IEEE,* and Martin J. Reed, *Member, IEEE*

***Abstract*—Providing high quality of service (QoS) to mobile end-users, and guaranteeing resilient connectivity for healthcare wearables and other mobile devices is a critical component of Industry 5.0. However, one of the biggest difficulties that network operators encounter is the issue of mobility handover, as it can be detrimental to end-users' safety and experience. Although various handover mechanisms have been developed to meet high QoS, achieving optimum handover performance while maintaining sustainable network operation is still an unreached goal. In this paper, random linear codes (RLC) are used to achieve seamless handover, where handover traffic is encoded using RLC and then multicasted to handover destination(s) using a mobility prediction algorithm for destination selection. To overcome the limitations of current IP core networks, we make use of a revolutionary IP-over-Information-Centric Network architecture at the network core that supports highly flexible multicast switching. The combination of the RLC, flexible multicast, and mobility prediction, makes the communication resilient to packet loss and helps to avoid handover failures of existing solutions while reducing overall packet delivery cost, hence offering sustainable mobility support. The performance of the proposed scheme is evaluated using a realistic vehicular mobility dataset and cellular network infrastructure and compared with Fast Handover for Proxy Mobile IPv6 (PFMIPv6). The results show that our scheme efficiently supports seamless session continuity in high mobility environments, reducing the total traffic delivery cost by 44% compared to its counterpart PFMIPv6, while reducing handover delay by 26% and handover failure to less than 2% of total handovers.**

***Index Terms*—Sustainable Mobility, Handover, Industry 5.0, Next Generation Networks, PFMIPv6, IP-over-ICN, Random Linear Codes, Prediction, Markov Chain.**

I. Introduction

DURING the last few years, we have faced a tremendous growth of mobile data traffic demand which is projected to reach 329 exabytes per-month by 2028 [1]. Evolving mobile standards, such as 5G and 6G are promising a 1000 fold capacity improvement to fulfill such a demand, which will inventively lead to increased traffic and deployment costs. Mobile network operators are already increasing the density of their access networks by deploying Heterogeneous Cellular Networks of small-cell base stations (SBSs) including pico-cells and femto-cells that work together with conventional base stations. However, as the SBSs deployment density increases, mobile users experience more frequent handovers when they move from one SBS coverage area to another. These handovers can take place within minutes or less considering the small coverage range of SBSs' (about 10 to 20 meters for femto-cells and less than 200 meters for pico-cells) [2]. More frequent handovers in high mobility environments, potentially, lead to higher handover failure rates. The problem is manifested in 5G networks with the increasing deployment of millimeter wave (mmWave) SBS's that suffer high propagation losses [3]. Therefore, for users frequently moving in and out of the SBS range within short periods, new handover handling mechanisms are required to ensure seamless handover with no service disruption. In this context, we examine pre-emptive handover to multiple destinations as a possible candidate to resolve handover failures. To enable pre-emptive handover, we propose a novel handover management scheme, which is based on three techniques: (a) flexible multicast switching to dynamically switch traffic to the (multiple) candidate destinations; (b) Random Linear Codes (RLCs) that can efficiently protect the data traffic from packet loss and help avoid transmission of redundant data during handover periods; and, (c) mobility prediction to select handover destination(s) in a proactive manner (i.e., where it is most probable that a user will move).

In current mobile core networks, handover to multiple destinations would require IP multicast solutions to transport the handover traffic to the group of neighbouring base stations (gNBs in case of 5G networks). This can be done through an anchor point in the network core, that routes and encapsulates user plane traffic towards mobile users. The necessity of an anchor point arises from the restrictions of conventional IP mobility, which closely associate addressing information with physical location. However, tunneling through an anchor is inefficient and results in sub-optimal (dog-leg) routing, and hence wasting network resources. This is becoming a pressing issue to solve, especially with Industry 5.0 requirements of sustainable and resilient network deployment and operation.

To provide efficient and agile multicast, we adopt our previously proposed architecture for networking that involves linking IP edge networks through an Information-Centric Network (ICN) core [4]. The adopted IP-over-ICN architecture does not impose any change in the way the end-users request their data, and changes are limited to the information-centric core network. The ICN core uses Publish and Subscribe (Pub/Sub) messaging as described in [5], [6]. It separates forwarding from addressing, as well as providing highly flexible stateless multicast. As a result, it removes the requirement for tunneling towards a central anchor point in the core network, thus offering scalable and sustainable network operation.

While there are ICN solutions that introduce the use of codes such as network codes or rateless codes to ICN architectures such as our previous work [7]–[9] and [10], [11] to improve pipelining and enhance Interest messages aggregation; these codes-enabled ICN architectures require modifications of the vanilla ICN protocols and necessitate that all the network is ICN based. Different from these solutions, we employ RLC codes to facilitate seamless mobility management in IP-over-ICN networks through multicasting RLC coded data during

handover.

The use of RLC codes is an efficient way to combat service disruption during handover periods. This is possible as the transmitted data can be recovered by receiving a number of RLC encoded packets equal or slightly larger than that of the source packets. This eliminates the need to specifically put in mechanisms to cope with out of order packets, jitter or even buffering resources. Therefore, in the adopted coding approach, RLC handover traffic is broadcasted/multicasted simultaneously to several destinations during the handover execution. To decide which candidates to multicast the traffic, we use prediction techniques, specifically first- and second-order semi-Markov chains [12] to represent SBS/gNB handover decisions arising from the distribution of Mobile Node (MN) residence times. A Markov process can accurately predict mobile users' future locations and improve the resilience of the handover process [13].

This paper extends our initial solution presented in [14] by enabling prediction to decide the set of target handover candidates, in addition to an extensive analysis and evaluation of the proposed handover solution. In summary, the contributions of our work are as follows:

- the proposal of a make-before-break handover solution based on RLC codes for IP-over-ICN networks that facilitates seamless handover, resilience to data loss and minimum handover failure. Different from [14], our improved solution (termed IP-over-ICN handover for the remainder of this paper) multicasts data to only a set of gNBs instead of all neighboring gNBs;
- the introduction of an offline mobility prediction model based on first- and second-order semi-Markov chains [12] that can accurately predict mobile users' future locations. Although IP-over-ICN handover employs semi-Markov chains, it is generic and transparent to the prediction method. Thus, other prediction algorithms can be applied;
- theoretical analysis of the signaling, packet delivery, and latency cost of IP-over-ICN handover and comparison with the corresponding cost of the Proxy MIPv6 solution;
- extensive evaluation of IP-over-ICN handover through simulations using a publicly available realistic mobility dataset that describes vehicular mobility within the Cologne metropolitan region in Germany, in addition to the actual deployment of cellular infrastructure for the same region [15]. Evaluation results show that the proposed solution reduces the total traffic delivery cost by 44% compared to its counterpart PFMIPv6, while reducing handover delay by 26% and handover failure to less than 2% of total handovers.

The remainder of the paper is organized as follows: Section II presents an overview of the handover problem and related work. Section III introduces IP-over-ICN handover, while a formal modeling of the mobility cost analysis is provided in Section IV. The evaluation of the proposal is presented in Section V, where the simulation results are discussed. The paper's conclusions are presented in Section VI.

II. Overview of Handover Problem and Related Work

The process of mobility handover can significantly degrade performance in cellular networks. As a result, standardization bodies like the Third Generation Partnership Project (3GPP) and the Internet Engineering Task Force (IETF) have developed different standardized handover mechanisms to improve the quality of service provided to end-users. [16].

To enable mobility in cellular networks, 3GPP has defined the General Packet Radio Service (GPRS) Tunneling Protocol (GTP) [17], which anchors user plane and control plane traffic at specific core entities. In 5G networks, the User Plane Function (UPF) serves as the anchor for user plane traffic, while the Access and Mobility Management Function (AMF) anchors control plane traffic [18]. Handover in 5G networks can be accomplished in two ways. The first method is through the Xn interface, which establishes a direct connection between gNBs. Alternatively, handover can be performed through the N2 interface between the gNB and the AMF when an Xn handover is not feasible, for example due to new radio conditions, load balancing, lack of Xn connectivity to the target gNB, etc [19]. The decision to initiate a handover process is made by the serving gNB. This process comprises three distinct phases, namely handover preparation, execution, and completion. To determine when a handover is necessary, the serving gNB relies on measurement reports from the MN, which include indicators of the radio signal strength of both the serving and neighboring cells, as perceived by the MN. If the handover is conducted over the Xn interface, the serving gNB sends downlink packets to the target gNB over the interface to ensure that there is no packet loss during handover execution [20], [21].

IETF has proposed the Proxy-Based Fast Mobile IPv6 Protocol (PFMIPv6), an advancement of Proxy Mobile IPv6 (PMIPv6) [22], where the Local Mobility Anchor (LMA) is the central topological anchor point for the MNs home network prefix(es). In PFMIPv6, the Mobile Access Gateway (MAG) is responsible for detecting MNs' movements to and from the access link and for binding registrations to the MNs LMA. During handover, a bidirectional tunnel is created between the serving MAG and the target MAG, which is used to transfer packets intended for the MN [23]. The main disadvantage of PFMIPv6 is that the tunneling cost explodes when preparing multiple handover destinations. The authors of this paper have shown, in an earlier work, that an IP-over-ICN architecture can overcome these costs [14], [24]. Specifically, these works demonstrated how the IP-over-ICN architecture can avoid the tunneling through a fixed anchor point and, thus, generally improving efficiency in the network. The method proposed in this paper builds on this concept but uses flexible multicast and handover prediction to overcome the disadvantages of earlier approaches as explained in later sections.

From the above discussion, it is apparent that there is still a pressing need for new HO solutions that are able to guarantee high QoS to the end-users. To achieve this, a large number of research efforts focus on using Software Defined Networking (SDN) for mobility management [25].

However, most of these SDN approaches cannot be directly applied to large-scale networks due to the fact that mobile flow entries are tested against matching rule fields through every OpenFlow switch along the path. This imposes high costs in mobile flow management. In [26], the authors propose a software-defined seamless handover strategy based on passive wireless link quality metrics. This strategy follows a handover management algorithm composed of handover decision and execution methods that improve the mobility experience and provide make-before-break handover. Differently, in [27], a mobility and available resource estimation strategy based on SDN is presented so that seamless handover is supported. In this system, a Markov chain formulation is utilized to estimate the transition probabilities of mobile nodes between neighbor base stations as well as their probabilities of resource availability. By utilizing this approach, it becomes possible to select the most suitable target base stations and assign them virtually to the mobile nodes. All the connections are then established using OpenFlow tables. Also in [28], a SDN based 5G core is proposed, that is based on the standard 3GPP 5G architecture. The work aims to bring about flexibility, simplified management, and eliminate vendor dependence within the network. Noticeably, all efforts described above involve updating the OpenFlow rules in every SDN router along the users path and require an individual rule for every MN. This would lead to flow-table exhaustion in typical large-scale deployments.

Other research studies have focused on optimizing HO parameters such as ping-pong HO, radio link failure and HO latency to facilitate seamless handover. In [29], the authors propose a fuzzy-coordinated self-optimizing HO scheme to achieve seamless HO while users move in multi-radio access networks. The proposed scheme aims to resolve the conflict between mobility robustness and load balancing functions by utilizing a fuzzy system considering three input parameters: signal-to-interference-plus-noise ratio, cell load and UE speed. Also in [30], the authors present a new model for selecting the best network during vertical handover based on a technique known as Improved-MEREC-TOPSIS. The objective priority criteria weight and the TOPSIS multi-attribute decision-making technique are combined in this work to rank potential networks and select the optimal network using a threshold to ensure an efficient and seamless handover decision. Although the efforts discussed above improve the handover success rate through parameter optimization, the evaluation results show that achieving an acceptable balance between the handover parameters unavoidably entails a considerable number of handover failures, with unnecessary handovers reaching up to 20% of total handovers. Differently, this paper proposes a fundamental change in handover management where RLC coded traffic is transmitted over the wireless link in the serving gNB in addition to all/several neighbouring gNB's during handover. This enables the MN to receive all downlink packets that are transmitted during handover from the serving gNB or any of the potential target gNB's over the wireless link without having to worry about sequential transmission, out of order packets, jittering or even buffering. Since no decision on a single handover target is needed, handover failure rates are reduced substantially and would only happen upon a handover prediction error.

III. IP-over-ICN Handover

This paper advocates the novel combination of three approaches to improve the handover performance: (a) RLC to support make-before-break handover, (b) an Information-Centric Core to allow efficient and agile multicast, and (c) Semi-Markov mobility prediction to avoid multicasting to all neighboring SBSs/gNBs. This section discusses these main pillars of the solution in detail.

A. RLC Codes

RLC has been widely studied so only a brief discussion is given here, and interested readers are referred to [31], [32]. In RLC, senders generate packets by combining linearly, at random, packets from the same source (session) or multiple sources (sessions) [33]. This allows maintaining high packet diversity in the network. At the receivers, decoding is possible once a set of coded packets is gathered that has rank equal to the number of source packets. For example, if a source consists of S packets, a receiver can acquire the original source packets after receiving $(K = S + \epsilon)$ packets, where ϵ is a small number. This happens when both the number of source packets (S), and the Galois field $GF(q)$ (which is the finite field where the operations are conducted) are high enough [31]. The extra packets are required to account for the fact that the coded packets arriving to a receiver might be redundant because of the random coding operations followed during RLC encoding. To reduce the probability of generating redundant packets, all the coding operations, i.e., packet combinations, should be performed in high enough $GF(q)$, where q stands for the size of the Galois field.

In any case, a number of RLC coded packets larger than S needs to be received by a user prior to being able to recover the source packets. Theoretically, The number of source packets forming a generation S can be unbounded, but it practically depends on the application, for example, in video streaming case, the number of packets can be equal to the number of packets of a group of pictures, or in MPEG-DASH systems the number of packets forming a chunk. In both examples, the packets comprising a generation have similar decoding constraints. In RLC codes, the higher the generation size and field length is, the higher the encoding/decoding complexity and delay will be, and at the same time, the higher the probability of decoding will be. It is worth to note the tradeoffs among the field size, the decoding probability, the computational complexity and the delay [34], [35]. This ensures that the set of encoding/decoding parameters would be optimized for efficient mobility handover [36], [37].

In IP-over-ICN handover, RLC codes are utilized during handover time, which means that the delay introduced by RLC encoding/decoding operations should be considered. The encoding/decoding delay increases with the $GF(q)$ size and generation length (coding window size). RLC decoding failure can happen in extreme cases, where the extra delay to transfer the additional packets may not be afforded. As shown in

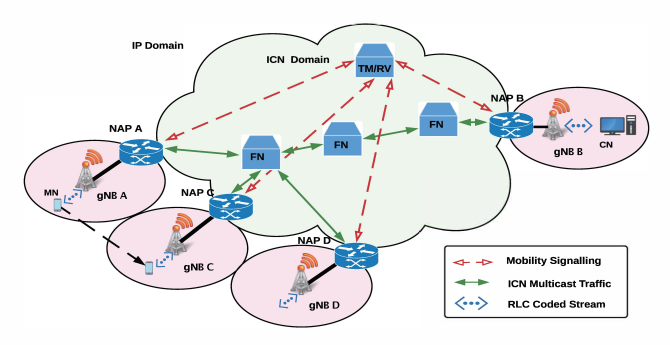

Fig. 1: IP-over-ICN Handover.

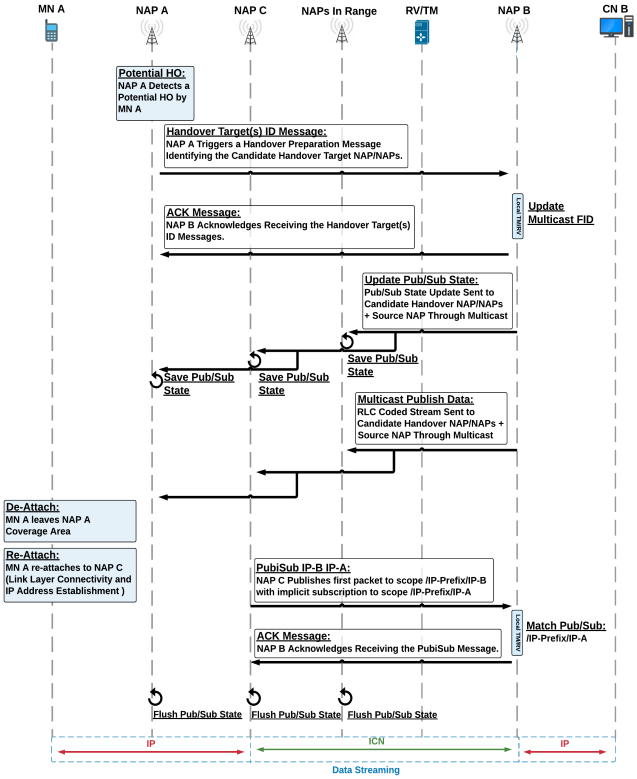

Fig. 2: IP-over-ICN Handover Sequence Diagram.

[38], RLC outperforms all the other transmission schemes in terms of delay with $GF(2^4)$. Therefore, in our evaluation (Section V), we restrict all coding operations to be performed in $GF(2^4)$ with a RLC generation of 100 packets and a RLC coding overhead equal to 0.02. These parameters setting ensure a decoding probability of 99.9%, as shown in [33] while in parallel maintain the encoding/decoding delay and coding complexity low.

B. Agile Multicast through ICN

The architecture that facilitates IP-over-ICN handover follows the gateway approach illustrated in Fig. 1. In our architecture, the NAP is based on the IP protocol and serves as an entry point to the ICN-based core network. Essentially, the NAP maps the IP protocol abstraction to ICN [4]. The employed ICN architecture uses Publish and Subscribe (Pub/Sub) messaging as described in [5]. This architecture maps ICN "names" to Pub/Sub messages using a central network function called rendezvous (RV). The RV works with a topology manager (TM) that is responsible for the forwarding process in the ICN. The TM issues senders with a forwarding ID (FID) that then allows path-based forwarding that can be unicast or multicast. There are also Forwarding Nodes (FN) that simply forward the information object to the Receiver using the specific FID generated for this transmission. It should be noted that the ICN multicast solution is quite different from IP and uses a path based approach where the FID uniquely defines the multicast tree with no multicast state required in the forwarding switches. It has been shown that this is realizable in standard software defined switches without modification and allows a multicast tree to simply be changed by a source changing the FID it places on the outgoing packet without any other signaling in the network [39]. The ICN core is transparent to the IPv4/IPv6 MN through the convergence function of a NAP collocated at the gNB. Therefore, IP becomes a service enabled through the ICN core [4] and IP addresses are ICN *names* rather than identifiers used directly for routing.

With the proposed IP-over-ICN handover solution, when a NAP detects a MN handover, mobility prediction is used to identify handover targets from a set of neighbours (potential handover candidates). Consequently, handover targets are dynamically decided and can be narrowed down to a single or small number of targets based on the prediction accuracy confidence. This approach reduces the handover traffic overhead compared to a naïve approach such as [14] that would send traffic to all neighbours.

To explain the proposed architecture let us assume two devices, MN A and CN B, that communicate and are attached to NAP A and NAP B, respectively. To complete the attachment the following happen: NAP A *subscribes* to receive packets sent to the IP address of MN A, and, likewise NAP B subscribes to receive packets sent to CN B. For MN A to send traffic to CN B, NAP A then *publishes* data to NAP B and similarly for CN B sending to MN A. This Pub/Sub matching is coordinated by the RV and then the TM allocates forwarding IDs (FIDs) for unicast communication to take place.

When MN A moves towards NAP C, a handover is initiated. There could be ambiguity regarding the next NAP towards which MN A is actually transitioning (NAP C could be one of a number of candidates, which are decided using the algorithm presented in Section III-C). An example of the handover process is shown in Fig. 2 where it is assumed that the initial Pub/Sub matching for the unicast communication between MN A and CN B has already occurred. The serving gNB A receives regular measurement reports of signal quality from the MNs which it uses to decide if a potential handover is about to occur. When it detects a handover, gNB A triggers a handover process that consists of three main phases: the first is Handover Preparation, the second is Handover Execution and the third is Handover Completion.

During the Handover Preparation phase, the serving NAP (NAP A which is collocated with gNB A) sends a handover target(s) ID message to NAP B (collocated with gNB B) that includes the ICN identifier of the target neighbouring

NAP/NAPs included in the Handover Preparation, in addition to the ICN identifier of the source NAP itself. The local RV situated at NAP B updates the FID from NAP B to NAP A, replacing it with a multicast FID. This is because there are now multiple destinations involved, including the source and potential target NAP/NAPs for handover. If for any reason, the local RV fails to create the FID, it can always refer to the domain RV/TM that have global knowledge of the network topology. Upon updating the FID, the NAP B acknowledges receiving the handover target(s) ID message and starts sending RLC coded traffic using the new multicast FID. By sending the traffic by multicast to NAP A, and the target neighbouring NAP/NAPs, the traffic can reach MN A whichever neighbouring gNB/NAP it is destined to join. During the handover process, RLC is employed to handle packet losses or delays in packet arrival. When RLC is utilized, the user requests RLC encoded packets instead of a particular uncoded source packet. Each coded packet incorporates information from several original uncoded source packets. Coded packets are useful when they contain novel information compared to the previously received packets. The source data is recovered by means of RLC decoding when a full rank set of coded packets is received by the MN (MN A in this example). The use of RLC removes the need for employing ARQ mechanisms as these can lead to additional congestion within the core network. Furthermore, NAP B transmits a message via multicast with the Pub/Sub state to NAP A and the target neighboring NAP/NAPs included in the handover preparation. This state remains stored at the NAPs that are involved until one of them assumes ownership of the state when the MN enters its coverage zone.

During the execution phase of handover, the L2 link is teared down at the previous serving NAP and established at the new serving NAP. The handover completion phase begins once the link at the new serving NAP is up. This phase involves the re-establishment of the session, where the new serving NAP (NAP C) uses DHCP to initiate IP address establishment while preserving the MN's current IP address. When the initial packet is ready to be sent from MN A to CN B via gNB C/NAP C, or slightly earlier when link-layer connectivity is established, NAP C notifies NAP B that it is publishing to /IP-Prefix/IP-B (an ICN name) and directs NAP B to implicitly subscribe to the scope of MN A's own IP address /IP-Prefix/IP-A (we term this *PubiSub*). Upon receiving the PubiSub message and acknowledging it, NAP B is prompted to discontinue the use of multicast with RLC. Instead, NAP B starts sending unicast responses to NAP C, allowing regular IP-over-ICN traffic to resume between MN A and CN B. At this point, the Pub/Sub state is dropped in all other neighbouring NAPs that participated in the handover process. This is denoted by the "Flush Pub/Sub State" in Fig. 2.

C. Semi-Markov Prediction Model for Mobility

The IP-over-ICN handover solution multicasts RLC encoded traffic to neighboring NAPs when a move to another NAP is likely. This improves the resilience of the communication process and lowers the load of the core network as the traffic is not sent through an anchor point. When the resources are constrained, it is desirable to send the RLC traffic only to a selective set of NAPs where it is most probable that a user will move. Towards this aim, mobility prediction methods should be employed. Next, we discuss the adopted semi-Markov prediction model [40] we used in our system.

*1) **First Order Markov Chains**:* The location of a user is identified by a unique cell ID, as commonly used in cellular networks. While the user location could also be identified by other means such as geographic coordinates, here we are interested in traffic tied to the base station, thus from network's perspective, the location identification via the cell ID is sufficient. To train the mobility prediction, a history of user mobility is recorded from the list of successive visited cells during a user's trip. Specifically, users' mobility history patterns are periodically recorded with: the cell-IDs; the handover count to neighboring cells; and, the residence time (the time spent in the current cell i before moving to the next cell l). Based on that, the cell-transition probabilities $p_{i,l}$ and the distribution of cell residence times can then be directly computed for each location. We assume that each cell, maintains a record of these session residence times and the cell ID of the next-cell transition [12].

A semi-Markov process is used to predict the mobility by using a Markov chain to represent decisions arising from the distribution of residence times. Within the semi-Markov process, time instants represent a user attaching to a new cell whereas the successive state occupancy is described by the transition probability $p_{i,l}$ of the Markov chain. The residence time in a state within the Markov chain depends on the current cell location and the next cell where the user will move. The generalized semi-Markov kernel for a time-homogeneous process is represented as $\Phi_{i,l}(t)$ [41], the probability that after making the transition into state i, there is a transition to state l within time t. $\Phi_{i,l}(t)$ is defined as

$$\Phi_{i,l}(t) = Pr\Big\{X_{n+1} = l, T_{n+1} - T_n \leq t | X_n = i\Big\} \qquad (1)$$

where X_n and X_{n+1} are the state of the system at the nth and $(n+1)$th transition, at times T_n and T_{n+1}. The kernel can also be expressed as $\Phi_{i,l}(t) = P_{i,l}\Psi_{i,l}(t)$, where

$$\Psi_{i,l}(t) = Pr\Big\{T_{n+1} - T_n \leq t | X_{n+1} = l, X_n = i\Big\} \qquad (2)$$

$\Psi_{i,l}(t)$ is the conditional probability that a transition from i to l will take place within time t. The residence times in this semi-Markov process can obey an arbitrary distribution. This is a useful departure from the common assumption that residence times are exponentially distributed, thus permitting a more general representation of the temporal behavior [42]. In the limit as time tends to infinity, $\Psi_{i,l}$ tends to one, we have $P_{i,l} = \lim_{t\to\infty} \Phi_{i,l}(t)$. Hence, the transition probability matrix $P = [P_{ij}]$, $\forall\ i,\ j \in [1, n]$ is given by

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & p_{2,3} & \cdots & p_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & p_{n,3} & \cdots & p_{n,n} \end{bmatrix} \qquad (3)$$

We also define the kernel $\phi_{i,l}(t) = p_{i,l}\psi_{i,l}(t)$ where $\psi_{i,l}(t)$ represents the residence time given by

$$\psi_{i,l}(t) = Pr\Big\{T_{n+1} - T_n = t | X_{n+1} = l, X_n = i\Big\} \quad (4)$$

and

$$\phi_{i,l}(t) = Pr\Big\{X_{n+1} = l, T_{n+1} - T_n = t | X_n = i\Big\} \quad (5)$$

The state transition probability matrix P is initialized as:

$$p_{i,l} = \frac{|H_{i,l}|}{|H_i|} \quad (6)$$

and the residence time distribution matrix Ψ is initialized as:

$$\psi_{i,l}(\tau) = \frac{|H_{i,l,\tau}|}{|H_{i,l}|} \quad (7)$$

where $H_{i,l}$ represents the count of handovers from cell i to l, while H_i represents the overall count of handovers for users from cell i. $H_{i,l,\tau}$ denotes the handover count of users from i to l specifically within a residence time interval of τ. When there is a handover from cell i to l, $p_{i,l}$ and $\psi_{i,l}(\tau)$ and $\phi_{i,l}(\tau)$ are updated. The cell with the highest probability, $\phi_{i,l}(\tau)$, is chosen as the predicted future destination when the time spent in cell i falls within time interval τ. The prediction algorithm can be utilized for both offline and online learning. In the offline case, a training phase may be necessary before implementing the learned prediction matrices. In the online case, the probabilities are continuously updated based on the recorded cell transitions. This ensures that any changes in the mobility behavior directly update the prediction probabilities. As operators are aware of the transition statistics which tend to follow usual daily and weekly patters, we focus on the offline prediction model here.

*2) **Second Order Markov Chains**:* After describing the semi-Markov process above, a second-order Markov chain is derived in which the occupancy of successive states is determined by the transition probabilities of the Markov process. In this second-order Markov chain, the semi-Markov process depends on the previous state, current state, and next state transition, while the residence time spent in any state depends on the previous state the user visited, as well as the current and next states the user is expected to move to. In this context, we assume that each cell in the network that is recording a profile for mobility pattern consisting of the count of handovers to neighboring cells, as well as the residence time (the time spent in the current cell i before transition to the next cell l, given that the previous cell attachment was h). Consequently, the cell-transition probabilities $p_{h,i,l}$ and the distribution of cell residence times at each cell can be directly computed.

The semi-Markov kernel for a second order time-homogeneous process with transition probabilities $p_{h,i,l}$ is given by $\Phi'_{h,i,l}(t)$, which denotes the probability that immediately after making the transition into state i from state h, the process makes a transition to state l within time t. $\Phi'_{h,i,l}(t)$ is defined as

$$\Phi'_{h,i,l}(t) = Pr\Big\{X_{n+1} = l, T_{n+1} - T_n \leq t | X_n = i, X_{n-1} = h\Big\} \quad (8)$$

where X_{n-1}, X_n and X_{n+1} are the state of the system at $n-1$, n and $(n+1)$; T_n and T_{n+1} are the times of the nth and $(n+1)$th transitions. Therefore, as we have done for first order Semi-Markov chains, the state transition probability matrix P is initialized as:

$$p_{h,i,l} = \frac{|H_{h,i,l}|}{|H_{h,i}|} \quad (9)$$

and the residence time distribution matrix Ψ is initialized as:

$$\psi_{h,i,l}(\tau) = \frac{|H_{h,i,l,\tau}|}{|H_{h,i,l}|} \quad (10)$$

where $H_{h,i,l}$ is the handover count from i to l, given the previous cell was h, and $H_{h,i}$ is the total number of user handovers from cell i, given that the previous attachment cell was h. $H_{h,i,l,\tau}$ is the handover count of users from cell i to l, given that attachment at the previous attachment cell, h, was within a residence time of τ.

*3) **Natural breaks for Residence Time Clustering**:* In order to classify users' residence time within an optimum number of finite residence time clusters, we use classes of natural breaks [43]. Classes of natural breaks are data clustering methods that find the optimum classification of values into distinct classes. The objective is to reduce the average deviation of every class from its respective class mean, while in parallel maximizing the deviation of each class from the means of the other groups. This reduces the in-class variance while maximizing the variance between the classes. The method identifies class boundaries (breaks) that best group convergent values and separate the divergent ones.

The natural breaks clustering method (summarized in Algorithm 1) requires an iterative process, where calculations are repeated using different breaks in the dataset to determine which set of breaks has the smallest in-class variance. The process starts by arbitrarily dividing the numeric data (observed residence times) into groups, and then the following steps are repeated:

- Calculate the sum of squared deviations between classes ($SDBC$).
- Calculate the sum of squared deviations from the array mean ($SDAM$).
- Calculate the squared deviations from the class means ($SDCM$), where:

$$SDCM = SDAM - SDBC \quad (11)$$

- Calculate the goodness of variance fit (GVF) statistics as:

$$GVF = (SDAM - SDCM)/SDAM \quad (12)$$

Note that GVF ranges from 0 (worst fit) to 1 (perfect fit) and that SDAM is a constant value that does not change unless the data changes. After the above calculations, the residence times are then moved from one class to another in an effort to reduce the sum of SDCM and therefore increase the GVF statistic. This process continues until the GVF value can no longer be increased [44]. The Natural breaks method is used because it identifies real classes within the data that have accurate representations of trends. Many alternative cluster analysis methods exist, i.e., Head/tail Breaks, Equal Interval,

Quantile, Standard Deviation, etc. However, investigating these alternatives is out of this work's scope.

Algorithm 1 Residence Time Clustering

1: Divide arbitrarily the observed residence times into a given set of classes.
2: Initialize the Goodness of Variance Fit, $GVF = 0$.
3: Define a GVF target, e.g., $GVF_t = 0.8$.
4: **while** $GVF < GVF_t$ **do**
5: Move residence times from one class to another
6: Compute the sum of the squared deviations between classes $SDBC$.
7: Compute the sum of the squared deviations from the array mean $SDAM$.
8: Compute the squared deviations from the class means as $SDCM = SDAM - SDBC$.
9: Compute $GVF = (SDAM - SDCM)/SDAM$
10: **end while**

*4) **Computational Complexity**:* With Markov Chain prediction, the accuracy of the prediction grows as the number of states increases. Furthermore, the number of states grow when the movement history (memory dimension) is increased, or when the number of possible directions (direction dimension) is increased. Increasing the memory dimension increases the number of states exponentially, while increasing the direction dimension increases it linearly. With the state-space growing, the computational complexity of the Markov state calculations also increases. As a generalization, the complexity is given by

$$\mathcal{O}(ERd^m) \tag{13}$$

where E represents the number of events (handover processes) in the network, R represents the number of residence time clusters used, d represents the mean number of directions at each cell, i.e., the mean number of neighboring cells, and m represents the Markov chain order which defines the memory length. Since the Markov chain prediction we examine in this paper is of first and second order, the complexity stays reasonable. Therefore, the proposed prediction requires relatively moderate processing and storage capabilities. It should also be pointed out that the prediction model can be created *offline* using recent mobility statistics to tune the probabilities rather than having to carry out the prediction tuning on every new mobility event. The results in this paper shown in Section V were generated using this offline approach.

IV. Analysing the Cost of Mobility Management

This section analyzes the cost of mobility management for IP-over-ICN Handover and compares it to the corresponding cost of the IETF counterpart solution PFMIPv6 [45]. For simplicity, it is assumed that only one end of the communication, MN, is in motion, while the corresponding node, CN, remains stationary. Therefore, the CN does not generate additional mobility signaling.

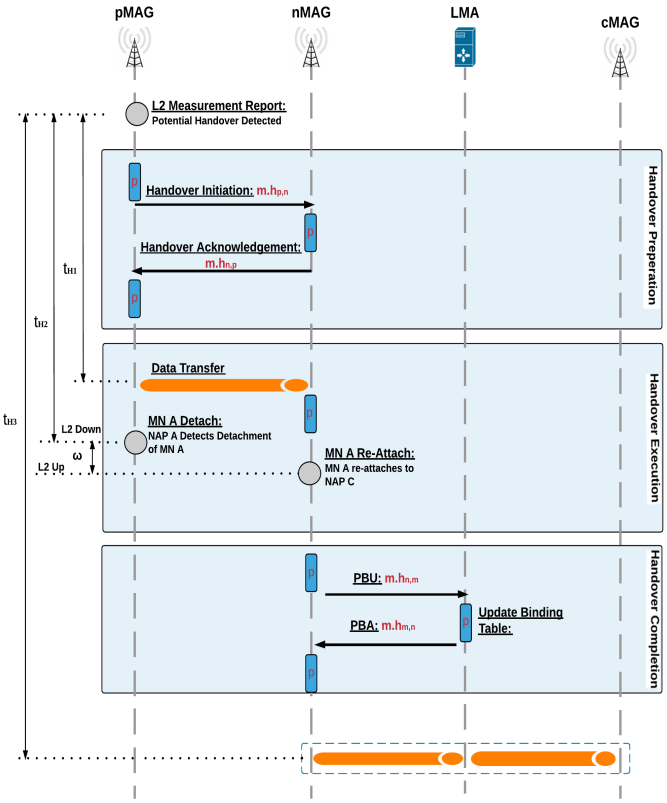

Fig. 3: PFMIPv6 handover time diagram.

A. Proxy-Based Fast Mobile IPv6

In PFMIPv6, when a MN moves from a *previous* MAG (pMAG) to the *next* MAG (nMAG) and the Received Signal Strength Indicator (RSSI) of the MN is detected to be less than a pre-determined threshold, the serving gNB triggers the Handover-Initiate process as described below [22]:

1) The MN finds the neighbouring gNB with the strongest RSSI and reports this together with the strongest (new) gNB to the previous serving gNB that it is about to leave using an *L2 report*.
2) The previous serving gNB then indicates the MN's handover to the pMAG which starts to set up a new IP-in-IP tunnel between the nMAG and itself.
3) Then the pMAG sends a Handover Initiation Request message H_r containing the MN's context information to the nMAG. The nMAG acknowledges this to the pMAG.
4) When the acknowledgement is received, the pMAG starts to forward data packets to the nMAG via the newly created tunnel. Now, the nMAG can forward packets to the next gNB once the MN is connected to the new access network, packets may be buffered during the handover and transmitted when it is complete.

The above is for *Predictive Handover* where the tunnel between the pMAG and the nMAG is established prior to the MN's attachment to the next gNB. If the MN hands over to the next gNB without transmitting a measurement report to the previous gNB, a *Reactive Handover* is applied where tunnel establishment takes place after the MN attachment to the nMAG as described in the standard [45].

1) Mobility Signaling Cost: The signaling cost of PFMIPv6 [46] consists of the proxy binding updates PBU and proxy binding acknowledgements PBA at the pMAG and nMAG sent towards the LMA, along with the signaling overhead of establishing a tunnel between the pMAG and nMAG for forwarding handover packets. The total signaling cost for successful PFMIPv6 handover (predictive or reactive) is given by:

$$\begin{aligned}\Upsilon &= \big\{h_{p,l}(|PBU|+|PBA|) \\ &+ h_{n,l}(|PBU|+|PBA|)+h_{p,n}(|H_r|+|H_a|)\big\},\end{aligned} \tag{14}$$

where $h_{p,l}$, $h_{n,l}$, and $h_{p,n}$ represent the hop count between pMAG and the LMA, the nMAG and the LMA, and the pMAG and nMAG respectively. $|H_r|$ [1] corresponds to the message size in *bytes* of the handover initiation request sent from pMAG to nMAG (in case of predictive handover) or from nMAG to pMAG (in case of reactive handover). Similarly, $|H_a|$ corresponds to the message size in *bytes* of the handover acknowledgment sent from nMAG to the pMAG (in case of predictive handover) or from pMAG to the nMAG (in case of reactive handover). By setting $|PB|$ = $|PBU|$ + $|PBA|$, equation (14) can be rewritten as:

$$\Upsilon = \big\{|PB|(h_{p,l}+h_{n,l})+h_{p,n}(|H_r|+|H_a|)\big\} \tag{15}$$

2) Mobility Packet Delivery Cost: Λ is the packet delivery cost necessary for facilitating Fast Handover in PMIPv6, which encompasses the packet delivery overhead. This cost is determined by multiplying the average packet arrival rate in *packets/sec*, the size of each packet in *bytes*, and the hop distance. In the case of PFMIPv6, the packet delivery cost is measured in *Bytes×Hops/Sec* and expressed as:

$$\Lambda = RO, \tag{16}$$

where R is the average packet arrival rate, and O is the direct path packet cost in PFMIPv6. O is determined by:

$$O = (h_{c,l}+h_{l,p}+h_{p,n})(\varphi+\zeta), \tag{17}$$

where $h_{c,l}$, $h_{l,p}$, and $h_{p,n}$ denote the hop count between the CN and the LMA, the LMA and the pMAG, and the pMAG and nMAG, respectively. Finally, φ is the tunneling overhead and ζ is the average data packet length (both in *bytes*).

3) Handover Latency Cost: Handover latency cost is mainly used to investigate the time duration of handover phases (i.e., preparation, execution and completion). Fig. 3 illustrates the seamless handover timing diagram in PFMIPv6 where Γ is the duration of a layer 2 handover, which is the time elapsed from when the MN sends an L2 report message to the previous MAG to when the MN's L2 connection with the next MAG is established. Γ can be expressed as:

$$\Gamma = t_{H2}+\omega, \tag{18}$$

where t_{H2} represents the duration between the delivery of the L2 report message and the occurrence of the L2 link down event, while ω indicates the time between L2 link down event at previous MAG, and L2 link up event at next MAG. We also define γ to be the time duration for handover mode transmission, which is the time from when the MN starts receiving redirected traffic from the previous MAG, or buffered traffic from the next MAG, to when normal traffic is resumed at the next MAG. γ is expressed as:

$$\gamma = \Gamma - t_{H1}, \tag{19}$$

where t_{H1} represents the time duration of pre-handover preparation. Therefore, if $t_{H2} > t_{H1}$, the MN will start receiving buffered traffic after attaching to the next MAG. While if $t_{H1} > t_{H2}$, the MN will start receiving forwarded (non-buffered) traffic.

To allow a straightforward analysis, latency is analyzed in terms of number of exchanged messages, required processes and traversed hops to complete a successful seamless handover. In this analysis, it is assumed that p represents the time to process a message, m represents the time to exchange a message, and h represents the message hop count as shown in the timing diagrams in Fig. 3. For simplicity, we assume that p and m are expressed in arbitrary time units, and both set to 1 time unit. This implies that the link transmission delay is comparable to the forwarding delay. Thus, in the case of PFMIPv6, the cost of handover latency T_c can be expressed as follows:

$$T_c = 7p + 2m.h_{p,n} + 2m.h_{n,l} \tag{20}$$

B. Seamless Handover in IP-over-ICN

This starts with a handover preparation process as illustrated in Fig. 2. As part of this process, the RLC coded traffic is multicast from the corresponding NAP to the handover neighborhood of the MN before the handover execution. Based on Layer 2 measurement reports at the serving access network, i.e., the RSSI of the MN falls below a predefined threshold, the handover preparation process is initiated for the MN, and the subsequent operations are performed:

1) The NAP on the previous link (NAP A) signals the corresponding NAP B by sending a handover target(s) ID message ℓ_s including the ICN identifier of the candidate target handover NAP/NAPs, in addition to the identifier of the source NAP itself.
2) NAP B replaces the previous FID with a multicast FID and employs the new FID to transmit a multicast stream of RLC coded traffic to the identified handover candidate neighboring NAP/NAPs of NAP A in addition to NAP A itself. Upon updating the FID, NAP B acknowledges receiving the handover target(s) ID message by sending an ACK message back to NAP A.
3) NAP B sends a multicasted state update message ℓ_u to NAP A and the identified handover candidate neighboring NAP/NAP's. This message includes MN A's Pub/Sub state to be stored at the NAPs participating in the handover process. This Pub/Sub state is used when MN A moves into the coverage area of one of the participating NAPs.
4) Once Layer 2 connectivity and IP address allocation have been established between MN A and NAP C, the latter then

[1] In this paper, the length of message x is denoted as $|x|$.

receives the first IP packet destined to the CN (at NAP B), and locally looks up the FID needed to reach NAP B. NAP C then uses this FID to send a PubiSub message ℓ_i to NAP B. This PubiSub message includes the first data packet sent from MN A to the CN in addition to an implicit subscription to MN A's own IP address scope.

5) After receiving the PubiSub message, NAP B utilizes its local Rendezvous (RV) to maintain a match Pub/Sub relation for the mentioned scope and acknowledges receiving the PubiSub message by sending an ACK message back to NAP C. It then locally looks up the FID needed to reach NAP C and uses it to start publishing data to the identified subscriber.

After the above steps, MN A and the CN can start sending/receiving data payload messages of size ζ.

1) Mobility Signaling Cost: The mobility signaling cost Υ' is the product of the signaling messages size in *bytes* and the hop count it traverses. Hence, the signaling overhead introduced to support seamless handover in IP-over-ICN is calculated as follows:

$$\Upsilon' = \big\{h_{a,b}(|\ell_s| + ACK) + h_{c,b}(|\ell_i| + ACK) + h_{b,j}|\ell_u| + \sum_{n\in\mathcal{N}_c} h_{j,n}|\ell_u|\big\}, \tag{21}$$

where $h_{a,b}$ represents the hop count between the previous NAP A (the serving NAP before handover) and the corresponding NAP B, $h_{c,b}$ represents the hop count between the next NAP C (the serving NAP after handover) and the corresponding NAP B, $h_{b,j}$ represents the hop count between the corresponding NAP B and the multicast route fan out node (i.e., the node that branches out to reach all members of the multicast group) represented as node j. $\mathcal{N}_c$ is the set of neighboring NAPs where handover will happen. $\mathcal{N}_c$ set is found by determining the $|\mathcal{N}_c|$ maximum values of $\phi_{i,l}(t)$ ($\Phi'_{h,i,l}(t)$) for 1st order (2nd order) Semi-Markov chain prediction. Finally, $h_{j,n}$ represents the hop count between the multicast route fan-out node j and neighboring NAP n.

2) Mobility Packet Delivery Cost: The packet delivery cost Λ' refers to the packet delivery overhead of the RLC coded stream that facilitates seamless mobility in IP-over-ICN. It is calculated by multiplying the average packet arrival rate in *packets/sec*, the packet size in *bytes*, and the hop distance, and is expressed as follows:

$$\Lambda' = R^{'} O^{'}, \tag{22}$$

where $O^{'}$ represents the direct path overhead for a RLC coded packet, and $R^{'}$ represents the average packet arrival rate of RLC coded packets during handover. The latter is calculated as $R^{'} = R(1+\epsilon)$, where ϵ represents the coding overhead which is equal to $\frac{K}{N} - 1$ for $K > N$ with K denoting the number of received packets and N the number of source packets. Due to the random structure of the employed RLC codes, each packet is also equipped with a header containing the coding coefficients that describe the coding operations employed to generate the coded packet. The header size varies based on the Galois field used for the operations and the number of source packets. The Galois field $GF(q)$ is equal to $K \log_2 q$, where q is the Galois field size. The coding header can be compressed

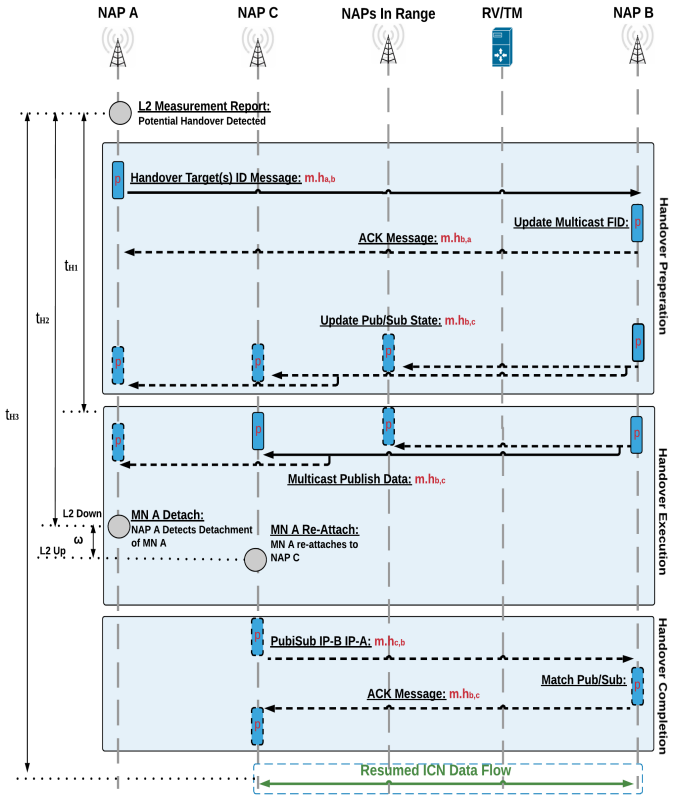

Fig. 4: IP-over-ICN Handover Time Diagram

in only two bytes by following an approach similar to that used in Raptor codes [47], where only the seed of the pseudorandom generator used to produce the code is sent instead of sending the coding coefficients, or by using the approach in [48].

$O^{'}$ in equation (22) can be obtained as follows:

$$O^{'} = h_{b,j}(\varphi' + \zeta) + \sum_{n\in\mathcal{N}_c} h_{j,n}(\varphi' + \zeta), \tag{23}$$

where $h_{b,j}$ represents the hop count between the corresponding NAP B and the multicast route fan out node j, $h_{j,n}$ represents the hop count between the multicast route fan-out node j and neighboring NAP n, and φ' denotes the size of the ICN payload packet header.

According to [33], the average number of RLC packets K that need to be sent in order to recover S source packets can be calculated as:

$$K = \sum_{k=S}^{\infty} k \cdot P_d(k, S), \tag{24}$$

where P_d represents the probability of the receiver having received S linearly independent packets out of the K transmitted packets. This probability can be calculated using the following formula:

$$P_d(K,S) = \begin{cases} 0 & , \text{if } K < S \\ \prod_{j=0}^{S-1} 1 - \frac{1}{q^{K-j}} & , \text{if } K \geq S \end{cases}, \tag{25}$$

3) Handover Latency Cost: Fig. 4 shows the timing diagram for seamless handover in IP-over-ICN where Γ' is the time duration for a Layer 2 handover, which is the time from

when the MN sends an L2 report message to the previous NAP to when the MN's L2 association with the next NAP is completed. Γ' can be expressed as:

$$\Gamma' = t_{H2} + \omega, \tag{26}$$

where t_{H2} represents the time between delivery of L2 report message and L2 link down event and ω indicates the time between L2 link down event at previous NAP and L2 link up event at next NAP. We also define γ' to be the time duration for handover mode transmission, which is the time from when the MN starts receiving RLC coded traffic from the previous NAP, to when normal IP-over-ICN traffic is resumed at the next NAP. In other words, it is the duration of RLC coded transmission throughout the network with regards to a single handover. γ' is given by:

$$\gamma' = \Gamma' - t_{H1}, \tag{27}$$

where t_{H1} represents the time duration of pre-handover preparation. Therefore, if $t_{H2} > t_{H1}$, the MN will start receiving RLC coded traffic prior to handover from the previous NAP. While if $t_{H1} > t_{H2}$, the MN will only receive coded traffic after the handover from the next NAP.

As in PFMIPv6, latency is analyzed in terms of number of exchanged messages, required processes and traversed hops to complete a successful seamless handover. However, as explained in Section III-C4, the prediction model used to identify target handover candidates takes an offline approach, which allow us to ignore the prediction processing time. According to the timing diagrams in Fig. 4, p denotes the time to process a message, m denotes the time to exchange a message and h denotes the hop count that a message traverses. Again, as in PFMIPv6 we assume that p and m are expressed in arbitrary time units, and both set to 1 time unit. This implies that the link transmission delay is comparable to the forwarding delay. Therefore, for IP-over-ICN, the handover latency cost T_c' can be computed as:

$$T_c' = 5p + m \cdot h_{a,b} + m \cdot h_{b,c} \tag{28}$$

The messages and processes in dotted line in Fig. 4 have not been included in equation (28) for the following reasons. The ACK message sent from the corresponding NAP B towards the previous NAP A does not affect MN A's detachment from NAP A or any of the other subsequent handover preparation steps. Its absence can only lead to retransmission of the Handover Target(s) ID Message. Also, the Update Pub/Sub State message sent from the corresponding NAP B to the neighboring NAPs does not affect the subsequent handover execution phase, i.e., MN receiving RLC coded handover traffic. In fact, when this message does not arrive, it can only lead to a hard rather than soft handover completion phase. The latency cost in equation 28 does not include the PubiSub message sent to trigger the handover completion phase, since the message also carries the MN's initial data payload, and therefore does not introduce any additional latency. Finally, it is worth noting that in case of prediction failure (i.e., the prediction model fails to identify the correct handover target), a break-before-make handover is performed. A detailed cost analysis of break-before-make handover for IP-over-ICN can be found in [24].

TABLE I: List of mobility messages and their sizes

Notation	Description	Size
PBU	Proxy binding update	76 Bytes [46]
PBA	Proxy binding acknowledgement	76 Bytes [46]
H_r	Handover initiation request	104 Bytes [22]
H_a	Handover acknowledgement	168 Bytes [22]
φ	Proxy MIPv6 tunnelling header	40 Bytes [46]
ζ	Average payload length	1024 Bytes
ℓ_u	Multicasted state update message	102 Bytes
ℓ_s	Handover target(s) ID message	160 Bytes
ℓ_i	Publish with implicit Subscription message (PubiSub)	166 Bytes
φ'	ICN payload packet header	96 Bytes
ACK	ICN Acknowledgement Message	64 Bytes

V. Simulation and Performance Evaluation

IP-over-ICN handover performance, and that of its counterpart (FPMIPv6) is evaluated using a discrete time event simulation that was built in R specifically for this purpose. The simulation was conducted on a PC with Intel Core i7 CPU at 2.3 GHz and 16 GB Memory. The simulation environment incorporates a realistic mobility dataset representing vehicular movement in the German city of Cologne. The city's actual cellular infrastructure, comprising of 247 base stations (eNodeBs) [15], in addition to 25 core forwarding nodes/switches have been deployed in the simulation. To approximate the coverage area of individual base stations in the region, we perform a Voronoi tessellation on the base station locations. The Open Street Map (OSM) database is used to extract the city's road topology map, and the Simulation of Urban Mobility (SUMO) software is employed to simulate the microscopic mobility of vehicles. To derive the traffic demand information on the macroscopic traffic flows across Cologne, we employ the Travel and Activity Patterns Simulation (TAPAS) methodology. The mobility dataset generated, resembles the vehicular mobility of Cologne for a 24 hour period, with more than $700,000$ individual trips in total [49].

To represent both the LMA and TM/RV in the core network, the same central node was utilized to ensure valid cost comparisons. Our data traffic model assumes that all network users transmit video data at a rate of 1 Mbps, following a Poisson distribution for packet arrival. It is also presumed that handover latency is between $50ms$ [50] and $2sec$ [22] following a conditional uniform distribution depending on the MN velocity; where MNs with higher velocity experience lower handover latency and vice versa. This is due to the fact that faster MNs spend less time in overlapping coverage areas, hence handover decisions can be made more rapidly [22].

We restrict all coding operations to be performed in $GF(2^4)$. We assume an RLC generation of 100 packets and an RLC coding overhead ϵ equal to 0.02, which means that two extra RLC packets are transmitted. These parameters setting ensure a decoding probability $P_d(K, S)$ of 99.9%, as shown in [33] while in parallel maintain the coding complexity low.

The mobility cost equations derived in section IV have been used in the simulation to evaluate the signalling, packet delivery, and latency costs for both IP-over-ICN and its

FPMIPv6 counterpart. Table I presents a summary of the mobility messages, along with their corresponding sizes, for both evaluated solutions. For ℓ_u, ℓ_s, ℓ_i and φ', these assume ICN FIDs and name ID lengths of 256 bits each and a single scope and ID for the IP naming.

A. Prediction Performance Evaluation with Respect to Number of Residence Time Clusters

In Fig. 5, we compare the prediction success rates of target handover cells for 1st and 2nd order Markov chains according to the number of residence time clusters used. The simulation was run 10 times, each for an hour slot covering the vehicular mobility in the city of Cologne for the hours between 8:00 and 18:00. The figure illustrates that the number of residence time clusters used has a significant impact on the prediction performance, where 1st order Markov chain prediction success rate increases from 73% when 2 residence time clusters are used, up to 84% when the number of residence time clusters increases to 10. On the other hand, expanding the prediction targets to two cells per handover for every user increases the hit rate for 1st order Markov chain prediction to about 97% for 10 residence time clusters. It can also be seen from Fig. 5 that 2nd order Markov chain prediction substantially increases the prediction success rate that reaches about 98% for 10 residence time clusters. This outperforms 1st order Markov chain prediction even when 2 target handover cells are predicted. This is due to the predictable nature of the vehicular mobility that is governed by road pathways, which makes very useful to know the previous cell that the user was attached to before moving to the current cell, in order to make a more accurate prediction of the next cell transition.

Based on the above, 10 residence time clusters have been considered the benchmark for the prediction technique proposed, as it is obvious from Fig. 5 that further increasing the number of residence time clusters only has marginal effects on the prediction accuracy. Finally, it is worth to note that 25% of the mobility data set was used for training the prediction model and the remaining 75% for evaluation. The split was based on thorough testing, where it was found that according to the characteristics of the data set, 25% is sufficient to achieve high prediction accuracy, while avoiding over fitting at the same time.

B. Prediction Performance Evaluation with Respect to Time of Day

Fig. 6 shows the prediction success rates between 8:00 and 18:00 with 10 residence time clusters. From this figure, it is clear that in general, the prediction accuracy is higher during peak hours from 8:00 to 10:00 and 16:00 to 18:00, than the rest of the day. This is due to people following similar mobility behavior during peak hours where destination convergence is higher. For example, people usually move toward the city center during morning hours where the majority of businesses, offices, universities, etc. are located. Whereas the opposite happens during evening hours where people usually travel home towards the city suburbs. For all other time periods, vehicular mobility usually has a scattered pattern. Another interesting conclusion from Fig. 6 is that 2nd order Markov

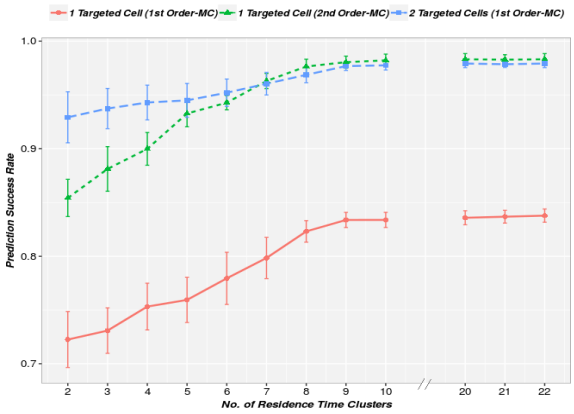

Fig. 5: Prediction success rates according to number of residence time clusters for 1st and 2nd order Markov chains.

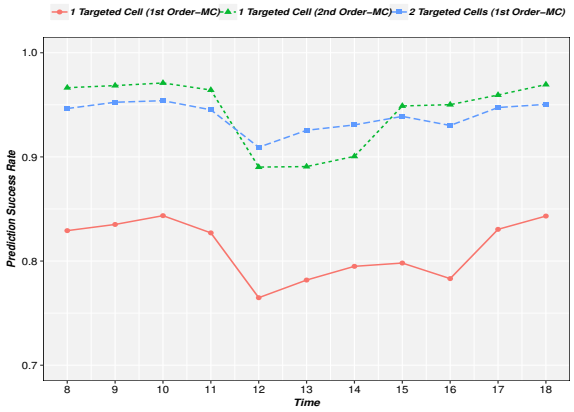

Fig. 6: Prediction success rates at different hours of the day with 10 residence time clusters.

chains are more affected by mobility patterns and therefore hours of the day than 1st order Markov chains. This is because 2nd order Markov chains take the previous cell attachment into account when calculating the prediction probabilities, and therefore the prediction success rate is more affected by the actual mobility patterns of the vehicular users.

C. Packet Delivery and Signaling Cost Evaluation for IP-over-ICN vs. PFMIPv6 Networks

Having explored the prediction performance, we now compare IP-over-ICN and PFMIPv6 in terms of packet delivery and signalling cost. Figs. 7, 8 and 9 depict the findings of a simulation run of $1800sec$ from 9:00 to 9:30 for the Cologne metropolitan region in Germany for both PFMIPv6 and IP-over-ICN. Fig. 7 shows the average and total Packet Delivery Cost (PDC) for PFMIPv6 and IP-over-ICN (with 1 target cell 2nd order Markov chain prediction in the IP-over-ICN case) to support seamless handover. We observe from the figure that PFMIPv6 incurs a higher total PDC of approximately 13×10^9 *Bytes.Hops* to support seamless handover, compared to 7.3×10^9 *Bytes.Hops* for IP-over-ICN. In other words, PFMIPv6 shows approximately 1.8 times the total PDC costs imposed by IP-over-ICN due to the central traffic anchoring used in PFMIPv6. Fig. 8 shows the average and total handover mode PDC for PFMIPv6 and IP-over-ICN

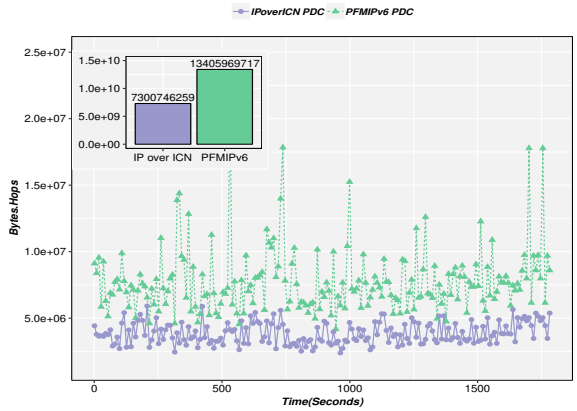

Fig. 7: PDC for PFMIPv6 vs. IP-over-ICN with 1 target cell prediction.

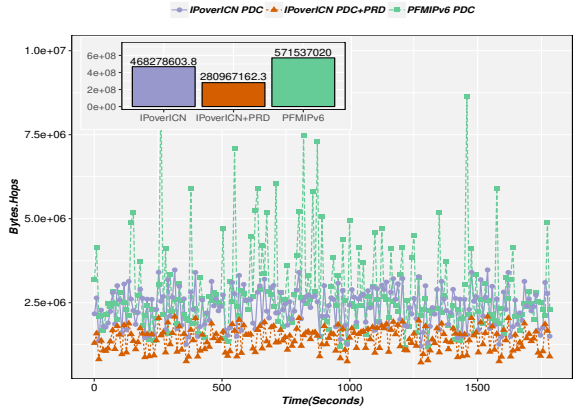

Fig. 8: Handover PDC for PFMIPv6 vs. IP-over-ICN with and without prediction.

(with and without prediction in the IP-over-ICN case). This represents the PDC incurred only during handover (i.e., for the MNs that are in handover mode), and not for the whole period of communication for all MNs which was the case in Fig. 7. We can observe from the figure that PFMIPv6 incurs 1.2 of the total PDC costs imposed by IP-over-ICN even without prediction, i.e., distributing handover traffic to all neighboring cells in IP-over-ICN. In the IP-over-ICN case, the difference in total handover PDC between prediction and no prediction mode is about 40%. This is due to the decrease in handover prepared targets from four in average to one target only. However, even preparing all handover target cells does not have a big impact on the air interface as it has been verified through the conducted simulations that a user, only spends 2% of the total connection time on average, handing over between cells.

Fig. 9 shows the average and total signaling cost (SC) for both IP-over-ICN (with no prediction) and PFMIPv6 to support seamless handover. It is worth to note that in the IP-over-ICN case, signaling costs are higher with no prediction due to the higher number of target cells taking part in the handover preparation. The figure clearly shows that IP-over-ICN incurs a higher SC of approximately 7.9×10^5 *Bytes.Hops* compared to 5.2×10^5 *Bytes.Hops* for PFMIPv6. The reason for this is that IP-over-ICN relies on source routing, which results in a higher number of signalling messages to convey delivery path information to the traffic source during mobility.

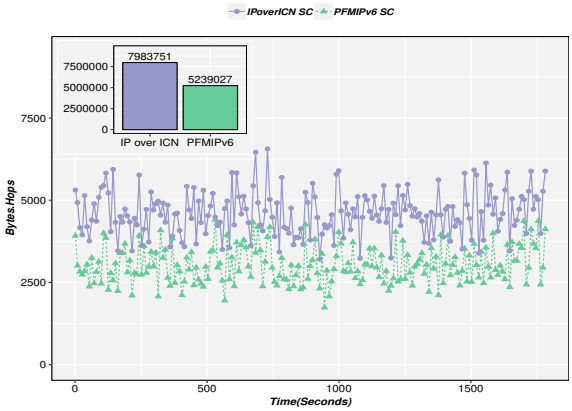

Fig. 9: SC for seamless handover in PFMIPv6 vs. IP-over-ICN.

In terms of the overall cost, the proposed scheme shows significantly better performance than PFMIPv6, as PDC is the dominant cost. Therefore, the figures clearly show that using an ICN core and RLC coding to facilitate IP mobility with better QoS (resilience against data loss and minimum HO failure) can be achieved with lower costs than PFMIPv6 on average and in total.

D. Handover Latency Cost Evaluation for IP-over-ICN vs. PFMIPv6 Networks

This simulation experiment focuses on comparing the handover latency of IP-over-ICN and PFMIPv6 schemes. Fig. 10 displays an Empirical Cumulative Distribution Function (ECDF) of the handover latency in both domains. The graph shows that IP-over-ICN outperforms PFMIPv6 in terms of handover latency, where in 90% of handovers, IP-over-ICN incurs a handover latency cost of less than 28 units time compared to 38 units time for PFMIPv6. This is because although IP-over-ICN imposes a higher number of signalling messages and processes in total due to its source routing approach, not all of these messages and processes directly affect the handover operation (and hence the incurred latency) as compared to PFMIPv6 and outlined in Section IV-B3. This clearly illustrates the efficient design of the proposed control plane signalling operations to facilitate seamless handover offering about 26% lower handover latency cost with significant savings on the data plane traffic as shown in the previous results.

VI. Conclusions

This paper has shown that RLC can be applied to seamless IP handover with minimum disruption if a novel IP-over-ICN approach is taken. The evaluation findings indicate that seamless handover can be facilitated with lower traffic and latency costs than an existing state-of-the-art solution like PFMIPv6, thus providing scalable and sustainable mobility support. The improvement is due to two main factors: RLC coding and using IP-over-ICN. The RLC eliminates the need for resuming packet transmissions when a handover occurs and obviates the need for protocol packet synchronization techniques such as acknowledgements. By utilizing ICN in the core network, delivery via the shortest path is ensured allowing sustainable and resilient network operation as opposed to the

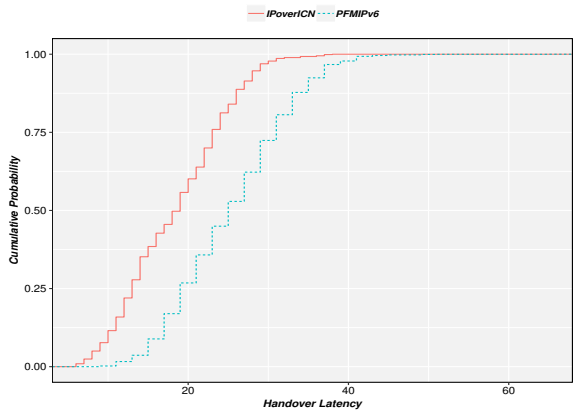

Fig. 10: ECDF of handover latency costs in an IP-over-ICN vs. PFMIPv6 cellular network.

highly sub-optimal routing caused by traffic anchoring and tunneling required by existing IP mobility solutions including PFMIPv6. Moreover, the results show that Markov chain prediction techniques provide very high success rates of up to 98% in predicting handover target cells in realistic mobility scenarios. Thus, the handover preparation cost is significantly reduced and the QoS perceived by the users is maintained.

Acknowledgment

This work was carried out within the project POINT, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 643990.

References

[1] "Ericsson Mobility Report," Jun 2023. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report/reports/june-2023

[2] Y. Zhu, Z. Zhang, Z. Marzi, C. Nelson, U. Madhow, B. Y. Zhao, and H. Zheng, "Demystifying 60GHz Outdoor Picocells," in *Proc. of Int. conf. on Mobile computing and networking, MobiCom'14*, Maui, Hawaii, USA, Sep. 2014.

[3] S. M. A. Zaidi, M. Manalastas, H. Farooq, and A. Imran, "Mobility management in emerging ultra-dense cellular networks: A survey, outlook, and future research directions," *IEEE Access*, vol. 8, pp. 183 505–183 533, 2020.

[4] D. Trossen, M. J. Reed, J. Riihijärvi, M. Georgiades, N. Fotiou, and G. Xylomenos, "IP over ICN - The better IP?" in *Proc. of EuCNC'15*, Oulu, Finland, June 2015.

[5] D. Trossen and G. Parisis, "Designing and realizing an information-centric internet," *IEEE Comm. Magazine*, vol. 50, no. 7, pp. 60–67, Jul. 2012.

[6] R. Al-Zaidi, J. C. Woods, M. Al-Khalidi, and H. Hu, "Building novel vhf-based wireless sensor networks for the internet of marine things," *IEEE Sensors Journal*, vol. 18, no. 5, pp. 2131–2144, 2018.

[7] J. Saltarin, E. Bourtsoulatze, N. Thomos, and T. Braun, "NetCodCCN: a Network Coding Approach for Content-Centric Networks," in *IEEE INFOCOM'16*, San Francisco, CA, USA, Apr. 2016.

[8] E. Bourtsoulatze, J. Saltarin, N. Thomos, and T. Braun, "Content-Aware Delivery of Scalable Video in Network Coding Enabled Named Data Networks," *IEEE Trans. on Multimedia*, vol. 20, no. 6, pp. 1561–1575, Jun. 2018.

[9] C. Anastasiades, N. Thomos, A. Striffeler, and T. Braun, "RC-NDN: Raptor Codes Enabled Named Data Networking," in *IEEE Int. Conf. on Communications, ICC'15*, London, UK, Jun. 2015, pp. 3026–3032.

[10] H. Malik, C. Adjih, C. Weidmann, and M. Kieffer, "MICN: A network coding protocol for ICN with multiple distinct interests per generation," *Computer Networks*, vol. 187, Mar. 2021.

[11] J. W. Byers and M. Luby, "Liquid Data Networking," in *Proc. of ACM conf on Information-Centric Networking*, Paris, France, Sep. 2020, pp. 129–135.

[12] H. Abu-Ghazaleh and A. S. Alfa, "Application of Mobility Prediction in Wireless Networks Using Markov Renewal Theory," *IEEE Trans. Vehicular Technology*, vol. 59, no. 2, pp. 788–802, Feb. 2010.

[13] F. Piccialli, S. Cuomo, F. Giampaolo, G. Casolla, and V. S. di Cola, "Path prediction in iot systems through markov chain algorithm," *Future Generation Computer Systems*, vol. 109, pp. 210–217, 2020.

[14] M. Al-Khalidi, N. Thomos, M. J. Reed, M. F. Al-Naday, and D. Trossen, "Seamless handover in IP over ICN networks: A coding approach," in *in Proc. of IEEE Int. Conf. on Communications, ICC'17*, Paris, France, May 2017.

[15] S. Uppoor, D. Naboulsi, and M. Fiore, "Vehicular mobility trace of the city of cologne, germany," *URL: http://kolntrace. project. citi-lab. fr*.

[16] N. Zohar, "Beyond 5g: Reducing the handover rate for high mobility communications," *Journal of Communications and Networks*, vol. 24, no. 2, pp. 154–165, 2022.

[17] D. Pineda, R. Harrilal-Parchment, K. Akkaya, and A. Perez-Pons, "Sdn-based gtp-u traffic analysis for 5g networks," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2023, pp. 1–4.

[18] R. S. Shetty, "5g overview," in *5G Mobile Core Network*. Springer, 2021, pp. 1–67.

[19] M. Al-Khalidi, R. Al-Zaidi, and M. Hammoudeh, "Network mobility management challenges, directions, and solutions: An architectural perspective," *Electronics*, vol. 11, no. 17, p. 2696, 2022.

[20] I. Shayea, M. Ergen, M. H. Azmi, S. A. Çolak, R. Nordin, and Y. I. Daradkeh, "Key challenges, drivers and solutions for mobility management in 5g networks: A survey," *IEEE Access*, vol. 8, pp. 172 534–172 552, 2020.

[21] M. Al-Khalidi, R. Al-Zaidi, A. M. Abubahia, H. M. Pandey, M. I. Biswas, and M. Hammoudeh, "Global iot mobility: A path based forwarding approach," *Journal of Sensor and Actuator Networks*, vol. 11, no. 3, p. 41, 2022.

[22] Kim, Mun-Suk and Lee, Sukyoung and Cypher, David and Golmie, Nada, "Performance analysis of fast handover for Proxy Mobile IPv6," *Information Sciences, Elsevier*, vol. 219, pp. 208–224, 2013.

[23] M. Balfaqih, Z. Balfaqih, V. Shepelev, S. A. Alharbi, and W. A. Jabbar, "An analytical framework for distributed and centralized mobility management protocols," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 7, pp. 3393–3405, 2022.

[24] M. Al-Khalidi, N. Thomos, M. J. Reed, M. F. AL-Naday, and D. Trossen, "Anchor Free IP Mobility," *IEEE Trans. on Mobile Computing*, vol. 18, no. 1, pp. 56–69, Jan 2019.

[25] Y. Zeng, B. Ye, B. Tang, S. Lu, F. Xu, S. Guo, and Z. Qu, "Mobility-aware proactive flow setup in software-defined mobile edge networks," *IEEE Transactions on Communications*, vol. 71, no. 3, pp. 1549–1563, 2023.

[26] I. V. S. Brito and G. B. Figueiredo, "Improving QoS and QoE through seamless handoff in software-defined IEEE 802.11 mesh networks," *IEEE Comm. Letters*, vol. 99, no. 1, pp. 2484–2487, Jan. 2017.

[27] T. Bilen, B. Canberk, and K. R. Chowdhury, "Handover Management in Software-Defined Ultra-Dense 5G Networks," *IEEE Network*, vol. 31, no. 4, pp. 49–55, Apr. 2017.

[28] A. Abdulghaffar, A. Mahmoud, M. Abu-Amara, and T. Sheltami, "Modeling and evaluation of software defined networking based 5g core network architecture," *IEEE Access*, 2021.

[29] A. Alhammadi, W. H. Hassan, A. A. El-Saleh, I. Shayea, H. Mohamad, and W. K. Saad, "Intelligent coordinated self-optimizing handover scheme for 4g/5g heterogeneous networks," *ICT Express*, vol. 9, no. 2, pp. 276–281, 2023.

[30] A. K. Yadav, K. Singh, P. K. Srivastava, and P. S. Pandey, "I-merec-t: Improved merec-topsis scheme for optimal network selection in 5g heterogeneous network for iot," *Internet of Things*, vol. 22, p. 100748, 2023.

[31] R. Bassoli, H. Marques, J. Rodriguez, K. W. Shum, and R. Tafazolli, "Network coding theory: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1950–1978, 2013.

[32] N. Thomos, E. Kurdoglu, P. Frossard, and M. van der Schaar, "Adaptive prioritized random linear coding and scheduling for layered data delivery from multiple servers," *IEEE Trans. on Multimedia*, vol. 17, no. 6, pp. 893–906, June 2015.

[33] O. Trullols-Cruces, J. M. Barcelo-Ordinas, and M. Fiore, "Exact Decoding Probability Under Random Linear Network Coding," *IEEE Comm. Letters*, vol. 15, no. 1, pp. 67–69, Jan. 2011.

[34] M. Gonen, M. Langberg, and A. Sprintson, "Latency and alphabet size in the context of multicast network coding," *IEEE Transactions on Information Theory*, vol. 68, no. 7, pp. 4289–4300, 2022.

[35] X. Liao, J. Yin, M. Chen, and Z. Qin, "Adaptive payload distribution in multiple images steganography based on image texture features," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 897–911, 2020.

[36] M. V. Pedersen, J. Heide, P. Vingelmann, and F. H. Fitzek, "Network coding over the 2 32- 5 prime field," in *IEEE International Conference on Communications (ICC)*. IEEE, 2013, pp. 2922–2927.

[37] J. Heide, M. V. Pedersen, F. H. Fitzek, and M. Médard, "On code parameters and coding vector representation for practical RLNC," in *IEEE International Conference on Communications (ICC)*. IEEE, 2011, pp. 1–5.

[38] M. Nistor, D. E. Lucani, T. T. Vinhoza, R. A. Costa, and J. Barros, "On the delay distribution of random linear network coding," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 5, pp. 1084–1093, 2011.

[39] M. J. Reed, M. Al-Naday, N. Thomos, D. Trossen, G. Petropoulos, and S. Spirou, "Stateless multicast switching in software defined networks," in *proc. of IEEE Int. Conf. on Communications, ICC'16*, Kuala Lumpur, Malaysia, May, 2016.

[40] J.-K. Lee and J. C. Hou, "Modeling steady-state and transient behaviors of user mobility: formulation, analysis, and application," in *Proc. of the ACM Int. Symp. on Mobile ad hoc networking and computing, MobiHoc'06*, Florence, Italy, May 2006.

[41] A. Nadembega, A. Hafid, and T. Taleb, "A Destination and Mobility Path Prediction Scheme for Mobile Networks," *IEEE Trans. on Vehicular Technology*, vol. 64, no. 6, pp. 2577–2590, Jun. 2015.

[42] R. Choquet, A. Béchet, and Y. Guédon, "Applications of hidden hybrid Markov/semi-Markov models: from stopover duration to breeding success dynamics," *Ecology and Evolution*, vol. 4, no. 6, 2014.

[43] M. J. De Smith, M. F. Goodchild, and P. Longley, *Geospatial analysis: a comprehensive guide to principles, techniques and software tools*. Troubador Publishing Ltd, 2007.

[44] G. F. Jenks, "The data model concept in statistical mapping," *International yearbook of cartography*, vol. 7, pp. 186–190, 1967.

[45] H. Yokota, K. Chowdhury, R. Koodli, B. Patil, and F. Xia, "Fast handovers for proxy mobile IPv6," IETF, RFC 5949, 2010.

[46] J.-H. Lee, T. Ernst, and T.-M. Chung, "Cost Analysis of IP Mobility Management Protocols for Consumer Mobile Devices," *IEEE Trans. on Consumer Electronics*, vol. 56, no. 2, pp. 1010–1017, Feb. 2010.

[47] A. Shokrollahi, "Raptor Codes," *IEEE Trans. on Information Theory*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.

[48] N. Thomos and P. Frossard, "Toward one Symbol Network Coding Vectors," *IEEE Comm. letters*, vol. 16, no. 11, pp. 1860–1863, Nov. 2012.

[49] S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Generation and analysis of a large-scale urban vehicular mobility dataset," *IEEE Trans. on Mobile Computing*, vol. 13, no. 5, pp. 1061–1075, May 2014.

[50] 3GPP TR 36.881 V 14.0.0, "Study on Latency Reduction Techniques for LTE," Tech. Rep., 2016.