Studies on Metric Spaces and Place Specification for Named Node Network

Named Node Networkの位置空間と場所規定に関する研究

July 2019

Jairo Eduardo　LOPEZ FUENTES NACARINO
ロペス　フェンテス　ナカリノ　ハイロ　エドアルド

Studies on Metric Spaces and Place Specification for Named Node Network

Named Node Networkの位置空間と場所規定に関する研究

July 2019

Waseda University
Graduate School of Fundamental Science and Engineering
Department of Computer Science and Communications Engineering, Research on Ubiquitous Communication System

Jairo Eduardo LOPEZ FUENTES NACARINO
ロペス　フェンテス　ナカリノ　ハイロ　エドアルド

To my wife Yanwei, our son Ian and Ian's grandparents

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Predictions for 2021 say that mobile data traffic will increase sevenfold, growing twice as fast as wired traffic. This will see mobile data traffic represent 17% of the total network traffic by 2021 [52]. The continued increase in mobile data traffic makes it imperative that the deployed network architectures implement seamless mobility support.

Mobility in networks has been a fundamental part of network research. Although supporting mobility schemes seems intuitively easy, implementing practical solutions that fit within a given network can be quite complex. For the early IPv4 architecture, one of many solutions can be seen in [48]. This particular solution is interesting as it touches on the issue of addressing in networks in a period when actual mobile terminals were uncommon. In 1982, Jerome H. Saltzer emphasized that correctly defining and naming network objects could enable mobility [45]. In the paper, the author proposed the naming of four network objects. These objects where services, nodes, attachment points and paths. These names can be leveraged through bindings that permit one to go from a service to a path, with intermediate steps including a node and a point of attachment. A key point in Saltzer's description is that a service may run simultaneously on many nodes and may need to *migrate* to another. In that migration, it may be necessary for the service to not lose its identity. A similar logic is applied to nodes. According to the paper, nodes might have connections to multiple network attachment points and may also need to *migrate* to another network attachment point. In that migration, the node may also need to preserve its identity as a node. Finally the paper mentioned that a pair of attachment points may be linked by multiple paths. Similarly, these paths may need be changed without consequences for the identity of the attachment points.

Saltzer's paper was considered a fundamental work for naming network objects and in 1993 was introduced as RFC 1498 by the Internet Engineering Task Force (IETF) to the Internet Society (ISOC) [46]. The definitions used in this work have maintained their relevance in computer network research, however it is clear that the concept has not reached a commonly converged implementation. The IETF in [58], 18 years after the publication of Saltzer's work, lists 18 different implementations for mobility enhancement in IP networks. In [16], a summary of the Locator/Identifier class of mobility solutions for IP networks is shown, with 3 different implementations for mobility enhancement being introduced. In the paper, the authors recognize that part of the issue related to IP network and mobility is the overloading of IP address semantics. In other words it is an issue that is directly tied to network naming and addressing.

One possible reason that IP networks have had such a complicated history with mobility may be due to engineering trade-offs that were taken, with the best of intentions in mind, but that have become a stumbling block for implementing complete solutions. One key piece of evidence is the migration from IPv4 addresses using a classful network design to using Classless Inter-Domain Routing (CIDR). Although CIDR was successfully integrated to IPv4 networks, it only delayed the urgency of problems related to routing state and address depletion. The IETF in [17] states that CIDR, although extremely successful, is considered a short-term solution because the CIDR integration does not modify the Internet routing or addressing fundamentals. This implies that the issues that necessitated CIDR have not been solved. It is also not very hard to imagine that a complete solution may not have appeared specifically because the routing and naming architecture of IP networks has not been modified.

As the Internet, and by extension the IPv4 and IPv6 network architecture, has expanded and is being used world-wide for all manner of academic, government, military and social purposes, network reliability has become more important. Under these conditions, making fundamental changes to the network architecture, such as modifying the network naming schemes, becomes justifiably unacceptable without a considerable amount of evidence and a very strict and clear road-map for roll-out. For such cases, a very different approach becomes necessary. One approach is the development of a clean slate network architecture. A clean slate network architecture is a restructuring from the ground up of the network, its objects and components. Such an attempt can offer the flexibility to modify fundamental aspects of a network architecture and test them without affecting the usage of current networks. An attempt of this type can provide the technology or the insight that could at long last completely solve the issues one is seeing in the initial network due to not being constrained to having backward compatibility [42].

Designing a network architecture is not an easy task. Apart from the numerous technical challenges, attention is increasingly placed on the economic, social and cultural challenges that can derail or promote any attempt. Fortunately, in recent years researchers, economists and historians have started publishing accounts and promoting newer concepts for building networks that include the history of IP network architectures such as [19] and a more general overview of what one key IP network architecture researcher believes network architecture design should be [8], the history of other computer network implementations, like the French Minitel [32] and the Soviet attempt to make a nation-wide network [40], as well as complete rethinking of computer network architectures such as in [10]. This is added to the enormous number of software developers and usable code that is available due to the effort of advocates of Free and Open Source Software (FOSS) movements [51]. It is our belief that under the current circumstances, with all the resources available, researching network naming schemes, particularly for mobility could benefit from a clean slate network architecture approach.

The list for clean slate network architecture has been growing since at least the beginning of the year 2000 [39], permitting researchers to attempt a lot of ideas that were unfeasible on the IP network architecture. Among the clean slate network architectures, there is one architecture above others that has captivated the imagination of a generation of researchers. Starting with the name Content-Centric Networking (CCN) [23], the idea of making network data the most important object was met with a lot of interest. The resulting body of research has been collected under the name Information-Centric Networking (ICN) and has produced a great number of implementations that have taken the initial idea down very different paths. The basic idea was that by naming only the data a lot of IP network architecture issues would be greatly decreased, among them mobility related issues. The promise of completely solving mobility related issues seems to not have been solved the way everyone expected as can be seen in survey [15], where mobility issues have been conveniently renamed "Consistent Routing".

In this environment, it is worth researching network naming. Due to there being a large number of unsuccessful mobility enhancement network additions in the known literature, it also seems equally prudent to take a look at how mobility is defined and how mobility related information is transmitted from a much wider perspective. One area that is highly overlooked is how mobility information is transmitted in language. Due to mobility being directly related to localization, it is important to first present the current understanding of the codification of place specification in languages. With a deeper understanding of how mobility, location and network naming are related, one can leverage from the available network architecture implementations to reevaluate our understanding of these topics.

Within the ICN implementations, the Named Data Networking (NDN) implementation took an interesting step when it came to naming and Protocol Data Unit (PDU) forwarding by utilizing names from an unmanaged, unbounded namespace for only naming data. In addition to this single endless namespace, its core paper proclaims the advantages of such a scheme, as it supposedly completely does away with addresses [57]. The NDN implementation has two simulators versions detailed in [2] and [33] as well as a fully working GNU/Linux compatible stack implementation detailed in [11]. All this software is licensed under the GPLv3 [1], permitting access to the inner workings of the implementations.

The liberty to freely use the single NDN namespace for all imaginable purposes and to be able to change the NDN code base, provides an amazing opportunity to test network naming models in an attempt to create a network architecture that has inherent mobility support.

1.2 Justification

Mobility has become one of the most fundamental part of networking. As of March 2019, the total number of mobile broadband subscriptions has surpassed 5.7 billion. It is predicted that by the end of 2024, close to 95% of all subscriptions will be for mobile broadband with practically all subscriptions being for smartphones for a total of 8.4 billion mobile broadband subscriptions [14]. With these estimates, it becomes absolutely necessary for our networks to be able to support this impressive growth of mobile connections along with the known explosion of network flows that come from data enabled devices.

It is known that current IP network architectures have issues with mobility. It is also known that there is no consensus for mobility enhancements in even one of the most developed and researched clean slate network architectures. This knowledge is sufficient to justify us taking a step back to verify if our definitions and our understanding of network mobility, network location and network naming is correct.

1.3 Objectives

Our objective is to understand network naming and how it relates to supporting mobility in network architectures in order to be able to improve network naming. Efforts will be made to create a method to verify if a naming scheme in a network can be improved. Then the ns-3 simulator [43] and the NDN simulator, ndnSIM, that is completely implemented within ns-3 [2], is leveraged to test whether our verification method can improve network mobility support by implementing the necessary structure to successfully pass our verification.

1.4 Document overview

This document explores if there can be a convergence of network naming strategies. Chapter 2 explains the theoretical framework used throughout this document. Section 2.4 utilizes Section 2.1 and Section 2.2 to develop a questionnaire to verify if a network naming scheme can be improved. Within this chapter, in Section 2.5, the basic required knowledge of Information-Centric Networking (ICN) is also introduced. Chapter 3 utilizes all the information from the previous chapter to evaluate the network naming schemes of various network architectures. Chapter 4 utilizes the information from the previous two chapters to construct a new network architecture based on ICN that has an improved network naming structure capable of supporting producer and push producer mobility. Chapter 5 details the framework used to evaluate the constructed network architecture. Chapter 6 describes the results of the constructed network architecture for mobile producer with mobile consumer and mobile push producer with mobile consumer scenarios. Finally chapter 7 will describe the lessons learned researching mobility, location and computer network naming.

Chapter 2

Theoretical Framework

The most important concepts required to explain our design decisions are explained in this chapter. Section 2.1 goes into detail about the importance of place specification in language and describes the linguistic primitives that will be of use throughout this document. Section 2.2 describes in detail the properties of language that enable place specification as well as some of the strategies used. This information will also be crucial when discussing naming and place specification in networks. The definitions for both these sections will be used throughout this document. Section 2.3 will go into detail about the current model used to consider the naming of network components.

The ideas of place specification from Section 2.1 will be tied to Saltzer's model to be able to understand where and how place specification takes place in a network structure in Section 2.4.

Section 2.5 will introduce the common Information-Centric Networking (ICN) structures and PDUs which will be later utilized in the construction of the network architecture with our improved network naming scheme.

2.1 How space is described in language

For any species that has a home base or has a foraging strategy, knowledge and reasoning about the space they live in is extremely important. Due to its importance in a species' survival, researchers have hypothesized that having spatial reasoning should be part of an ancient and modular system supported by a species' central nervous system. In the case of humans, researchers have found some evidence that supports this hypothesis. However, Steven C. Levinson and his colleagues, who are linguists, have found that researched human languages differ much more than expected in how they share and transmit spatial information. They have found that different languages use fundamentally different coordinate systems and utilize different principles for constructing their coordinate systems. Although there is an inherent underlying support in the biological composition of humans to permit spatial cognition, Levinson's work puts forward the idea that spatial cognition is greatly supported by language and how it is defined in the language.

To scientifically and reliably compare space cognition in human languages, Levinson and his collaborators had to create a set of primitives to leverage for the framework to test their theories. The primitives were then leveraged in the description of Levinson's frames of reference. Levinson's intrinsic, relative and absolute frames of reference are the only three universally available frames of reference currently proved to exist in cognition and human language. After the description of the primitives in Section 2.1.1, The frames of reference are described in Section 2.2.2.1, Section 2.2.2.2 and Section 2.2.2.3 along with how these are used in specifying the location of objects.

2.1.1 Primitives

1. **System of labeled angles**: Language specific labeled arcs specified by coordinates around an origin. Some examples are *front*, *left* and *north*.

2. **Coordinates**:

(a) Coordinates are *polar*: On a defined plane, each point is determined from a reference point by a distance and from a reference direction by an angle. A single set of terms may require more than one coordinate system.

(b) An established primary coordinate system C_1 on a origin X_1 can be mapped to another origin X_2, by any combination of the transformations below to obtain a *secondary coordinate system* C_2.

- Translation
- Rotation
- Reflection

3. **Points**:

- F, is a figure with a center point at a volumetric center F_C
- G, is a ground with volumetric center G_C and a surrounding region r
- V, is an observer's viewpoint
- X_1, is the origin of the primary coordinate system, with X_2 being the origin of the secondary coordinate system
- A, is an anchor point to a defined set of coordinates
- L, is a designated landmark

4. **Anchoring system**: Locks the labeled angles in 1 into the coordinate system in 2.

- A, is an anchor point within a G or V. In systems using landmarks $A = L$.
- S, is the slope of a fixed bearing system. The system has infinite parallel lines throughout the environment

2.2 Solutions in language for place specification

All researched languages have been found to have *Where*-questions [**?**], strongly suggesting an inherent notion of place and an inherent requirement to handle place specification.

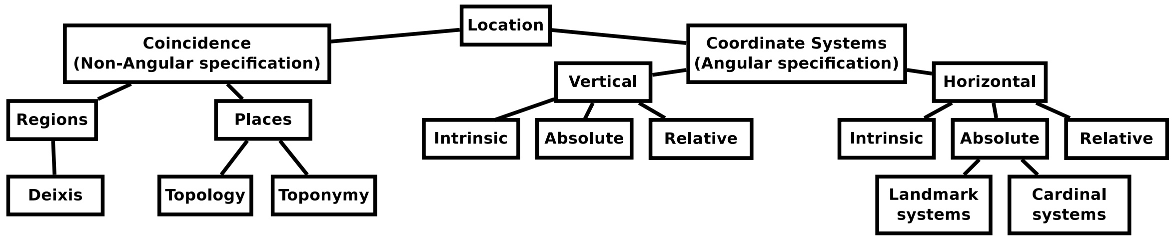

Figure 2.1: Strategies for locating referents for *Where*-questions

Another important uniformity found in languages is the distinction between *figure* (F) and *ground* (G). F being the object to be located and G being the reference domain in which F will be located. In all languages, *Where*-questions tend to evoke answers in which the location of F is specified as in some relation to a G. This G becomes the search domain in where 'near', 'far', 'between' and other similar notions of location are defined. Most languages implicitly use the surface of the Earth as the ground G of the largest scope, even though such a definition is generally unnecessary in daily communication.

Having found these shared traits in languages, Levinson and his collaborators cataloged the distinct strategies used for locating referents. The resulting locative classes are shown in Figure 2.1, and are divided into classes which utilize coordinate systems and those that do not. Section 2.2.1 and Section 2.2.2 focus on describing the locative classes that are relevant to naming structures in a computer network. A complete description of the locative classes can be found in [26, Chapter 3].

2.2.1 Solutions not involving frames of reference

2.2.1.1 Topology

Topology studies geometrical properties that persist in a constant state under transformation or 'deformation'. What is particularly interesting in language is that in [**?**] the authors argue that spatial reasoning stages exist in child development, with children initially only grasping topological notions for spatial relations. The authors list proximity, order, enclosure and continuity as topological primitives. Examples for these spatial relations in English include the use of words such as 'near', 'at', 'between' or 'in'. Much later, children understand Euclidean notions that include distances and angles. It is in the final stage that children are able to grasp projective geometrical notions.

Topological prepositions are available in all frames of reference described in Section 2.1. Due to their availability, these prepositions have a complex relation with the frames of reference. These frames of reference are defined by coordinate systems. On the other hand, topological relators do not express coordinate or angle information. What is fascinating to linguists is that although everything seems to be measurable in our daily perceived space, the encoding of topological information is always available in all languages and that the encoding method can greatly vary from one language to another. The fact that topological information is always available makes it critical to understand topological spaces and topological reasoning, particularly to name and locate things [26, Chapter 3.3.3].

2.2.1.2 Toponymy

This is a simple solution to location specification as one can simply say X is at the place named Y. However, most societies do not employ widespread generative systems for placenames, and no natural languages make systematic use of coordinates. The reasons for this are thought to be varied, but it is generally thought that the number of placenames that would need to be created along with the overhead in learning all the necessary placenames to function within a society discourages the implementation of such a system. Another demerit of a placename heavy system is that these names themselves do not provide a way of *finding* the location. In order to find a placename, the user must maintain a mental map of placename locations that must have been constructed utilizing the user's language [26, Chapter 3.3.1].

2.2.2 Solutions involving frames of reference in the horizontal plane

The main issue that languages have to solve in regard to space cognition is how to define angles on the horizontal plane in order to facilitate the creation of reliable search domains projected off an object. Each of the three identified frames of reference provide a possible answer. The use of one particular frame of reference does not impede the usage of another frame of reference. There are languages and moments when one frame of reference is more useful than the others. At the same time, it may not be possible to easily translate from one frame of reference to another, but with the use of additional information, such a frame of reference translation is generally possible.

2.2.2.1 Intrinsic

In the intrinsic frame of reference, an intrinsic spatial relator R is a binary spatial relation. The relation has arguments F and G, where R typically labels some sector of G. The volumetric center of G always defined by the origin X of the coordinate system C. The definition of the intrinsic relation $R(F,G)$, signifies that F is in a search domain that extends from G, via an angle or line projected from the center of G, through an anchor point A that extends for a determined distance. F and G have no limitations as to the objects that can be used. F can also be a part of G. The relation R is inflexible as it does not support transitive or converse inferences.

The intrinsic system is a search domain projected from a named facet of a landmark object. The main way that these facets are differentiated and named is by finding asymmetries or differences among the facets. There are three mayor ways to assign those asymmetries:

1. Using a fixed armature that is gravitationally oriented when superimposed on a chosen ground gives us a ‘top’, ‘bottom’ and ‘sides’.

2. Using the internal axial geometry of the ground object to assign major and minor

axes, supplementing these axes with an analysis of volumetric properties, a way of segmenting objects into parts, and a classification of protrusions.

3. Using notions of canonical orientation, functional orientation, functions of parts and direction of motion.

However, it is very important to note that designation of facets does not follow an 'inherent' methodology, meaning that they are all culturally imposed and assigned.

2.2.2.2 Relative

In the relative frame of reference, a relative relator R expresses a ternary spatial relation. The relation has arguments V, F and G. F and G have no limitations. V must be centered on an observer with V being different than G. V is the center of the primary coordinate system. A secondary coordinate system with center G can be used simultaneously. When used, the secondary coordinate systems on G can be mapped from the primary origin on V by utilizing rotation, translation and reflection.

The relative system imports the viewer's intrinsically defined bodily axes and maps them onto the ground object, creating angles in the process. There are at least three ways that this can be done. The viewer's egocentric axes can be translated as is onto the ground object. The other two options translate the egocentric axes under rotation or reflection. Due to this requirement, it is clear that the relative system strongly codes for the viewpoint of the user.

The system becomes quickly complicated as the user must keep both coordinate systems in mind, but there are many advantages to using such a system. First of all not all useful landmark objects will offer clearly distinguishable facets using only intrinsic criteria. Second, relative systems support logical inferences permitting users to come to conclusions such as if A is right of B, and B is to the right of C, then A is to the right of C. Lastly, this system connects directly to visual experience, since in one picture or a memory, one immediately has all the necessary information to able to describe the picture or the memory using relative terms.

The relative frame of reference is limited by the viewpoint V utilized. In order for another user to come to the same description, they would have to adopt the initial user's viewpoint. This is an interesting point as the relative frame of reference also allows the description of motion events and these are realized by normally holding the viewpoint constant.

2.2.2.3 Absolute

In the absolute frame of reference, an absolute relator R expresses a binary relation between F and G. The relation affirms that F can be found in a search domain at the fixed locus R from G. G is almost always the center of the coordinate system's origin X. The system is composed of an endless sequence of parallel lines that define a conceptual slope S across G. G may be any object. F may be a part of G.

The absolute system uses a fixed locus or 'slope' to define a direction from a ground object. These systems give us a bearing, which may have be a landmark within a territory, but are abstracted so that the resulting 'slope' can be used in any situation. This has the advantage that the system functions without the need of viewpoints. The lack of viewpoints yield elegant spatial descriptions. Not only do you obtain the ability to make logical inferences, as when using the relative frame of reference, but you also can know that your inferences are valid for everyone using this frame of reference.

Motion descriptions come extremely natural in these systems as direction can be described without the need of places, landmarks or grounds. Two moments in time are sufficient to fix an absolute angle of motion and similarly alignments can be specified without locations since any linear figure passing through two points would give a bearing.

2.2.3 Solutions when motion is involved

Motion events can be located as wholes, meaning that one can use all the resources available for describing a static location. On the other hand, the direction of a motion event can be explained without the need of coordinate systems or frames of reference. This can be easily achieved by mentioning two points along a given trajectory, with the source, the goal or a particular waypoint of the motion being the most common utilized points. Although it is possible for the direction of motion to be described without the use of frames of reference, most languages frequently employ all three frames of reference presented to describe motion. Motion is without a doubt more complex than location because it involves the temporal dimension. Integrating time automatically includes the change of location, the manner, the medium being used and the instrument enabling the motion. All this information is coded into languages, but there is no one method that encompasses all the possible codifications [26, Chapter 3.5].

2.3 How naming is modeled in computer networks

The basic model of network naming was introduced by Jerome H. Saltzer in his paper which appeared in 1982 [45] and later became RFC 1498 in 1993 [46]. In this paper the author applies the successful principles utilized in naming for operating systems to networks. Saltzer is able to adhere to Shoch's initial definitions for names, addresses and routes [47] to note that there are at least four types of objects that he believes need to be named in a data communication network: services, nodes, network attachment points and paths. The author touches on the point that these objects can be described from the point of view of the changes in the bindings between these objects.

In his work, Saltzer makes two interesting affirmations for these objects. First, the author points out that an object's name must be constant when related to some property in an established scope. Secondly, the author's requirements that the objects referenced to by a service, node or network attachment name be able to be transfered within their scope without losing their identity supports the idea of inherent mobility support. To accomplish this, the names given to services, nodes, network attachments points or routes are not as important as maintaining and updating the list of their bindings, also commonly referred to as *mappings*.

In order to discover these bindings between the objects, Saltzer maintains that there have to be three conceptually unmistakable binding services within the network. The services Saltzer names are the Service name resolution, Node name location and Route service. Each one of these services permits the mapping of one named object to another. Respectively this mapping is from service to node, from node to network attachment and the network attachment that leads from where one is to where one wishes to go.

The use of *distinct* binding services is of particular interest. This seems to imply that each identified object has distinct names at the service, node and network attach point level. That seems to imply separate namespaces. Another implication is that service names seem to be location independent and that the mapping from service names to node names tells us where the service is located.

In the book [10], John Day, comments about Saltzer's process to send a Protocol Data Unit (PDU). Day mentions that Saltzer seems to know that there is a distinction between node names and point of attachment, but he doesn't specify it in the process he wrote referring to send a PDU. Day attributes this oversight to the fact that multiple links between adjacent nodes were extremely rare at the time Saltzer was writing. Noticing this more general case for networks, Day proposes an amendment to Saltzer's work, stating that one must first choose the next hop to get to the desired destination and then obtain all the point of attachment names that are associated to the chosen next hop. The revised process to discover bindings in order to send a PDU can be thought of as follows:

1. Find the desired service.

2. Find the node name on which the required service operates.

3. Find the next node name that will bring you one hop closer to the desired node

name.

4. Find mappings of all point of attachment names for chosen node name.

5. Choose a point of attachment through which to send a PDU.

With the modifications to the process, Day also suggests modifying the names for the binding services and their descriptions to those shown below:

1. **Directory**: maps service names to node names where the requested service is being executed.

2. **Routes**: returns a sequence of node names that permit PDUs to go from the current network location a a desired network location.

3. **Paths**: returns a list of point of attachment points that can reach the node name desired.

The resulting structure from Saltzer's and Day's statements are visualized in figure 2.2. The resulting model highlights some key aspects of network naming. There are

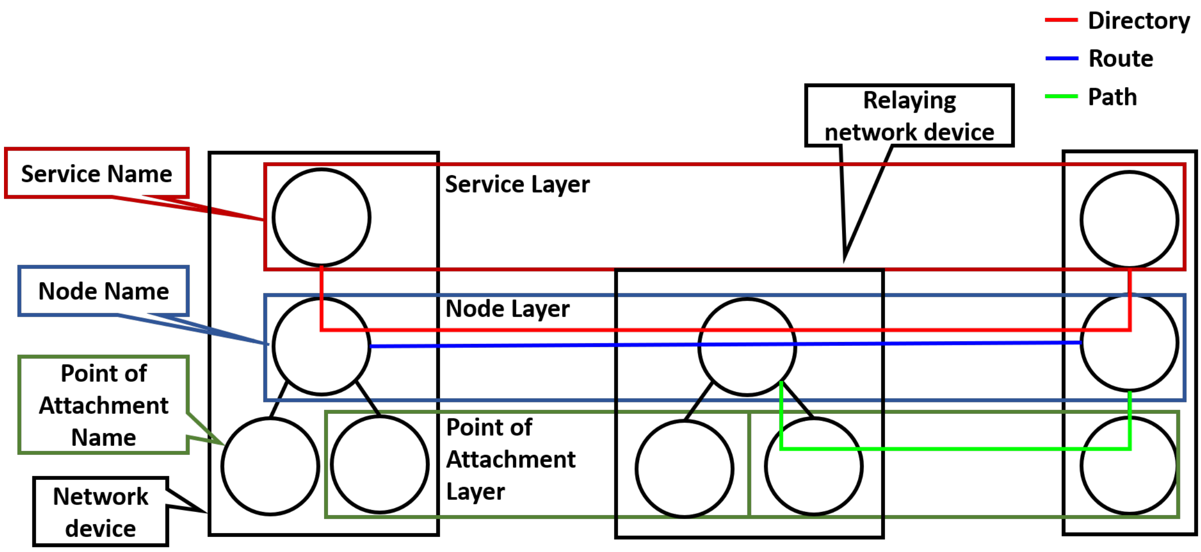

Figure 2.2: Saltzer's network naming structure and mappings with Day's amendments

three main objects to name; services, nodes and points of attachment. These objects are in three separate layers with corresponding separate namespaces. The scope of the point of attachment namespace is smaller than the node name namespace. The node name namespace in turn in smaller than the scope of the service name namespace. At the same time, these names are only used within the respective layer, meaning that the names must only be unique within the layer's defined scope. The model shows that routes are sequence of node names, thus the routing to a particular service should be done on a sequence on node names. This seems to imply that routing for a service name is confined to the namespace belonging to the layer below. Due to the logical construction of the structure and it's simplicity, one can imagine that following this naming structure would have implications in more specialized networking functions in which names play a primordial role, such as multi-homing, mobility and multicast.

It has to be stated that this paper is at best a suggestion for a naming structure as it does not go into how to go about actually build the namespaces or layers that it proposes. There is no model to guide us in this effort. This particular lack of a guide probably made it difficult to implement this type of network naming structure. In Section 2.5, it will become apparent that a network architecture can function without following this naming structure. Following such a network naming structure has trade-offs, but it is not clear what type of trade-offs are absolutely necessary.

2.4 Use of linguistic properties in network naming schemes

Section 2.1 established that human languages have universal properties that make them well suited to place specification. There are three specific properties addressed in this document.

The first property is that place specification is always expressed as a spatial relator. Section 2.2.2 mentioned Levinson's frames of reference and how they can be constructed.

One particular interesting fact is that all frames of reference start by establishing the figure (F) primitive, what one is looking for, and then the speaker establishes a ground (G), which is a surrounding area in which the F can be found. The spatial relator is usually denoted as R. Although the definition of G varies in every frame of reference it is always possible to define a G thanks to topological prepositions. Before touching on topology, it is important to note that place specification always entails the definition of a binary relation between a F and a G. In human languages, the G of highest scope is usually implicitly defined as the surface of the Earth.

As mentioned in 2.2.1.1, topological prepositions are always available. In language, topological prepositions help describe spatial relations of proximity, order, enclosure and continuity of objects in a given space. If one can define a figure F, then via topological prepositions, using semantic notions like 'near', 'at', 'between' or 'in', one can easily narrow down the G in which one wishes to operate.

The final property to focus on is the temporal dimension. As discussed in more detail in Section 2.2.3, motion is more complex than static location because it necessarily involves the temporal dimension. Integrating time automatically permits the inclusion of information about the change of location, how the location changed, the medium being used to create that change and the instrument enabling the motion. All that information is not always needed, but in order to be able to track certain types of motion, some form of time related information is absolutely needed.

The availability of these properties implies that even simpler human made languages, such as the set of questions and answers used in computer network place specification, could also have these traits. Since humans can communicate location information among members of the species in a highly sophisticated manner, it seems possible to leverage the work and findings of Levinson and his collaborators to evaluate forwarding strategies in ICN. The evaluation holds merit because what forwarding strategies attempt to do is to guide the PDUs, in other words *how* to best reach the location for a network name after calculating *where* it is located. Assuming that the place specification strategies used in human languages can be adapted for forwarding strategies, it would be extremely important for properties found in human languages to be available in the namespaces utilized in computer networks.

This reasoning motivated us to elaborate a simple questionnaire to determine if the properties of human languages appear in the names used in a computer network. Section 2.2.1.1 described how Levinson and his collaborators found that topological properties permeate all codifications of place specification in languages. When dealing with computers networks the scope of the questions is limited to a metric space. Metric spaces have the advantage that they satisfy the conditions for topological spaces, allowing us topological reasoning, while giving us numerical values to deterministically define conditions related to 'near' and 'far'. A metric space is formally defined as an ordered pair (M, d) where M is a set and d is a metric on M. In other words a function

$$d : MxM \to \mathbb{R}$$

such that for any $x, y, z \in M$, the following holds:

1. $d(x, y) \geq 0$
2. $d(x, y) = 0 \iff x = y$
3. $d(x, y) = d(y, x)$
4. $d(x, z) \leq d(x, y) + d(y, z)$

Function d is commonly referred to as a distance function.

For a network object with name n that needs to be found, the questionnaire is as follows:

Question 1.

Is the name n part of a metric space $M_N, N \in \mathbb{N}$?

Question 2.

Can every $n \in M_N$ be located in a single defined ground G with G being a metric space $M_{N-x}, x \in \mathbb{N}$?

Question 3.

Is n's location updated over time?

This simple questionnaire is derived from the fact that in place specification in languages there are always topological descriptors available. This is extremely important for naming since it is known that human languages heavily leverage these descriptors in place specification.

Question 1 asks if a name n being used in a network naming scheme can be placed in a particular set of names and if these can be categorized in such a way that they can be organized and ordered by a distance function. The ability to do these actions permits forming a notion of 'near' between two names of the same space.

Question 2 deals with the scope of the ground G used by a name n. Section 2.2 discussed the implicit use of the Earth's surface as the ground G with the largest scope in human languages. From the same section it is also known that in human languages, place specification relies on establishing a binary relation between a defined figure F and a defined ground G, even if G is only implied by context. This means that the name n referred to in question 1 is the F in the binary relation. This also implies that in network naming structures, a G always exists. An interesting implication is that if it is known that the relation between a F and a G is binary, a recursive place specification can be performed such that the G_1 of a F_1 can also be seen as the the F_2 of a G_2, where G_2 has a different scope when compared to G_1. This implies that F_1 could also be located in G_2. What question 2 aims to determine is whether all n used can be located in in some G, regardless of the number of necessary steps to obtain that mapping. The requirement for G to be part of a metric space is to be able to form a notion of 'near ' to be able to narrow the search within G for a particular F.

Question 3 is derived from our knowledge of place specification in languages when motion is involved, as described in Section 2.2.3. Although there are no definite methods to codify motion in languages, the locations g_n of a name n must be somehow maintained and updated over time.

Our questionnaire is of no use unless one can concretely define what it is that one are attempting to find. In other words, one needs to define the figure F. Without defining figure F, it is impossible to contemplate the definition of a shared ground G. In order to help define what F is, one can utilize Saltzer's model as was described in 2.3.

In Saltzer's model the components are placed in layers, with the service layer being on top of the node layer and this one in turn on top of the point of attachment layer. Using the definition of place specification, one can then easily say that a particular service name, our figure F, is located on a particular node name. What our research has highlighted is that the node name would be part of a ground G, a search area. Using the same methodology, a node name could be said to be located at a particular point of attachment name, cementing the idea that the point of attachment name is the ground G for the node name. When discussing the reasoning of question 2 it was mentioned that it didn't seem to matter how many spatial relators were necessary to get from the defined figure F to the ground G. Under this assumption, one could think that it would make sense to completely skip the node layer and link the service name to the ultimate ground G as a point of attachment name. In these cases, the idea of scope has to be used to determine if such a mapping is acceptable.

Scope forces one to think about how to communicate between sections in the scope. From the way that Saltzer defines the layers, the point of the attachment turns out to be much like a point-to-point link. This would mean that making the ultimate ground G be part of the point of attachment name would be too narrow a scope. On the other hand, the scope of the node layer is much larger, permitting the connection between any two services on the layer above. It is on this type of layer that question 2 seems to best fit.

In Saltzer's model, the services being used are not thought of as being part of some group. This easily permits the service names to not have any structure. However, with any big enough number of services, there might be the need to know whether a set of services belong together. In these cases, question 1 becomes helpful, as defining a distance permits the categorization of certain types of services to easily know which are related.

The notion of time is not part of Saltzer's model. Although Saltzer mentions that bindings are useful so that when a network component moves, Saltzer and Day do not go into detail about how these bindings should be maintained. Section 2.2.3 showed that mobility requires a look at the temporal dimension. Question 3 is thus not directly realized in Saltzer's model.

Utilizing the Saltzer's model and the emphasized information obtained from understanding place specification in human languages, one can hypothesize that a computer network architecture which answers the three questions formulated above affirmatively when the architecture's naming scheme is investigated, would be able to maintain a higher level of connectivity for nodes. This becomes possible because it would bring network naming schemes closer to what human languages are already being capable of realizing. The benefits of such a naming scheme would be seen in lower delay and higher goodput, particularly in mobile environments. These benefits are feasible because a routing scheme leveraging these naming properties would be aided by the calculation of distances and the continuous locating of network objects.

The use of frames of reference (Section 2.2.2) makes it seem that for network naming, an absolute frame of reference might be the best. It would be beneficial for every node in a network graph to be able to interpret the absolute directions of PDUs, without having to know much about the neighboring devices. In the absolute frame of reference, one can define only one 'slope'and permit all notions of direction to extend from this idea. Fortunately, an overview of Internet generators [31] [54] [44] and a list of famous network graph collections [37] show a distinct tendency to create various trees joined by mesh networks. For such a case, one can always apply an absolute frame of reference that points to the root of the trees. As was detailed in Section 2.1 the main use of a particular frame of reference does not limit the use of other frames of reference when they are required. What may happen is that one requires more information to translate from the absolute frame of reference to the other frame of reference required.

2.5 Information-Centric Networking Architecture

ICNs were created in a desire to move from host-centric networking to information-centric networking. These networks function by labeling information. This naming of information means that every piece of information that can be requested has a label which will be referred to in this paper as an ICN name. Communication is driven by consumers who request information utilizing an Interest PDU with the desired ICN name. The basic Interest PDU is shown in Figure 2.3. Note that the PDU has a field Nonce in order to distinguish two PDUs with the same ICN name. The PDU also holds a Type field, as it is required for some of the enhancements that utilize the Interest PDU for operations other than retrieval. Nodes who participate in this network, named Content Routers

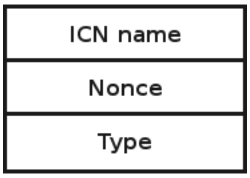

Figure 2.3: Interest PDU

(CRs), forward the Interest PDU until the corresponding data to the ICN name is found. The information is returned in a Data PDU which uses the ICN name. The basic Data PDU is shown in Figure 2.4. To carry out Interest and Data PDU forwarding, each CR

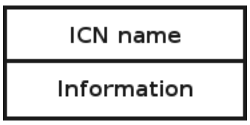

Figure 2.4: Data PDU

maintains three data structures. The Pending Interest Table (PIT), shown in Figure 2.5, keeps the mapping between ICN names from Interest PDUs and their ingress interfaces.

The entries in the PIT are those which have not yet been satisfied. The Forwarding Information Base (FIB), shown in Figure 2.6, keeps the mapping between ICN names and egress interfaces. These entries are leveraged by algorithms, named forwarding strategies, to determine where to forward Interest PDUs by their ICN names. The CS, shown in Figure 2.7, is a information cache used to keep copy of information from Data PDU that has already been retrieved. The information in a CS can be retrieved by ICN name.

ICN Name	Interface

Figure 2.5: Pending Interest Table (PIT)

ICN Name	Interface

Figure 2.6: Forward Information Base (FIB)

ICN Name	Information

Figure 2.7: Content Store (CS)

The flow of Interest PDUs is shown in 2.8. The flow of Data PDUs is shown in 2.9. When an Interest PDU arrives at a CR, the CR checks the CS for data matching the ICN name used in the Interest PDU. If a match exists, the CR returns the information in a Data PDU on the interface from which the Interest PDU arrived. Otherwise, the ICN name is looked up in the PIT and if a matching entry exists, the incoming interface identifier is aggregated to the PIT entry. A forwarding strategy is then utilized to forward the Interest PDU. The Interest PDU continues to be forwarded until an ICN application or a CS has the desired information and returns it in a Data PDU or when the Interest PDU time-to-live (TTL) is surpassed and is discarded. When a Data PDU arrives, the contained ICN name is used within the CR to find the related PIT entry. The Data

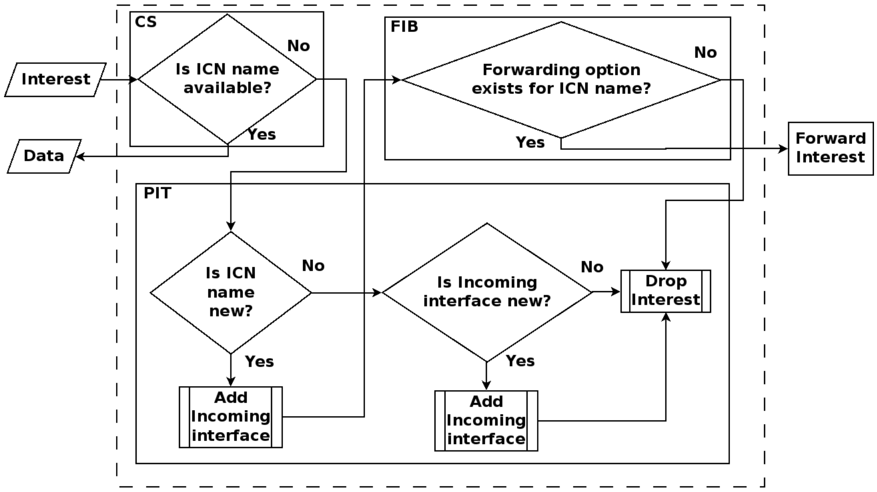

Figure 2.8: Forwarding process for Interest PDUs at an ICN node

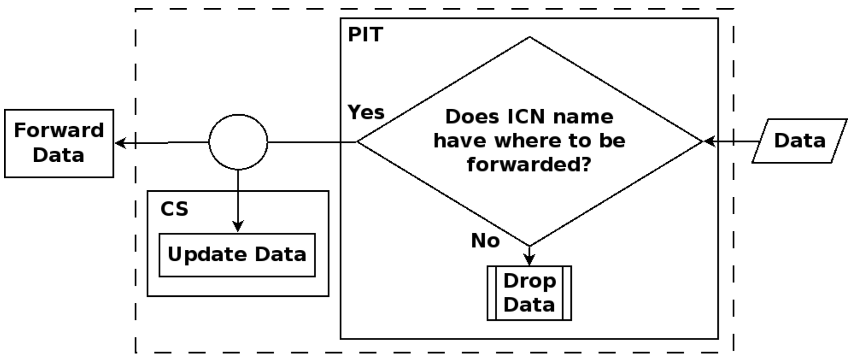

Figure 2.9: Forwarding process for Data PDUs at an ICN node

PDU is then forwarded to all interfaces listed in the PIT entry. Once the Data PDU is forwarded, the PIT entry is removed and the information in the Data PDU is stored in the CS for future used. The maintenance and cooperation of CSs in an ICN is one of the key elements to obtain a high user Quality of Experience (QoE). How CSs cooperate is a topic that continues to present interesting solutions [?] [?].

Another key element to obtain a high user QoE in ICN is the selection of an appropriate forwarding strategy. Forwarding strategies determine where received Interest PDUs should be transmitted. Usually the forwarding strategy is linked with the FIB, but this varies depending on the ICN implementation. Since forwarding strategies are usually based on the mapping of ICN names to possible transmission locations, Section 2.5.1 and Section 4.4 is dedicated to describing these strategies in more detail. There are numerous differences among the ICN implementations. However, most implementations tend to stick to the generic flow described in [23].

The most important ICN architecture proposals and research topics have been summarized in [15].

2.5.1 Forwarding strategies

Due to the initial simplicity of ICN, each implementation offers a large selection of forwarding strategies which map a given ICN name to an interface. All other identifiers are completely by-passed. The definition for an interface is vague in ICN architectures. A lot of the implementations assume a core network layout, leveraging the assumption that multiple users, via the aggregation of PIT entries by ICN name and interface, will want the same data [57] [23]. Thus one can assume that ICN architectures are referring to an entity through which one can communicate to a distinct set of neighboring nodes on the network when they talk about interfaces. The main forwarding strategy and enhancements that are of interest in this document are described in the following sections. A summary of ICN forwarding research topics can be found in [24].

2.5.1.1 Flooding

Interfaces in ICN receive local identifiers. This makes it possible to know from which interfaces Interest PDUs ingressed into the system and through what interfaces the PDU was forwarded. In Flooding, this information is used build a simple forwarding strategy that forwards Interest PDUs through every single interface available, except the ingress interface. Since the algorithm does not need to keep track of anything other than the ingress interface for a given ICN name, the processing at a CR is extremely low. Consecutive nodes repeat the same process until a Data PDU is received or some set threshold is surpassed. The drawback of this forwarding strategy is that it continuously floods neighboring nodes with Interest PDUs.

2.5.1.2 Smart Flooding (SF)

In Smart Flooding (SF), described in detail in [55], interfaces are given one of three states, green if it is known that information can be retrieved using the interface, yellow if it is uncertain that the interface can retrieve information and red if it is clear that information cannot be retrieved using that interface. When an ingress Interest PDU gets passed to the FIB, the PDU is forwarded through the highest ranked green interface available. After a predetermined amount of time, if the PIT entry for the related ICN name continues to be unsatisfied, the previously high ranked green interface is demoted and transmission using lower ranked green interfaces are tested. As long as the PIT entry is unsatisfied, the process continues, in descending order, through all green, yellow and red interfaces available. When a PIT entry exhausts the FIB entries available for the ICN name, then a special Interest PDU, called a Non-Acknowledgement (NACK) PDU, is transmitted. Neighboring nodes whose ingress interface receive this NACK PDU are automatically set to red state for the related ICN name. Should the NACK PDU set all FIB entries for the ICN name to the red state, then the NACK PDU is again forwarded. On the other hand, when the ICN named data is found and sent in a Data PDU, the nodes who receive this

PDU automatically promote the related ingress interface to the green state.

2.5.2 MAP-Me enhancements (MM)

Due to the FIB mapping ICN names to interfaces, when a producer is mobile, the neighboring node that used to connect to the producer may continue sending PDUs out the interface to which the producer is no longer connected. As only consumers can retransmit Interest PDUs, depending on how these are handled, significant delay is expected [4].

To overcome this disadvantage, the authors in [3] create an update protocol to support producer mobility events by dynamically updating the ICN FIB. When a producer changes point of attachment, it sends a modified Interest PDU, named MAP-Me-IU. The MAP-Me-IU includes the ICN name that the producer is offering, along with a sequence number. This method ensures that the newest MAP-Me-IU is utilized to update the FIB. Since the MAP-Me-IU leverages an Interest PDU, when the PDU is received by a CR it can be forwarded in the same way as a normal Interest PDU. This permits the MAP-Me-IU to be forwarded via all interfaces from which Data PDUs related to the ICN name have been received. After forwarding the MAP-Me-IU, the interface on which the MAP-Me-IU was received is used to update the CR's FIB. The ingress interface for the ICN name is enabled if the sequence number on the MAP-Me-IU is newer than previously see MAP-Me IU at the CR. A significant drawback of this protocol is that to correctly function, the network must have a separate global routing protocol that can create and update the FIB of network CRs.

Chapter 3

Network Naming and Addressing

The previous Section 2.3 described Saltzer's naming structure for networks, the objects that apparently need to be named and some of the interesting implications that looking at this model makes clear. It was also mentioned the fact that Saltzer's structure was at best a suggestion. This chapter maps Saltzer's resulting structure to real functioning network architectures and sees what kind of insights can be obtained.

Using the developed model and questionnaire, Saltzer's structure will first be mapped to the Institute of Electrical and Electronics Engineers' (IEEE) 802 Extended Unique Identifier (EUI) 48 and 64 identifiers in Section 3.1. Seeing how most of the global public Internet infrastructure is based on IP network architectures, Saltzer's structure is mapped to that architecture in Section 3.2 and 3.3. In Section 3.4 Saltzer's results are mapped to the ICN architecture described in Section 2.5. The evaluation of the network naming schemes with the questionnaire developed is described in 2.4.

3.1 Naming in IEEE 802

In the widely used IEEE 802 standards, which encompass the Ethernet and WiFi network technologies, globally unique identifiers are generally used for network interfaces in a burned-in-address manner. The IEEE Registration Authority (RA) controls the assignment of 48 bit EUI-48 or 64 bit EUI-64. EUI-48 and EUI-64 are globally unique and are intended to be bound to a hardware device instance or object that requires identification.

The initial 24 bits of both EUI-48 and EUI-64 are assigned numbers to organizations or companies for identification purposes. In the case of organizations, the 24 bits are called the Organizationally Unique Identifier (OUI). In the case of companies, the 24 bits are called the Company ID (CID). In both cases, the identifiers come from the same namespace. EUI-48 and EUI-64 are generally represented by pairs of hexadecimal numbers separated by colons [20].

Utilizing Saltzer's structure (refer to Figure 2.2) one can come to the conclusion that IEEE 802 identifiers are being used as part of the Point of attachment layer to name network interfaces. If this is true, then one has to wonder why the IEEE 802 namespace is so big. Even when taking out the initial 24 bits for organization and company identification, this leaves us, in the case of the EUI-48, with 24 bits for network interfaces, which is over 16 million names.

The point of attachment layer where this namespace is being used has the smallest scope. Looking at common IEEE 802 network setups, such as Ethernet switches, the number of ports commonly range between 8 and 48 devices directly connected to the Ethernet network device. If the point of attachment layer namespace for IEEE 802 were managed by the interconnecting network device, like say the switch, there would be no need for a namespace bigger than the number of ports the physically available on the device. This is because neighboring physically connected devices could not exceed the physically available ports. In the case of wireless networks, where physical connections are relatively easier, even if one exaggerated, 16 million devices in a mesh network would be a senseless configuration for a network.

Since the IEEE RA sells EUI-48 and EUI-64 to organizations, the consumers who end up using the organization's devices have no way to make their assigned EUI-48 and EUI-64 into metric spaces without ignoring the assigned number. There is no logical way to aggregate the identifiers, to summarize them into a shorter equally precise identifier to save space, and routing is useless because the scope of the point of attachment layer only reaches the next physical network hop.

It is clear, taking a look at Saltzer's structure, that the IEEE 802 namespace is not the size that it is because of networking requirements. A network device managed namespace anywhere in the range of 4 to 2048 names would suffice. The fact that IEEE 802 names are usually burned-in-addresses, in other words are physically written into the network interfaces so that they can't be changed easily, makes it easy to assume that what is desired is to globally identify the network device. This requirement is not network related and has already been widely exploited to track users [49] [9].

The method used to name objects in IEEE 802 is not, from any perspective, incorrect. However, if taking advantage of the questionnaire and the knowledge of place specification, the naming scheme for MAC addresses is strange. If one is to ask the question, "Where is a MAC address?", as mentioned in 2.4, one can notice that there is no defined ground G. At the same time, the current structure of MAC addresses doesn't give any method to know the distance between two addresses. This means that MAC addresses do not satisfy the conditions to be considered a metric space.

Making MAC addresses into a metric space becomes even harder when you realize that the addresses are managed globally but that they have no meaning among themselves. Taking into account that MAC addresses are generally used for one network hop operations, it is strange that these addresses are considered globally unique.

One must note that it is only in very rare cases that services are placed on directly on MAC addresses. In the cases where such structures are used, mostly in hard real-time systems that need heavy cooperation between specific devices, the simplicity of MAC addresses permits the creation of a toponymy. However, due to the MAC addresses being burned into the devices, there is very little flexibility to create a bigger network without having to add some other sort of naming structure. A list of such networks that use MAC addresses for these purposes can be found in [27].

3.2 Naming in IPv4 network architectures

When one thinks of IPv4 networks, one generally thinks of the 32-bit namespace, usually represented in dotted octet decimal notation, commonly known as IP addresses. However, IPv4 networks also depend on a 16-bit namespace for ports and has incorporated the Domain Name System (DNS) in order to not have to directly deal with the 32-bit namespace [25].

The history of the way the IP address namespace was defined is of extreme importance. Initially, the namespace was unmanaged and unbounded, with the only name blocks available were what would be later known as the Class A networks [7]. These class A networks permitted 2^8 networks to be created, leaving 24-bits for use by the owner of one of these networks. Seeing the initial growth of the network, in 1981 a naming scheme called classful network was implemented and documented in RFC 791. An IP address was assigned a class depending on the number of bits that defined the network. The remaining bits representing the number of hosts in the network [21]. The initial classes created were Class A, which contained all names in which the most significant bit is zero, using the next 7 bits to designate the network. This accommodates 128 networks each with 16777216 names. Class B had the two most significant bits set to 1 and 0, using the next 14 bits to represent the network. This created 16384 networks with each network having 65536 names. The final class, Class C had the 3 high order bits to 1, 1 and 0 and used the next 21 bits to designate the networks for a total of 2097152 networks with each network having 256 names. The new blocks of names were handed out in the order requested by organizations. This meant that you could get continuous blocks of names that could be assigned to geographical locations on different sides of the planet.

As the public Internet using IPv4 grew in size, it became increasingly obvious that a more fine-grained sort of namespace management was required. From the beginning of the class assignations, it became obvious that giving certain organizations over 16 million names was an inefficient use of the namespace. Better control of the namespace was provided by the integration of Classless Inter-Domain Routing (CIDR) [41] [18] and the ability to automatically give IP addresses to users via the Dynamic Host Configuration Protocol (DHCP) [13].

When considering how IP addresses are assigned, usually DHCP hands over an IP address per MAC address. This creates a one to one mapping from the IP address to the MAC address. It is now common for a lot of the devices, particularly mobile ones, to have various types of interfaces that may or may not have a MAC address but at least have some sort of identifier. In any case whenever a new interface is used, a new IP address is required. This would seem to suggest that IP addresses are basically being used as point of attachment names.

The IP network architecture does not have service names in it's specification. Instead, there is the definition of sockets which identifies an end-point of communication. Processes running in the operating system would use sockets to be able to use the IP network to communicate to other nodes. The biggest issue here is that sockets are defined by an IP address and a completely local name, a number between 1 and 65535, named a *port*. To be able to connect to a socket that is on another node, one would require knowing the IP address and the port that the process is using. Within the IP specification, there are registered sockets which have particular pre-determined services, like port 22 for SSH servers and port 80 for HTTP based servers, but these are conventions. One can run an SSH server on any port, thus can be using any of the 65535 sockets available and the user would have no idea. The only way to find out if the SSH server is even available is to try to connect to all the sockets. In the higher level protocols, User Datagram Protocol (UDP) and Transmission Control Protocol (TCP), which use ports are defined by socket pairs, which concatenate the IP address and the port number of the sender and the receiver.

Mapping what was described about the IP network architecture to Saltzer's structure results in Figure 3.1.

The creation of the the Domain Name System (DNS), formalized in RFC 1035 [34] took a step forward in trying to complete naming in the IPv4 network architecture, pro-

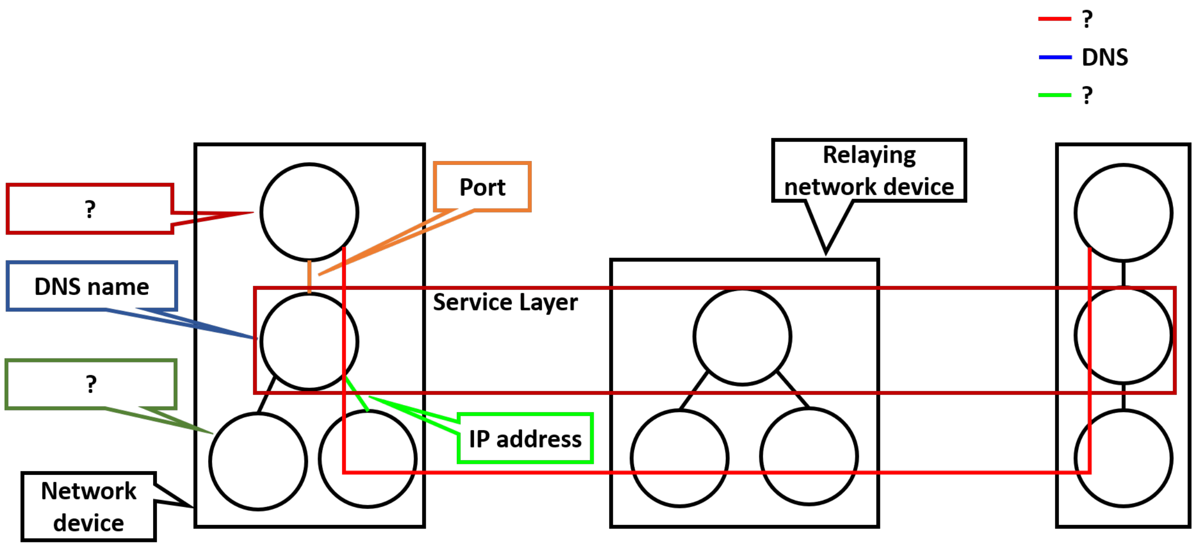

Figure 3.1: Mapping Saltzer's concepts to the IP network architecture

viding, among other things, a domain name to many IP address mapping. However, since connections are still being maintained via socket pairs, it means that at the network architecture level, there is still a one-to-one mapping between domain name, IP address and the point of attachment name. If the IP changes, every binding must change simultaneously, making it inevitable that you will lose the connection.

The questionnaire developed in Section 2.4 is answered for the IPv4 network architecture. When in the IPv4 network architecture, the first big question is what one wants to find. Remembering the information in Section 2.4, this means defining the figure F. This is already complicated in the IPv4 network architecture because it means finding a service, which is on a locally named port. This means that the port can't be searched until an IP address is found.

IP addresses are part of a 32-bit namespace to name interfaces. IP address blocks are managed by Regional Internet Registries which are then handed to Internet Service Providers (ISPs). ISPs use DHCP servers to hand out IP addresses to their clients. Although there is an implicit hierarchy, due to the way that IP address blocks have been managed, there is no assurance that two IP addresses that look similar are actually close within the graph that could be created using IP addresses. IP addresses are usually shown in the 4 octet notation, again emphasizing their hierarchy, but due to the the use of CIDR, it is not easy to create a clean definition of a distance between two IP addresses. The use of DNS names could have given the us the service name required, but the issue continue to be that DNS names map to IP addresses that map to a machine's interface. This means that the structure of the IP addresses, is both the key and the bottle neck in the architecture. IP address is the F. It doesn't have a distance function and the way it is tied to the other components in the network make it hard to create a viable distance function for the namespace. This means that for the network architecture, question 1 is negative.

As previously described, IP addresses are generally mapped to physical identifiers, such as MAC addresses. Under the definition of place specification, if an IP address is F, this would make the MAC address the ground G. As seen from the discussion on MAC addresses in Section 3.1, in the current IEEE 802 naming scheme, it is hard to develop a metric space. This implies that question 2 is also negative.

Unless the IPv4 network is employing Mobile IP infrastructure [22], the IPv4 network architecture does not track IP addresses throughout the network. DHCP servers do maintain a lease of the IP addresses, but these are considered for internal use only. The information about changes of IP addresses are not maintained for the network further hindering their use to locate nodes. There has been a lot of research into locating IP addresses into our physical realm via geolocation algorithms such as [56]. This location capability is unfortunately misguided. One thing is to know where on the planet a IP address is being used and another is to know where in the IP address graph a node is attached. It could well be that these two location are in fact mapped in a one to one fashion. However, in the current IPv4 network architecture this mapping cannot be used. Without being able to have information about where in the graph a node is located, it is useless to know where on the planet the node is located because this information cannot be used in routing. Remember that routing, according to the Saltzer structure, is done directly on the node names. In any case, if the IPv4 network uses Mobile IP infrastructure question 3 can be answered positively. Without the use of Mobile IP, the question is answered negatively.

3 negatively.

3.3 Naming in IPv6 network architectures

Apart from the size of the namespace for IPv6, which is enlarged to 128 bits, there is not much difference in how naming in an IPv6 network differs from how naming is realized in an IPv4 network. This means that the conclusions from Section 3.2 regarding IPv4 network architectures functions apply for IPv6 network architectures. One cannot rule out that future amendments of the IP network architectures may require a more thorough investigation into this particular topic and may yield different conclusions.

3.4 Naming in ICN architectures

The ICN name space is unbounded, unmanaged and currently has no real limitations in length or structure. The ICN names don't really identify applications of any sort, they identify specific unique information. Confusingly, ICN names are also heavily used in deciding through which interfaces Interest and Data PDUs are transmitted through. Using the descriptions of Section 2.3, since ICN names are used as location independent identifiers for what the requester wants, this can be thought of as a service name. The interface name is a local identifier. The point of attachment layer is left to network interface technologies, such as the IEEE 802 naming standard. In most ICN specifications, even the point of attachment name is made unnecessary as point-to-point connections are thought to become the most common among ICN CRs. The point of attachment name is included in the mapping as these are usually burned into the network interfaces and could be of use should the basic ICN architecture be modified.

Taking these points into account, one can map Saltzer's concepts to the ICN architecture as shown in the Figure 3.2. Please note that to explain the concept, MAC addresses are used as the point of attachment name in the figure because most computer users have experience with these names. There is no limitation in ICN to the network interface technologies that can be used.

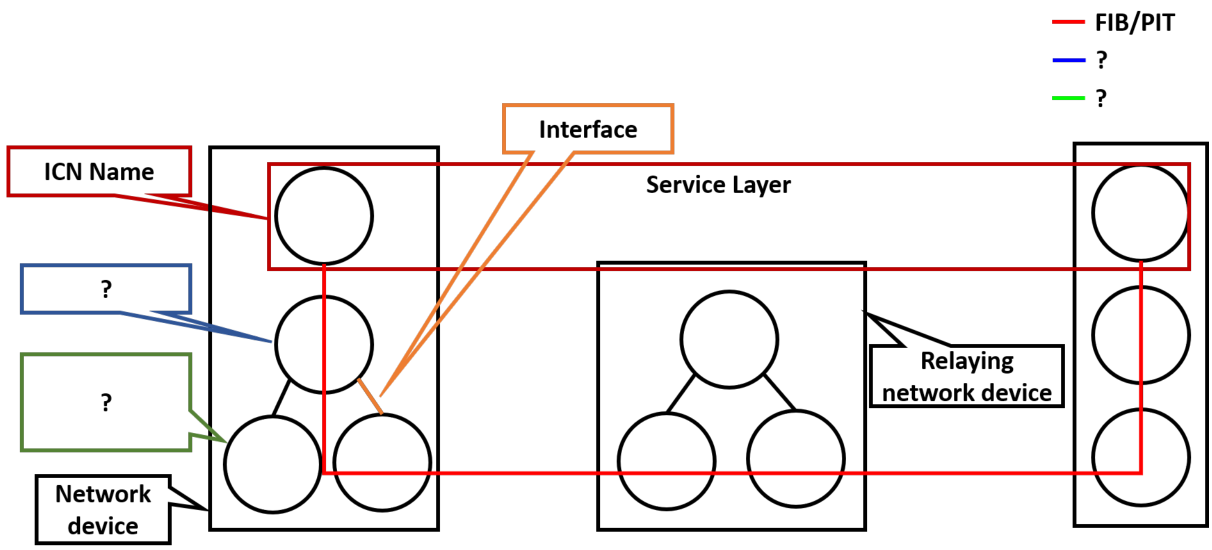

Figure 3.2: Mapping Saltzer's concepts to the ICN architectures

When Saltzer's binding services are mapped to the ICN architecture, the results are curious. In this document ICN names are viewed as service names. If that is the case, as shown in Figure 3.2, the node name space does not exist in the architecture. This would make the Directory binding service non-existent. On the other hand, forwarding, which is essentially finding the next hop in the network and also the definition of the Route binding service can be initially thought of belonging to the the FIB. However, when Data packets are returning, the route back is constructed by the PIT. In both cases, the FIB and PIT search for ICN names and return interfaces, which are local ports that connect to the point of attachment layer. This would make the FIB and PIT be Path binding services, the FIB only for Interest packets and the PIT only for Data packets. If the interpretation of the ICN architecture components is correct, the ICN architecture only has a Path binding service.

Since point of attachment names have the smallest scope, there is no easy inherent way to maintain track of where nodes have moved. This might explain why loss-less mobility and specifically provider side mobility is complicated in ICN architectures. The addition of node names would undoubtedly complicate the whole ICN architecture but would probably make it easy to maintain the bindings for mobile nodes be they users or providers. The introduction of node names might also be easier and cleaner than some of the proposals for mobile node tracking like [50].

Another result from applying Saltzer's concepts to the ICN architecture has very little to do with the resulting figure and more to do with the namespace chosen for ICN names. If routing is done with only this information on a unmanaged, unbounded namespace with no limitations, the ICN FIB would grow exceedingly fast. From Saltzer's concepts, it is clear that the layer must be managed so that the ICN names can be aggregatable in some form. If these aggregatable managed ICN names were to be used, the FIB would not increase in size dramatically over time. A drawback would be that the name space would not be as limitless as it is currently being defined.

The questionnaire developed in Section 2.4 is answered for the forwarding strategies presented in Section 2.5.1. From the main ICN explanation in Section 2.5 it is known that there is no naming structure for ICN names and that there is no defined distance between two particular names. Hence all forwarding strategies that use ICN names without modifications, question 1 is a negative. This applies for Flooding, Smart Flooding (SF) and Smart Flooding with MAP-Me enhancements (SF MM).

Section 2.5 also states that in the ICN architecture, the mapping of ICN names, the figure F of the place specification relation, is done to interfaces, the ground G. These interfaces are normally labeled to identify them. Even if the naming of interfaces used numbers, which would satisfy the metric space definition, the scope of the naming would be local to a particular node. This means that for all forwarding strategies that use the ICN name to interface mapping without modifications, question 2 is a negative. This applies to Flooding, SF and SF MM.

SF has the NACK PDU to avoid attempts to forward PDUs via interfaces through which there is historical information that suggests that a particular ICN name will not be found. Since NACK PDUs update the FIB over time, there is an update of the location of ICN names over time in SF. For SF, question 3 is therefore positive. SF MM attempts to speed up the update of ICN name to interface mapping by ensuring certain interfaces have a green state for future Interest PDUs. In the update, only the prefix of all the ICN names provided by the producer are updated, pointing all matching ICN names to one given interface, possibly undoing the work done by the SF's NACK PDUs. Due to this conflicting update question 3 is only partially satisfied. For Flooding, the system promotes itself as the simplest possible system, so it doesn't save any information over time, answering question 3 negatively.

The ICN naming scheme presented is not the only interpretation of ICN naming available. In [38], another interpretation of Saltzer's work for the ICN naming structure is presented. This interpretation does not include Day's modifications and the work does not go into any detail about the binary relator for place specification or the properties required for names for ICN to be able to leverage the names in a useful manner.

Chapter 4

Named Node Network Architecture (3NA)

The 3NA was proposed to handle seamless mobility by leveraging the use of network component names. The foundation of the 3NA architecture is the ICN architecture as described in [36] and [23]. Mobility is an inherent part of the 3NA architecture and is not treated as a service. Utilizing mechanism PDUs, all 3NA nodes are designed to have the same network components and capabilities. This was done to make network expansion relatively easy. The ICN namespace is not overloaded, but is slightly modified in Section 4.7 to deal with push producer scenarios. Two new managed namespaces are included in the network architecture. These namespaces provide support for mobility. Section 4.1 will detail the namespaces used in 3NA. Section 4.2 describes the modifications and additions made to the ICN structures. All the mechanisms and data transmission PDUs introduced in 3NA are detailed in Section 4.3. As 3NA is a ICN inspired network architecture, the forwarding strategies that have been implemented are described in Section 4.4. How 3NA integrates mobility is explained in Section 4.5. Due to number of mechanism PDUs involved in mobility, a simple example of mobility is offered in Section 4.5. The knowledge acquired in Section 2.3 and Section 2.4 is then applied to 3NA in order to determine improvements to support other mobility use-cases. The details of the improvements are described in Section 4.7.

4.1 Namespaces

Section 3.4 showed that two fundamental namespace were lacking in ICN architectures, the namespace that names nodes and the namespace that names points of attachment. These namespaces were added to 3NA.

Nodes that belong to the 3NA network are uniquely labeled from the first additional namespace. 3N name is the label given to names from this namespace. The 3N namespace has a single multi-level structure which aims to be a metrizable topological space [6] [5]. Assignation of names from this namespace is heavily managed by the network by the use of mechanism PDUs (see Section 4.3.1). The management ensures that the namespace can maintain topological properties even if the nodes in the network increase. The topological properties will be discussed in greater detail in Section 4.7. The strict assignation of 3N names also permits the names to be aggregated within the structures that require the naming information. As was expressed in [**?**], such strict rule based naming shows promise in improving PDU forwarding. The uniqueness of 3N names is limited to the size of the network being built.

The Point of Attachments (PoAs) used by a node to engage with the desired network are labeld from the second additional namespace called the PoA namespace. The namespace is composed of names usually used to identify interfaces. The uniqueness of PoA names is limited to the nodes that co-exist on a shared communication medium. EUI-48 and EUI-64, are examples of names that are included in this namespace.

The ICN namespace that labels information within an ICN network was left unmodified. For this document, the NDN architecture's format described in [36] is followed for the names. The common label for names belonging to the ICN namespace are ICN names.

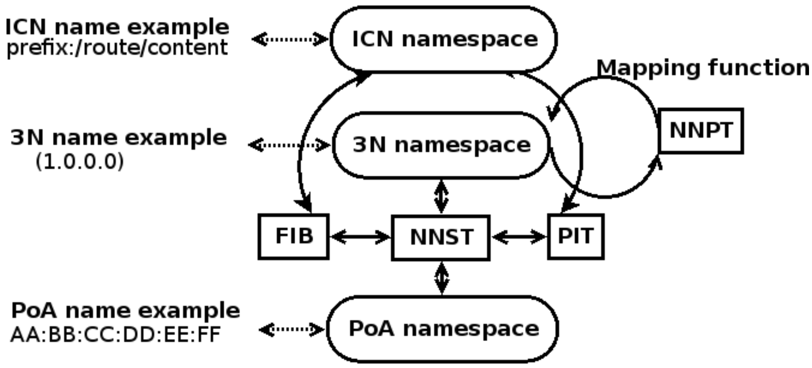

Figure 4.1: 3NA namespaces and mappings

4.1.1 Namespace mappings

As was described in Section 2.3, the bindings between namespaces is important. The bindings for the 3NA and the usage of 3N names are shown in Figure 4.1. There is a one to many mapping of 3N names to PoA names via the NNST. One 3N name maps to the newest 3N name used in the network via the NNPT. The initial network naming scheme avoids creating a direct map between ICN names and 3N names. Ingress interfaces along with 3N names are aggregated in the 3NA PIT after seeing Interest PDUs from 3NA enabled applications. This results in a one ICN name to many interface and 3N name mapping. The FIB was extended to permit the storing of 3N names, but the functionality was not used, meaning that there is only a one ICN name to many interface mapping. This is unmodified from the common ICN architecture naming scheme.

Additional information about the structures mentioned here can be found in Section 4.2.

4.1.2 Naming in 3NA

The questionnaire detailed in Section 2.4 is answered for 3NA's initial proposal [1] [2].

As was briefly described in 4.1, 3NA goes out of its way to not map ICN names to 3N names. This was an intentional choice as at the time the purpose was to convince more people that including a namespace for the nodes was worth the effort. As the questionnaire was developed, this initial effort seemed to be problematic, since the lack of mapping made question 2 be negative. However, the node name space was initially thought to maintain a metrizable topological name space it is possible to give an intermediate rating to question 2.

The ICN namespace in the original 3NA proposal was not modified. As in the normal ICN architecture, this meant that ICN names could be considered services, and thus the figure F to look for in the network. Just as in the ICN architecture, the mapping for ICN names was limited to interfaces. The ICN names themselves didn't follow any naming scheme, making question 1 also negative.

The inclusion of the Node Name Pair Table (NNPT, see Section 4.2), permitted the mapping of 3N names to 3N names. Under the ideas detailed in Section 2.2.2, such a mapping within the same search domain is possible for place specification, particularly if the mappings are maintained over time. As will be explained in Section 4.3.1, the mechanism PDUs ensure that the mappings are maintained, leading to the questionnaire's question 3 to be answered positively.

3NA does not originally modify the ICN names, so using the information from Section 2.4, it is known that question 1 has the same result as the ICN architecture. 3NA in it's current state attempts to keep the names given to nodes in a strict topological manner, meaning that 2 could be positive due to the 3N names having that structure. However, without the mapping from ICN names to 3N names, the answer to question 2 cannot be positive. The question is given a $\triangle$, or intermediate response to this question. Due to all the mechanism PDUs being used, in particular the INF PDU, the network is keeping track of 3N names over time. This means that the network would answer question 3 positively.

4.2 Structures

The PIT and FIB which are common components of the ICN architecture were modified for 3NA. These were extended to keep 3N name information as is shown in Figure 4.2 for the 3NA PIT and Figure 4.3 for the 3NA FIB. The modification of the 3NA PIT allows the additional aggregation of 3N name information. The 3N names are stored alongside the ingressed interface for the Interest PDU. This modification allows communication to nodes with 3N names by encapsulating ICN PDUs in DO PDUs. Although not initially utilized, the 3NA FIB was also modified to allow the storing of 3N names.

3NA also has two additional structures that handle the additional Node Name namespace and the Point of Attachment namespace.

ICN Name	3N name	Interface

Figure 4.2: 3NA PIT

ICN Name	3N name	Interface

Figure 4.3: 3NA FIB

The first additional structure was the Node Name Signature Table (NNST). The binding of 3N names to many PoA names is maintained in this structure. Neighboring nodes can be directly contacted using the information in this structure. A diagram of the NNST can be seen in Figure 4.4. The second additional structure was the Node Name Pair Table (NNPT). The binding of an old 3N name to a new 3N name is maintained in this structure. This mapping is utilized in verifying whether a particular node with a specific 3N name has seen movement within the network. The NNPT was created to deal with network flows with long Round Trip Times (RTTs). A diagram of the NNPT can be seen in 4.5.

3N name	PoA name	Interface

Figure 4.4: 3NA NNST

The main use for Interest PDUs, Data PDUs and interfaces within 3NA structures are not changed in order to maintain interoperability with common ICN architecture

3N name	3N name

Figure 4.5: 3NA NNPT

implementations. The way CSs are used in 3NA does not differ from the way they are utilized in the common ICN architecture [1]. All nodes that can operate in 3NA have the structures detailed in this section. Due to 3NA nodes maintaining the same columns of information as in a normal ICN implementation, 3NA capable nodes can handle communications with nodes only capable of transmitting with Interest and Data PDUs.

4.3 PDUs

With the addition of new namespaces and structures, it was necessary to create new PDUs to leverage the newly available network information. PDUs were separated into data transmission PDUs and mechanism PDUs. The purpose was to completely separate the PDUs that transmit information between applications in the network (data transmission PDUs), from the PDUs that directly modify the operation of the network layer (mechanism PDUs). All the PDUs function with timers to determine the nodes protocol machine state. This design attempts to minimize PDU exchanges [53].

4.3.1 Mechanism PDUs

In 3NA, the network layer can be modified by interacting nodes to optimize operations between them. Mechanism PDUs are the way that these modifications can be realized. During this research the requirements for network layer modification were related to the 3N names in use by the network. Of particular interest were the procedures for enrollment, dis-enrollment and re-enrollment of nodes into the 3NA network. As a design principal, there was a need to be able to have simple debugging for the procedures, which is why

Table 4.1: Mechanism PDUs defined in the 3N Architecture

PDU	Description
INF	This PDU is utilized to inform neighboring nodes about 3N name changes
EN	Begins the enrollment of a node into a 3NA network
OEN	Enrolling nodes are offered 3N names in this PDU
AEN	Enrolling nodes acknowledge the acceptance of an offered 3N name using this PDU
REN	A re-enrolling node uses this PDU to connect to a 3NA network. The node must already have a valid 3N name
DEN	Nodes wishing to dis-enroll from a 3NA network utilize this PDU
ADEN	Dis-enrolling nodes receive this PDU from ONNs to acknowledge their dis-enrollment

the created mechanism PDUs, described in Table 4.1, have singular uses. Since these mechanism PDUs are related to 3N names, all the PDUs enable actions on the NNST and NNPT. As it is not easy to picture the usage of the mechanism PDUs with only their description, later examples are based on the network graph shown in Figure 4.6.

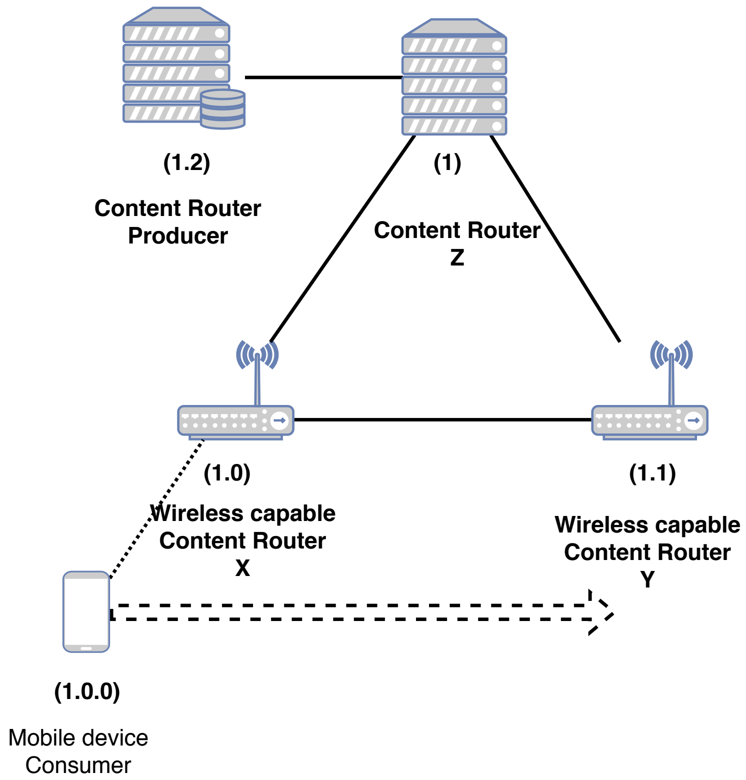

Figure 4.6: Example 3NA network

All data transmission in 3NA is enabled by 3N names. The usage of 3N names implies being able to obtain a 3N name. Nodes wishing to use a 3N names must solicit one from the network. This process is called enrollment. A node can be part of 3NA without a 3N name, but in those cases the node will be treated like an ordinary ICN capable node.

Utilizing an EN PDU, any node that has a PoA name and is connected to a physical medium that leads to a 3NA network, can obtain a 3N name. The enrolling node, the node that transmits an EN PDU, is called a Soliciting Node (SN). As it is possible for a node to have many PoAs on many physical mediums, the specification for the EN PDU requires the PDU to be sent containing all the SN's available PoA names.

When an EN PDU is received by a Offering Node (ONN), a verification is executed on the ONN's NNST for the PoA name present in the EN PDU. ONNs are capable of functioning without having a 3N name. However, in order to generate a 3N name when receiving a EN PDU, the ONN must have enrolled to the network at some point in time. The generation of the 3N name at the ONN attempts to maintain the properties desired by the network for the 3N namespace. The generated 3N name is created along with a lease. The 3N name along with the lease information is transmitted back to the SN in an OEN PDU as an offer. When the OEN PDU is received by the SN, if no issue is found with the 3N name or the lease, the SN can use the 3N name for any future data transmission. In response to the successful use of the 3N name in a OEN PDU, the ONN receives a AEN PDU from the SN, acknowledging the SN's successful enrollment. The arrival of a correct AEN PDU signals to the ONN that the whole enrollment procedure has been successfully finalized. The ONN must then update its corresponding NNST entries to reflect the new information.

Steps 1 to 3 in Figure 4.7 show an enrollment example.

In 3NA, changing the point of attachment to the network is handled by two different procedures. Assuming that a successful enrollment has taken place, the first procedure is a dis-enrollment. Once dis-enrolled and having selected the next point of attachment to use, the re-enrollment procedure can commence. Nodes that want to change point of

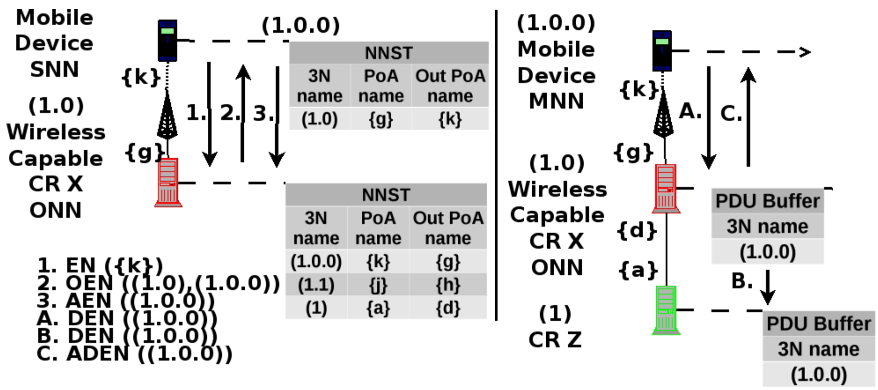

Figure 4.7: Example of Enrollment and Dis-enrollment in 3NA

attachment are called Mobile Nodes (MNN).

Dis-enrollment is begun by having the MNN send a DEN PDU with the currently used 3N name to the ONN that initially offered the MNN the 3N name. If PDU buffering is enabled, all nodes that receive the DEN PDU can buffer PDUs that are destined to MNN's 3N name. This buffering is done in an attempt to deliver PDUs that the MNN might be waiting for after its re-enrollment to the 3NA network. This option can be used to minimize overall response delay and application goodput. The ONN that receives the DEN PDU can also choose to forward the DEN PDU to its own parents in the network graph. The 3N namespace's topological properties help to easily determine parents. The ONN also acknowledges the dis-enrollment by transmitting an ADEN PDU to the MNN. With the ADEN PDU, the MNN is aware that all network procedures have successfully finalized and the change of point of attachment or re-enrollment can be initiated.

Steps *A* to *C* in Figure 4.7 show an example of dis-enrollment example.

To commence re-enrollment, the MNN attaches its 3N name and its available PoA names to a REN PDU which it transmits to the newly selected ONN. This 3N name is called the Old 3N name (O3N). The ONN receives the REN PDU and checks its NNST for matches to the information contained in the REN PDU. An OEN PDU with a new 3N name, called the new 3N name (N3N), is generated on the ONN if no match is found. This OEN PDU is then transmitted to the MNN. When the MNN receives the OEN PDU, much like in enrollment, the MNN starts using the N3N. The O3N is maintained by the

MNN until its original lease is expired. The MNN, like in enrollment, acknowledges the offer in an AEN PDU. The ONN receives the AEN PDU, integrating the 3N name to PoA mappings in the NNST. Different from the EN PDU process, the AEN PDU after a REN PDU triggers the creation of an NNPT entry, linking the O3N to the N3N. The ONN also creates a INF PDU containing the same information as the newly created NNPT entry to transmit all the way to the topological parent of the O3N. On each node that receives the INF PDU using each node's NNST to get the the ONN that generated the O3N, a NNPT entry is created, linking the O3N to the N3N. If the node has previously received a DEN PDU for O3N, the PDU buffer is retrieved and all PDUs are rewritten and redirected to the N3N. Once the INF PDU reaches the previous ONN and parent of the O3N, the ONN deletes the O3N related NNST entry, completing the re-enrollment.

Any node with an updated NNPT that sees PDUs using O3N as a destination, will redirect the PDUs to N3N. The redirection of PDUs using the latest available information can lower the number of PDUs sent to erroneous destinations.

Figure 4.8 shows a re-enrollment example.

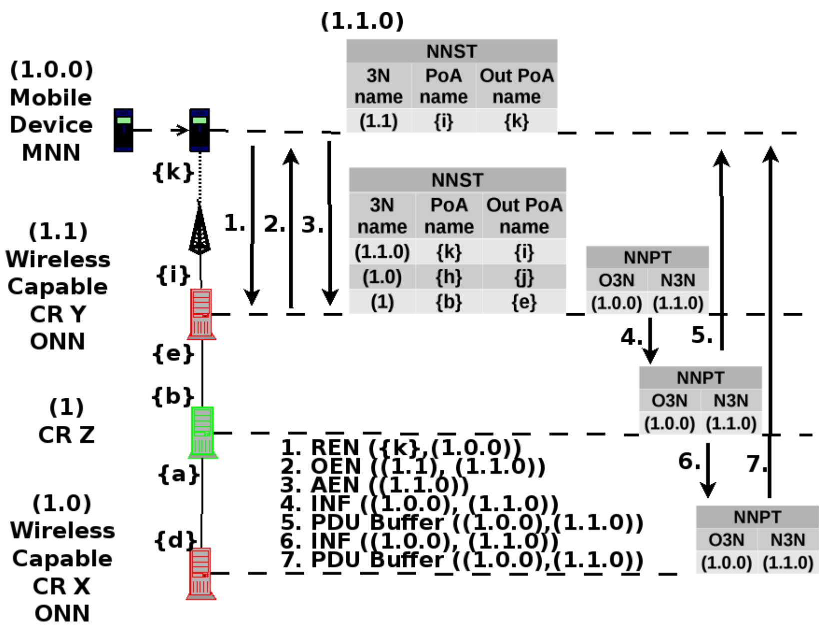

Figure 4.8: 3NA Re-enrollment flow

4.3.2 Data Transmission PDUs

Table 4.2: Data Transmission PDUs defined in the 3N Architecture

PDU	Description
SO	Uses 3N name to indicate source node
DO	Uses 3N name to indicate destination node
DU	Uses 3N names to indicate both source and destination node

Data transmission is seen as a completely separate network action in the 3NA, needing a specifically designed set of PDUs which are described in Table 4.2. The use of data transmission PDUs improves forwarding by leveraging the 3N namespace. The design of 3NA PDUs enables the interoperability of normal ICN nodes with 3NA nodes. The successful enrollment to a 3NA enabled network, or having received a SO PDU with 3N name information from another node enables the use of 3NA data transmission PDUs. SO PDUs can only be generated by 3NA enrolled nodes, as the PDU's creation requires a 3N name. However, any 3NA node, whether or not it has enrolled, can forward SO PDUs or DO PDUs. In particular DO PDUs can be generated by non-enrolled 3NA nodes if they have seen SO PDUs, as the 3N name information is aggregated in the 3NA PIT.

Upon receiving a SO PDU, a 3NA capable node checks if an Interest PDU or a Data PDU is encapsulated. On finding an Interest PDU, the PIT is searched for a corresponding PIT entry for the PDUs ICN name. If no entry is found, a new entry is created and the SO PDU's information is aggregated. If an entry is found then the SO PDU's information is aggregated. In 3NA, the information aggregated is the ingress interface and the SO PDU's 3N name. When using a simple ICN forwarding strategy, the FIB is queried to find the next interface through which to forward the PDU.

On the other hand, if a Data PDU is encapsulated, the SO PDU can only be forwarded utilizing the information available in the 3NA PIT related to the ICN name in the Data PDU. If the retrieved information contains 3N names, these names are extracted and the NNPT is checked for updates to the original aggregated 3N names. Here, the 3NA capable node can choose to generate DO PDUs to satisfy pending Interests by leveraging the NNST to forward the resulting PDU. After forwarding all the DO PDUs, should any interface in the PIT entry not be satisfied, the original SO PDU is forwarded to these interfaces.

If the 3NA PIT's entry only contains interface related information, then the standard ICN Data PDU forwarding procedure is followed.

When SO PDUs are received by 3NA capable nodes, the SO PDU's 3N name can be used to create DO PDUs. If the NNST and NNPT are maintained over time, the DO PDUs will arrive to the destination 3N name.

A detailed diagram of the SO PDU flow for ICN PDUs is shown in 4.9.

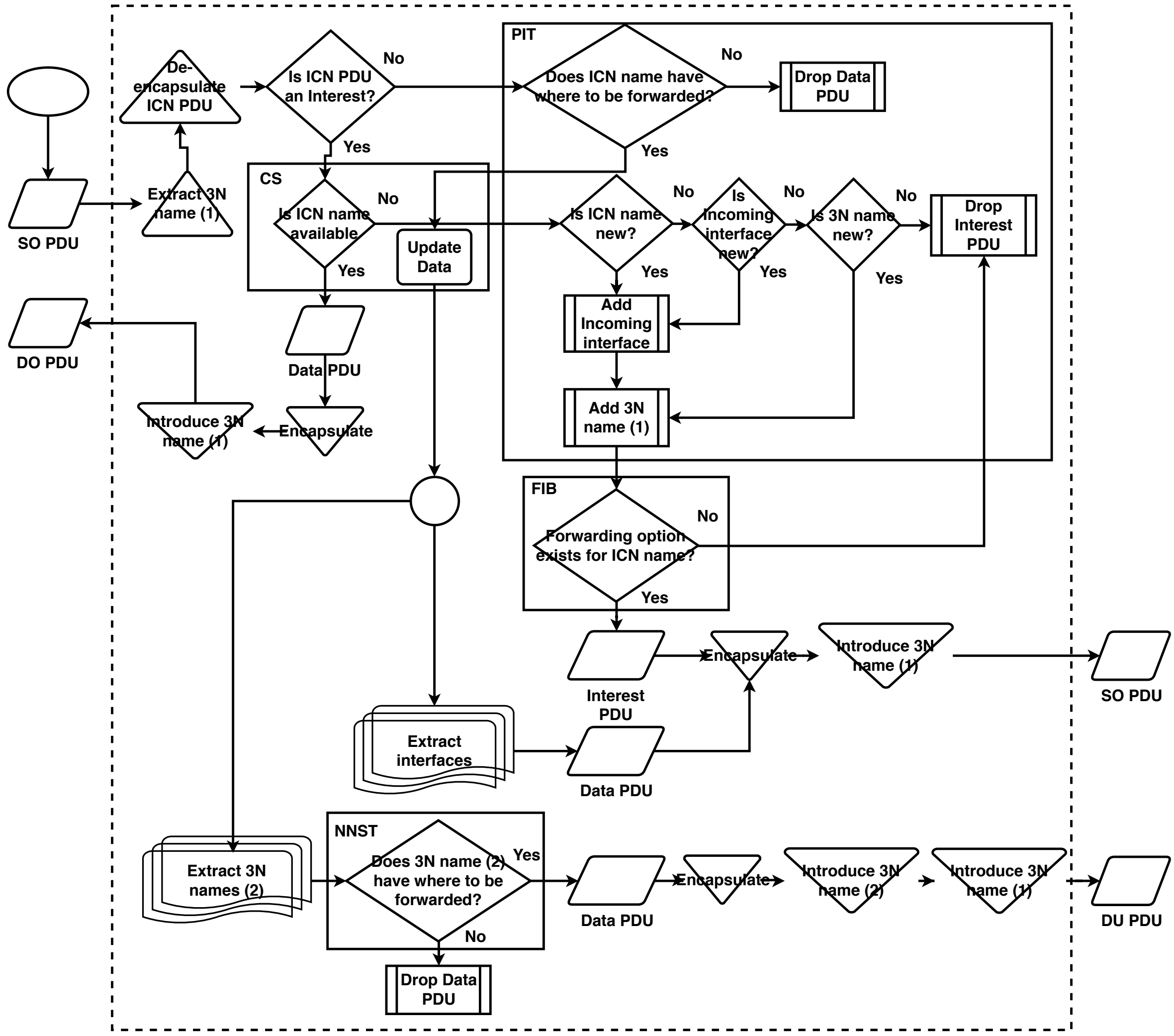

Figure 4.9: SO PDU flow for ICN PDUs

When a DO PDU reaches a 3NA capable node, if it contains an encapsulated Interest

PDU, the DO PDU's 3N name is verified against the NNPT for possible updates. Should updates exist, the DO PDU's 3N name is replaced and the NNST is queried to forward to the new destination. The Interest PDU generates a 3NA PIT entry for its ICN name to avoid future unnecessary propagations. On finding an encapsulated Data PDU, the 3NA PIT entries are checked. If 3N names are aggregated to the corresponding 3NA PIT entry, those 3N names are verified against the NNPT for updated 3N names. The resulting 3N names are utilized to generate DO PDUs which are then forwarded by consulting the NNST. The original DO PDU is forwarded utilizing information from the NNST related to the DO PDU's 3N name, even if there is no related PIT information. Once the forwarding has completed, the standard Data PDU's ICN name check against the CS is executed and the information is saved if it doesn't exist. A detailed diagram of the DO PDU flow for ICN PDUs is shown in 4.10.

When utilizing only SO and DO PDUs, if both consumer and producer are mobile, then both applications need to begin their communications using SO PDUs. The receiving side would then return the encapsulated ICN PDUs using DO PDUs with the 3N names contained in the SO PDUs. However, with any 3N name change both parties would need to wait for 3N name updates that would arrive in the next SO PDU. Particularly if the movement speed is fast, there might be a delay in receiving the latest 3N name in use. In order to reduce this delay, the 3N name changes have to be propagated quicker, allowing for PDU redirection to be utilized. In many network architectures, PDUs that contain both source and destination information are common. This type of data transmission PDU for the 3NA is created and called DU. A detailed diagram of the DU PDU flow for ICN PDUs is shown in 4.11.

The DU PDU is especially useful when one application wishes to notify the other application that the communication is mobile. Since the DU PDU could be generated by nodes with 3N names, DU PDU generation is limited to the mobile points of the communication for this document.

The use of SO PDUs from the consumer to get DO PDUs is named 3NC communica-

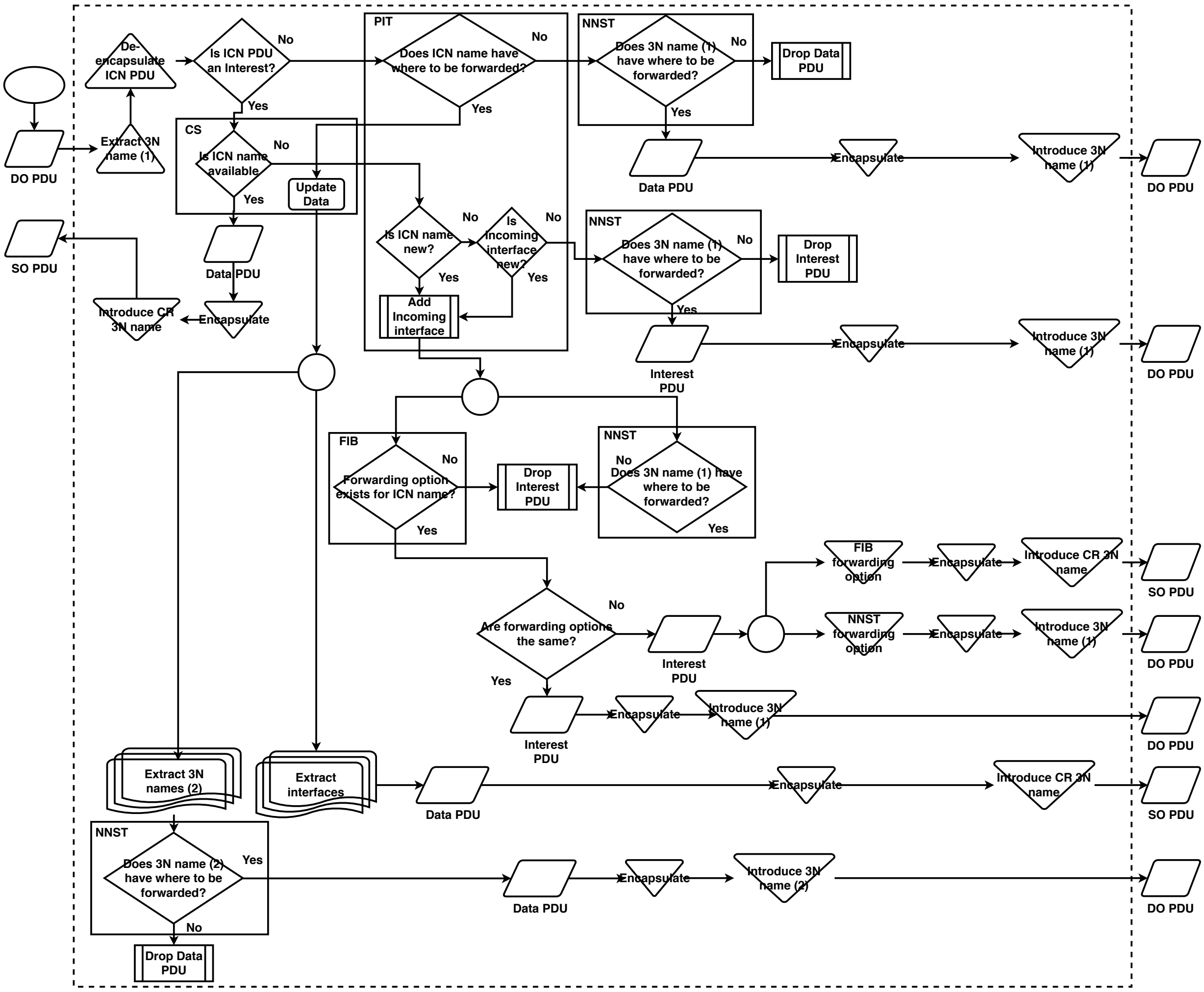

Figure 4.10: DO PDU flow for ICN PDUs

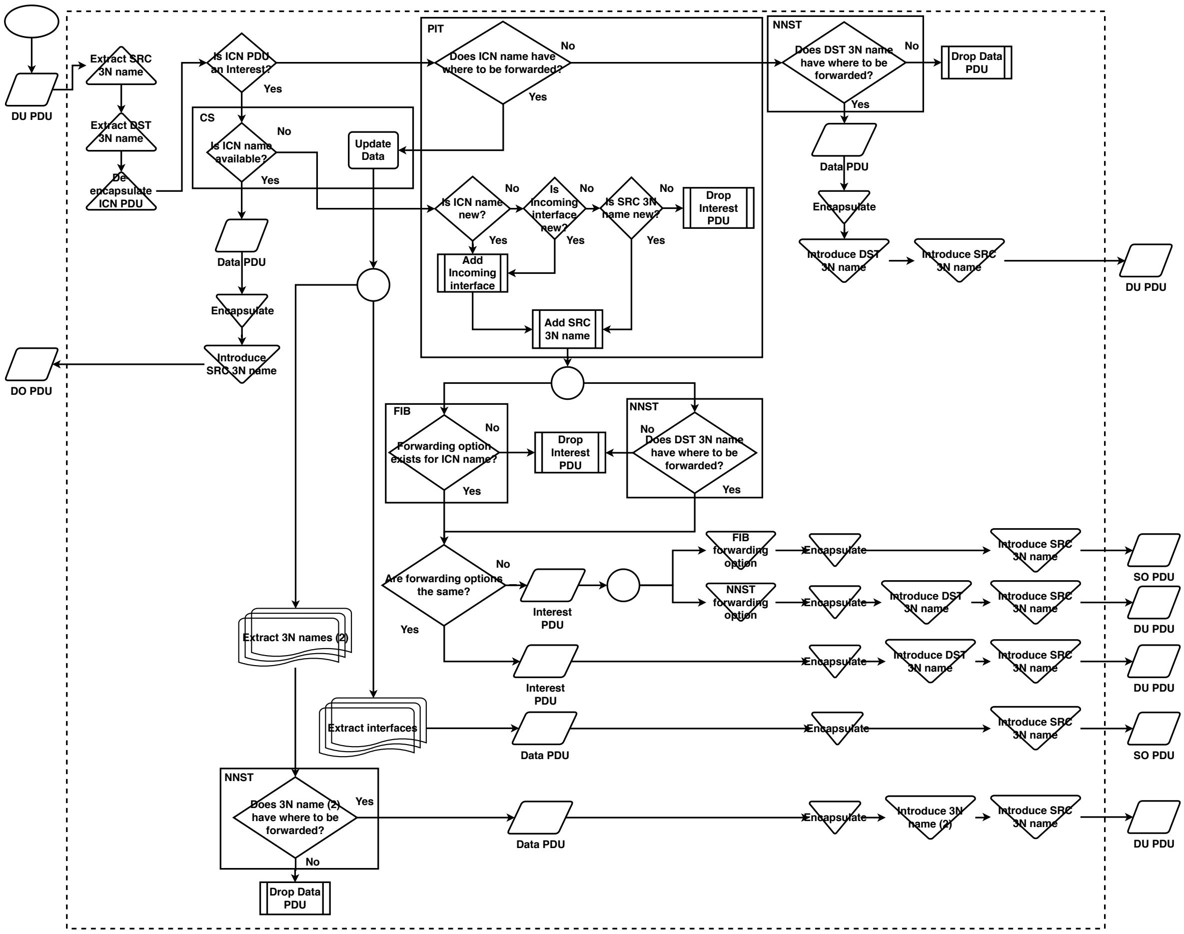

Figure 4.11: DU PDU flow for ICN PDUs

tion. The use of SO PDUs from the producer to make consumers use DO PDUs is named 3NP communication. When both consumers and producers end up using DU PDUs, then it is called 3ND communication.

4.4 Forwarding strategies

3NA relies on SF to forward Interest PDUs. Compared to a pure ICN, the 3NA gives all nodes participating in the network a 3N name, permitting the aggregation of those 3N names within the PIT. When a Data PDU is sent, a 3NA node will search the 3NA PIT entry trigger for aggregated 3N names. Before the 3N name is used in the delivery of a Data PDU, a check in the NNPT is performed to make sure that the 3N name being used is the newest available 3N name. The NNST is then used to Figure out which path to for the forwarding process by comparing the destination 3N name with the PoA to 3N mapping maintained in the structure. This search results in a PoA name and an associated interface that would best fit the requirements. The use of 3N names in the Data PDU means that in 3NA, one is routing directly on the 3N names and can leverage the mathematical properties of the namespace. The resulting forwarding strategy is called Smart Forwarding with 3N names.

4.5 Mobility

The descriptions of the mechanism PDUs in Section 4.3.1 and of the data transmission PDUs in Section 4.3.2, leave clear that mobility in 3NA depends on the mobile nodes enrolling, de-enrolling and re-enrolling to obtain and update their 3N names. Without correctly finalizing the mechanism PDU procedures, the network cannot guarantee mobility.

The procedures for 3NA network integration are designed using timers, following the protocol design of Delta-t [53]. If the timeout of the 3 main mechanisms are set in an adequate fashion, it is not complicated for communication to suffer a complete loss. This permits 3NA mobility to be initiated and finalized by the mobile nodes themselves. Permitting the mobile nodes signal to the network their movement is extremely important since it allows the network to react to node movement only when it is necessary. It also puts the burden on the mobile node to signal the necessary information for its seamless mobility, as it is only the mobile node that knows what the user intends to do.

Once the network integration procedures are successful, the updated namespace mappings provide the network connections with indirection. When data transmission PDUs with destination 3N names are seen in a DO or a DU by a node, the NNPT, described in Section 4.2, is verified to check whether the PDU's destination has to be modified. Due to the header of the data transmission PDUs being completely decoupled from the ICN PDUs, should the 3NA related information be modified, there will be no effect on the ICN related information. Once modified, the PDU has a higher probability of reaching the intended location. This is because PDUs, even those en route, will not need to get all the way to the initial ONN before being redirected to the correct location by intermediate nodes with updated NNPT information.

One key feature of 3NA is that mobility is not considered a service within the network architecture. The signaling provided by the INF mechanism PDU is only considered for network flows with long Round Trip Times (RTTs), but mobility can work just as well without out. Mobility comes from naming the nodes in a orderly and managed fashion. The mechanism PDUs that enable enrollment, dis-enrollment and re-enrollment can be handled by any node that is part of 3NA, because a network is a distributed system that requires the cooperation of all nodes that wish to be part of the network. This means that there is no specialized server to implement in a 3NA network.

A brief example of the flow of mechanism and data PDUs when mobility takes place is described in Section 4.6.

4.6 Mechanism and data transmission example

This explanation refers to Figure 4.6. At the beginning of the scenario, X has 3N name (1.0), Y has (1.1) and Z has (1). The NNST tables of all nodes contain the neighboring node's PoA to 3N name mappings.

When in range of ONN X's wireless connection, the mobile device can enroll and be offered the 3N name (1.0.0). After the mobile device receives the offer for (1.0.0), the node can encapsulate all ICN PDUs in SO PDUs from that point onwards. Under normal circumstances when the mobile device sends a SO PDU with the 3N name (1.0.0) containing a Interest PDU, the PDU would be transmitted to node X using the 3N name (1.0). Upon X receiving the SO PDU, the node would aggregate the ingress interface and the 3N name (1.0.0) in its PIT after creating an entry related to the Interest PDU's ICN name. The Interest PDU's ICN name would be searched in the CS and FIB. As no information is available in the CS or FIB, the PDU would eventually be forwarded to node (1). A similar process would occur in every node until the PDU reaches the producer.

On the node with 3N name (1.0.0) on which the producer is executing, information obtained from the SO PDU, can be used to create an DO PDU. The producer can also create a DU PDU, if it wants to include its own 3N name (1.2) information. In either case, the resulting Data PDU for the request in the Interest PDU is encapsulated in the utilized 3NA data transmission PDU. The PIT-based ICN PDU forwarding specified in [23] is followed for 3NA. The moment when this procedure is not followed is when the mobile device changes its network point of attachment.

When the mobile device decides, for whatever reason, to change its point of attachment to the wired network by selecting ONN node Y, the dis-enroll procedure is initialized by the mobile device. If DEN mechanism PDUs in the dis-enroll procedure are being used with the PDU buffer creation option turned on, a PDU buffer is created for the mobile device's 3N name, (1.0.0), on ONN node X. As node Z with 3N name (1) is the parent of node Y with 3N name (1.0), the DEN mechanism PDU is forwarded to node Z where a PDU buffer for (1.0.0) is also installed. Upon confirmation of a successful dis-enroll from ONN node `X`, the re-enroll procedure is initiated on ONN node `Y`. An offer for (1.1.0) is received by the mobile device. An INF PDU is sent to ONN node `X` from ONN node `Y` utilizing the paths connected to node `Z` once the re-enrollment procedure is completed. When the nodes see and INF PDU, their NNPTs are updated, leaving a mapping from (1.0.0) to (1.1.0) and at the same time emptying possible buffers created by DEN mechanism PDUs.

If for whatever reason, the producer does not become aware of the 3N name update of the mobile device, the producer may continue using the old 3N name (1.0.0). Since PIT-based forwarding is still available on all nodes, the producer could even opt to not send the related Data PDUs encapsulated in 3NA data transmission PDUs. For this case, the 3N names aggregated in the PIT of intermediate nodes would be used to deliver the Data PDU to the mobile node. In any case, the node that receives the PDU will verify the 3N name to be used against the NNPT. Since the NNPT was updated, the PDU destination would be rewritten and redirected to the mobile device utilizing 3N name (1.1.0). This quick redirection would minimize delay, PDU loss and increase application goodput. Although the use of intermediate NNPT tables would maintain the delivery of PDUs, it is expected that the producer would at some point obtain an Interest PDU encapsulated in a SO or DU PDU containing the mobile device's new 3N name. When the producer obtains this information, the process detailed above would begin once again until another re-enrollment related event occurred.

4.7 Improvements derived from the developed questionnaire

Section 3.4 has shown that question 1 and question 2 from Section 2.4 are answered negatively in the case of the original ICN architecture. Therefore modifications are proposed that permit an ICN architecture to answer to question 1 positively as detailed in Section

4.7.1. The previous modifications can be leveraged to answer question 1 positively, by integrating these to 3NA to be able to answer positively to question 2. The details of the integration with 3NA are detailed in Section 4.7.2. Table 4.3 summarizes the questionnaire results for all the forwarding strategies and networks utilized in this article.

Table 4.3: Questionnaire evaluation results for ICN forwarding strategies

Strategy	Question 1	Question 2	Question 3
Flooding	×	×	×
SF	×	×	◯
SF MM	×	×	△
SF MM PR	◯	×	◯
SF 3NC	×	△	◯
SF 3NP	×	△	◯
SF 3ND	×	△	◯
SF MM PR 3N	◯	◯	◯

◯ = Satisfied, △ = Partially satisfied, × = Unsatisfied

4.7.1 Smart Flooding with MAP-Me and Part Routing (SF MM PR)

Aggregation of ICN names is extremely important for PIT and FIB memory usage. For ICN names to be highly aggregatable, the names are suggested to have a hierarchical format. Knowing this, one can strengthen the ICN naming rules to create a tree structure. $\varnothing$ would be the root of the tree and the beginning point for the construction of any ICN name. Each edge would add a '/' to the final name when traversed. This article does not limit the valid strings for vertices in the ICN name graph. A complete ICN name graph is shown in Figure 4.12. It is known from graph theory that one can define in graphs a distance $d(u, v)$, with u and v being graph vertices and the distance being defined as the minimum number of edges needed to be traversed to get from vertex u to v. Defining a distance, as was explained in Section 2.4, permits the definition of a metric space over the graph to which u and v belong.

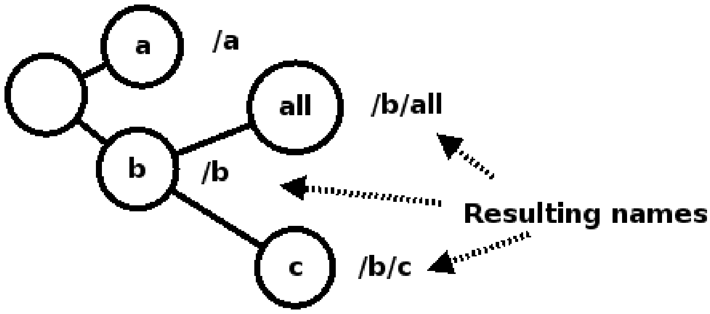

Figure 4.12: A graph for ICN names

Figure 4.12 shows that ICN names can be easily mapped to a graph with the above procedure. However, as was also mentioned in Section 2.4, the information mapped to an ICN name would be divided into chunks in order to transmit the information across the network. This means that the ICN name search, although important, is not as important as finding the chunks to which an ICN name responds to. Adding a numbered label to the chunks to identify them is a simple procedure. These identified and related chunks of data to an ICN name are called Parts. Having labeled Parts with sequential numbers permits the assignation of the one-dimensional Euclidean distance on the space. This permits the definition of '$\geq$' and '$<$'. These definitions can be later utilized to aggregate the Parts present in a CS by defining intervals.

MAP-Me enhancements can easily make use of this new information. The MAP-Me-IU, before being transmitted by the producer, would create a new field to add the next ICN name Part p_j which would be transmitted to the wired network. The use of the MAP-Me-IU with this information would pass the ingress interface to a green state for only the Parts that satisfied $\geq p_j$. This modification of MAP-Me-UI is called Part Routing (PR). SF MM PR is the abbreviated forwarding strategy when Smart-Flooding, MAP-Me enhancements and Part Routing are used together. The use of this forwarding strategy would respond to question 1 and 3 positively.

4.7.2 Smart Flooding with MAP-Me, Part Routing and 3N names (SF MM PR 3N)

3NA, as defined in Section 4.1.1, does not map ICN names to 3N names. Utilizing the 3NA as is, would also return a negative answer for question 2. However, 3NA also attempts to have the names used by node when enrolling into the network meet topological requirements. Although having every ICN name have a defined ground could be a laborious task, leveraging 3NA to define a ground is a positive step forwards. Figure 4.13 displays an interpretation of the graph to describe 3N name construction. In the graph, each vertex is labeled with a number. A '.' is added to the final name each time an edge is traversed. Using the same logic as the tree graph built in Section 4.7.1, one can define the graph distance metric. This allows the 3N namespace to be considered a metric space.

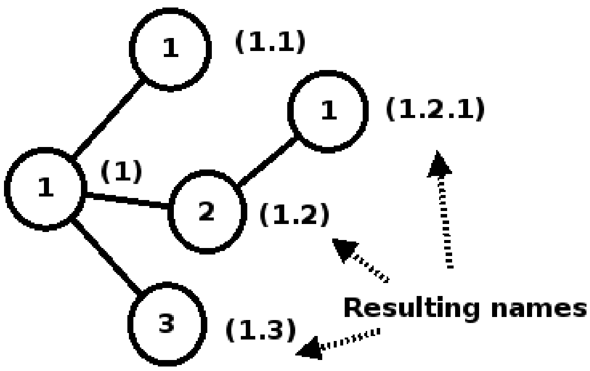

Figure 4.13: A graph for 3N names

Mapping every ICN name to a 3N name could become an endless task. One can reduce the burden by taking advantage of the changes to the ICN namespace in Section 4.7.1. Using this approach, ICN names can be aggregated by the '/' name construction and by the use of Part intervals. Part intervals would then be mapped to 3N names in order to ease the number of Parts that would have to be managed in the PIT and FIB.

The same as in Section 4.7.1, the MAP-Me-IU can be used to update FIB information. In this scenario, the MAP-ME-IU would additionally map a ICN Part interval directly to a 3N name.

One very important difference in 3N name utilization has to be considered. To satisfy question 2, 3N names must belong to a single namespace to which all ICN name can be mapped. Due to the constant changing of 3N names for mobile devices, the implication is that ICN names should map to 3N names that have a large lifespan within the network. In the particular case of push producers, it is better for the MAP-Me-IU PDU to use the 3N name not of the producer but of the node to which the producer is connected. In this article, the discussion is centered around this type of 3N name usage for MAP-Me-IU PDUs. The definition of place specification described in Section 2.2 is satisfied when mapping a Parts interval to a 3N name. Due to 3N names being able to be described in a tree graph, the 3N namespace can also be considered a metric space. With the mapping established, question 2 can be answered positively.

Of particular interest is that as long as mobile nodes use all the mechanism PDUs correctly, then PDUs utilizing 3N names have a higher chance of arriving at their desired destinations. This is an advantage of using 3N names, particularly when forwarding Data PDUs. This is because the transmission of Data PDUs when using 3N names is not limited to the ingress interface data available in the PIT. 3NA also keeps the NNPT up to date, which can be used to redirect PDUs to their final destinations.

SF MM PR 3N is the abbreviation used when Smart-Flooding, MAP-Me enhancement and Part Routing are used with ICN names being mapped to 3N names. The mapping is maintained with the update mechanisms available in 3NA and leveraged in all 3NA structures. Under this configuration, the forwarding strategy can answer the whole questionnaire positively.

Chapter 5

Evaluation

The results of the network naming evaluations in chapter 3 gave insight into what improvements could be made to the network architectures to improve network naming. Additionally, to be able to evaluate both sets of forwarding strategies, it was necessary to create a framework. The framework offered a simple portable way of measuring the intended performance metrics while letting the focus of the research be on network naming. It is also important to remember that in order to cover the many use cases for communication in the improved ICN based network architecture, the communication methods available in 3NA also need to be tested.

Section 5.1 describes the scenario along with the parameters that were common during our research. Section 5.2 discusses the evaluation of 3NA data transmission PDUs for mobile producers and mobile producers with mobile consumers. A discussion of the performance metrics for this evaluation is written within the section. Section 5.3 describes the evaluation of 3NA for the push producer scenario and the performance metrics utilized.

5.1 Mobility framework

A lot of work has been done to attempt to model network graphs that represent current networks [31] [54] [44]. One particular interesting attempt is to collect current network data by using common network tools to decipher the underlying network graph, resulting in a collection of representative network graphs to use in simulations [37]. At the lower scale of networks, such as edge cases, the models and the collections of network graphs is much smaller and more diverse. Since what required testing in this document was mostly the naming aspect of networks, no previously made scenario offered the benefits and flexibility required to test the ideas presented. It became necessary to build a mobility framework that could be reused in the simulator framework while being flexible enough for us to be able to compare results without a lot of effort.

The results are the mobility frameworks which were described in [1] and [2].

To assess the impact of using a correct naming structure can have on network mobility, nnnSIM [30], an implementation of 3NA, which is installed as a module in the ns-3 network simulator framework [35] was leveraged. This mobility framework consists of a single Autonomous System (AS) Graph of a Network (GN) named the Intra-AS scenario. This scenario includes 7 wireless access points (WAPs). The wireless access points are connected in a hierarchy creating a graph that is shown in Figure 5.1. The WAPs are located 215 meters apart on points that create the perimeter of a regular hexagon. Each WAP has its own SSID and utilizes the ns-3 framework's default WiFi implementation[1]. ICN CRs are connected via wired connections.

Note that 3NA capabilities derived from using 3N names are only enabled when the consumer node uses these capabilities. The network nodes obtain 3N names from the central WAP. When the network utilizes 3NA functions, all 3NA related buffering of PDUs, particularly during node name changes, network enrollment, dis-enrollment and re-enrollment, are turned off to be able to compare forwarding strategies. When a consumer does not use 3NA functionalities, the wired network nodes act as regular ICN CRs. All the network parameters used in the simulations are summarized in Table 5.1. For simplicity, there is only one consumer and one producer in the scenario. Both the consumer and the producer have a single interface to connect to the WAPs. These nodes have no other

[1]The default setting used are for ns-3.29

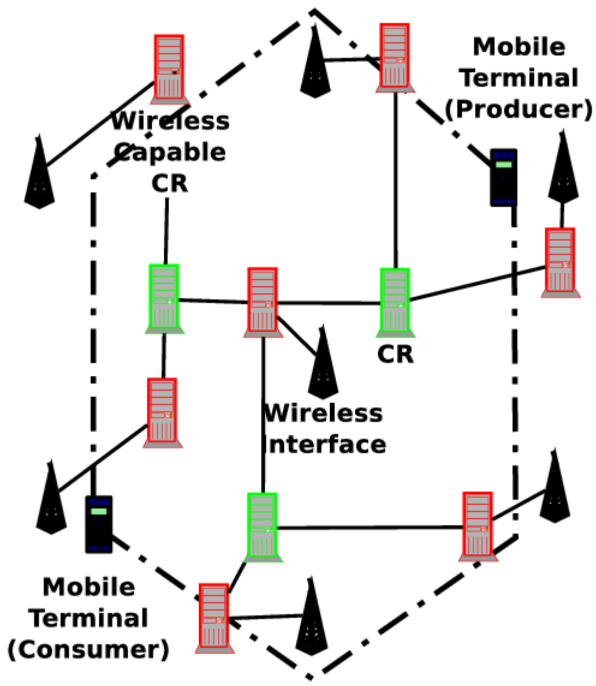

Figure 5.1: Network graph of the mobility framework

interfaces. The producer and consumer cannot cache ICN information as they have no CSs.

A simplified handover procedure for mobile nodes was chosen. When a node travels

Table 5.1: Common parameters for the mobility framework

Parameter	Value
Interest retransmission timeout	$50ms$
Number of Interest retransmission attempts	4
PIT Entry Timeout	$1s$
Capacity of the Content Stores	10 million objects
Consumer start time	$20s$ after producer
Lease time for 3N names	$300s$
Timeout for 3N Buffers	$165s$
Link delay (wired)	$5ms$
Bandwidth / Link capacity (wired)	100Mbps
Producer and consumer velocities	1.4, 2.8, 5.6, 7, 8.4, 11.2, 12.6, 14.0 m/s
Video bit rate	2500`kbit`$/s$
Generation rate for Interest PDUs	306 `PDUs`$/s$
Link properties (wireless)	Nakagami Propagation Constant Speed Propagation Three Log Distance Propagation
Simulation time	$1000s$

100 meters, it checks the RSSI signal strength of the neighboring WAPs. The nodes automatically migrate to the WAP with the best signal at that point in time. When using 3NA, migration means executing signaling procedures before and after changing WAP. The whole handover process performs appropriately during the simulation. The signaling procedures, managed by 3NA mechanism PDUs update the NNST, NNPT and 3N related buffers [1].

Within the simulation, mobile nodes always exhibit a constant velocity. The nodes' flow creates a hexagon. The hexagon is regular with each side being 230 meters. The resulting perimeter is shown in Figure 5.1. The cyclical movement of the nodes, permit the comparison of speed using the different forwarding strategies.

In 2016 the second largest data streaming rate for video on the Internet was 2400kbit/s. Due to PDU overhead, the rate of PDUs was increased to 2500kbit/s. This value was used within the framework for the experiments.

5.1.1 Performance analysis

3NA's overall mobility performance is analyzed from the from the perspective of the consumer. For the mobile producer with mobile consumer, the performance metrics evaluated are described below.

- **Consumer's mean delay**: Quality of Experience (QoE) generally requires a low overall delay for streamed content.

- **Consumer's mean goodput**: Goodput, if correctly defined, is able to summarize network aspects into one value [12]. For applications that require a constant level of input for a high QoE, such as video streaming, goodput performance is an interesting metric because it permits one to know whether the consumer has enough information to correctly display the information within a given time frame. For the mobility framework, the goodput percentage is compared to the ideal goodput. The formula shown in equation 5.1 is used to calculate this metric.

$$\text{Goodput} = \frac{\substack{\text{RX's received \# of} \\ \text{in sequence PDUs} \times \text{Capacity of PDU payload}}}{\text{Interval of time}} \tag{5.1}$$

- **Network timed out Interests to satisfied Consumer Interests ratio**: Interest timeout in ICN means that the network can track a consumer's request by using the PIT to remove Interest request that cannot be satisfied after a specified timeout period. If a forwarding strategy has a low network Interest time outs to satisfied consumer Interests, it means that a generated Interest is generally locating the requested Data. This also implies that the consumer does not put a considerable burden on the whole network by forwarding the Interest via unnecessary interfaces.

5.2 Mobile producer with mobile consumer

3NA offers three different data transmission PDUs. Counting the two ICN methods described in Section 2.5.1, 3NA has five different ways to establish communication between two network applications. The communication methods which are of interest for the mobile producer with mobile consumer scenario are detailed below.

- **Flooding**: Flooding is the ICN forwarding strategy described in Section 2.5.1.1.

- **Smart-Flooding (SF)**: Implemented in ndnSIM [2], the ICN model for the ns-3 simulator, this stateful forwarding strategy has been widely used in the ICN literature. The strategy was also detailed in Section 2.5.1.2.

- **Consumer using SOs (SF 3NC)**: In this communication method the consumer uses SO PDUs. This allows the producer to return the solicited information in DO PDUs. Smart Flooding is used for the initial search for the producer.

- **Producer using SOs (SF 3NP)**: In this communication method, the producer returns any solicited information in SO PDUs. When a consumer obtains the producer's 3N name from the SO PDU, along with a correct Data PDU, the consumer can then begin to send Interest PDUs in DO PDUs. Smart Flooding is also used for the initial search for the producer.

- **Producer and consumer using DUs (SF 3ND)**: In this communication method, consumers use SO PDUs when transmitting Interest PDUs. This transmits the consumer's 3N name to the producer. The producer, also having a 3N name, can then transmit the corresponding Data PDU in a DU PDU with the producer's 3N name. Once an initial DU PDU carrying a correct Data PDU has been received by the consumer, the consumer can opt to begin using DU PDUs for future Interest PDUs. Smart Flooding is again used for the initial search for the producer.

This scenario focuses on having the producer and consumer mobile within the mobility framework described in Section 5.1.

5.3 Mobile push producer with mobile consumer

For this scenario, the producer node is an 3NA enabled edge pushing content node. This means that the producer has a 3N name and continuously pushes it's content to the WAP to which it is connected. The producer does not respond to any requests for content from the network. The use of a 3N name enables us to push content without having to modify NDN's Interest/Data PDU protocol flow as established in [57]. In this scenario, the consumer moves as described in Section 5.1.

The four communication methods of interest are shown below:

- **Flooding**: Flooding is the ICN forwarding strategy described in Section 2.5.1.1.

- **Smart-Flooding (SF)**: Section 2.5.1.2 details this stateful forwarding strategy. 2.5.1.2.

- **Smart Flooding with MAP-Me and Part Routing (SF MM PR)**: Section 4.7.1 details this forwarding strategy which gives names to ICN name Parts within a metric space.

- **Smart Flooding with MAP-Me, Part Routing and 3N names (SF MM PR 3N)**: This is the forwarding strategy detailed in Section 4.7.2 which gives names to ICN name Parts within a metric space and then maps the Parts intervals to 3N names.

Chapter 6

Results

Utilizing the mobility framework described in Section 5.1, we focus on producer mobility when the consumer is mobile as described in Section 5.2. The results for this experiment is found in Section 6.1. Section 6.2 utilizes the parameters from Section 5.3 for the experiments and discusses the results. For all scenarios the Flooding forwarding strategy is separated from the other forwarding strategies in order to have a clear picture of the absolute worst case scenario.

6.1 Mobile producer with mobile consumer

The mean network delay for Flooding is shown in Figure 6.1. The average delay is under 30 seconds which is extremely poor performance. The delay doesn't seem to go past 31 seconds, event at the highest node speed, which is interesting, but conveys the uselessness of finding a mobile produce utilizing only flooding. The mean goodput compared to the ideal goodput of 2500kbit/s for Flooding is shown in Figure 6.2. The graphs has it's best goodput at 8.7% but doesn't at any point in the experiment pass this value. The final graph related to Flooding is the ratio of network timed out Interests vs satisfied consumer Interests and is shown in Figure 6.3. The results are somewhat expected as in the framework there are only 6 possible paths to take from the center. It is interesting to note that the network timed out Interest vs satisfied consumer Interest ratio has it's lowest value just above 5 : 1 when the nodes are moving at $4.2m/s$. It is clear from the results that the Flooding strategy is not suited to connected to a mobile producer within the same network sector.

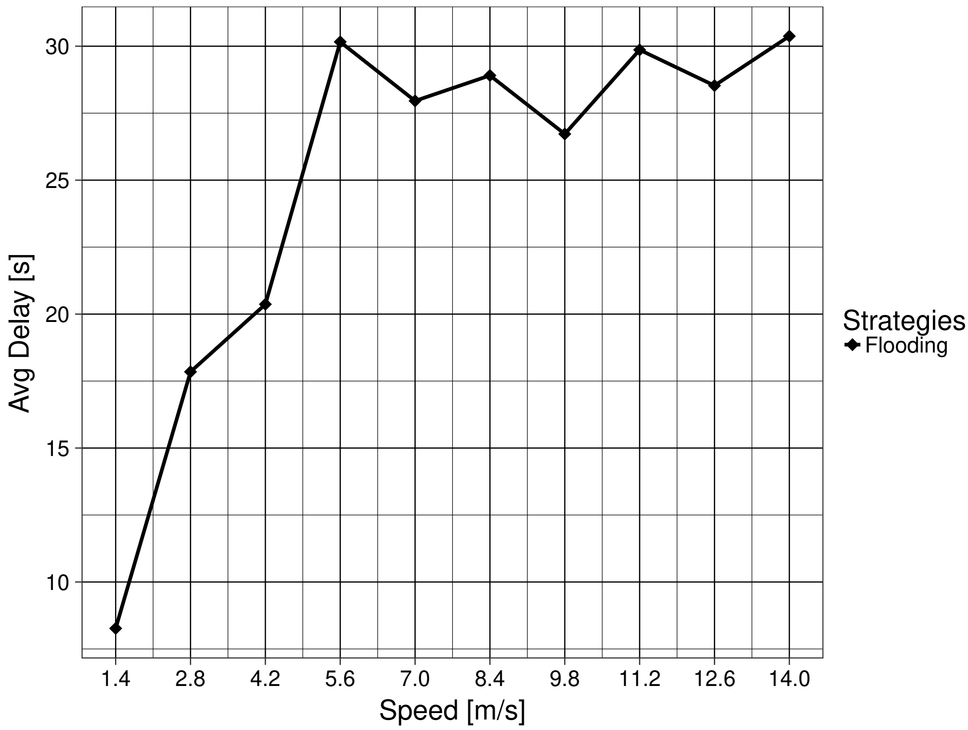

Figure 6.1: Mean network delay for mobile consumer and producer using Flooding forwarding strategy

The mean network delay for SF, SF MM, SF 3NC, SF 3NP and SF 3ND are shown in Figure 6.4. These results are mixed. The delay for all forwarding strategies is below 1.5 seconds and is stable for all forwarding strategies regardless of the mobile node speed. The worst performing forwarding strategy is SF 3NC, followed by SF, SF 3NP, SF MM and finally SF 3ND. SF 3ND outperforms the SF MM by over 0.3 seconds.

The mean goodput compared to the ideal goodput of 2500kbit/s for the four forwarding strategies are shown in Figure 6.5. The results are extremely stable, obtaining similar mean goodput regardless of the mobile node speed. The most interesting result of this experiment is that SF, SF MM, SF 3NC and SF 3NP has a goodput that is below 30% of the ideal, while SF 3ND is above 80%. The order, from worse to best goodput is SF,

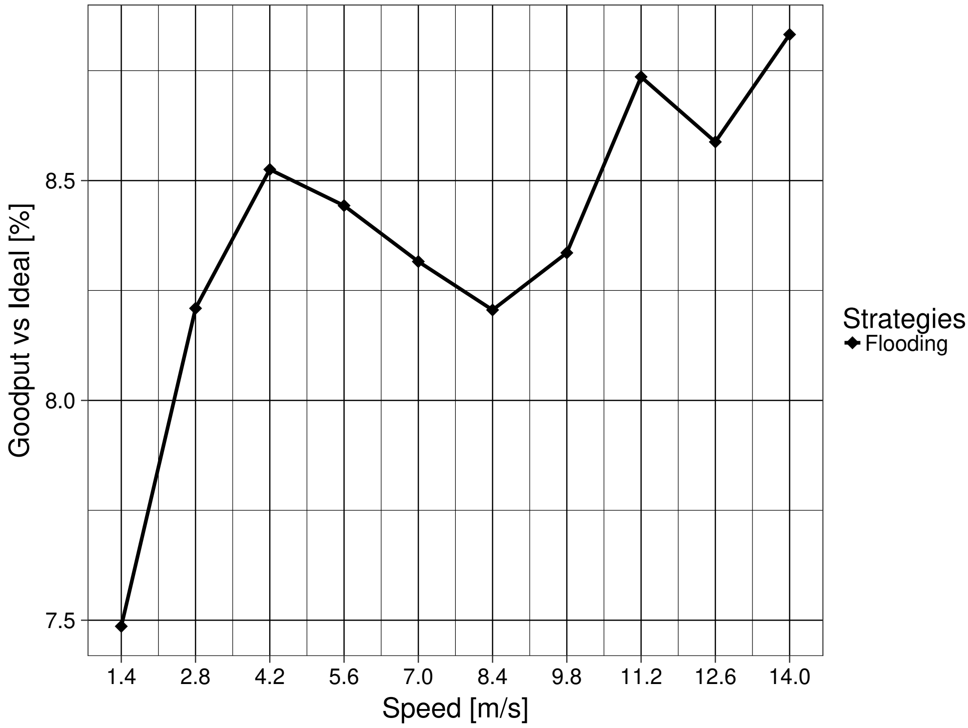

Figure 6.2: Mean goodput percentage of ideal for mobile consumer and producer using Flooding forwarding strategy

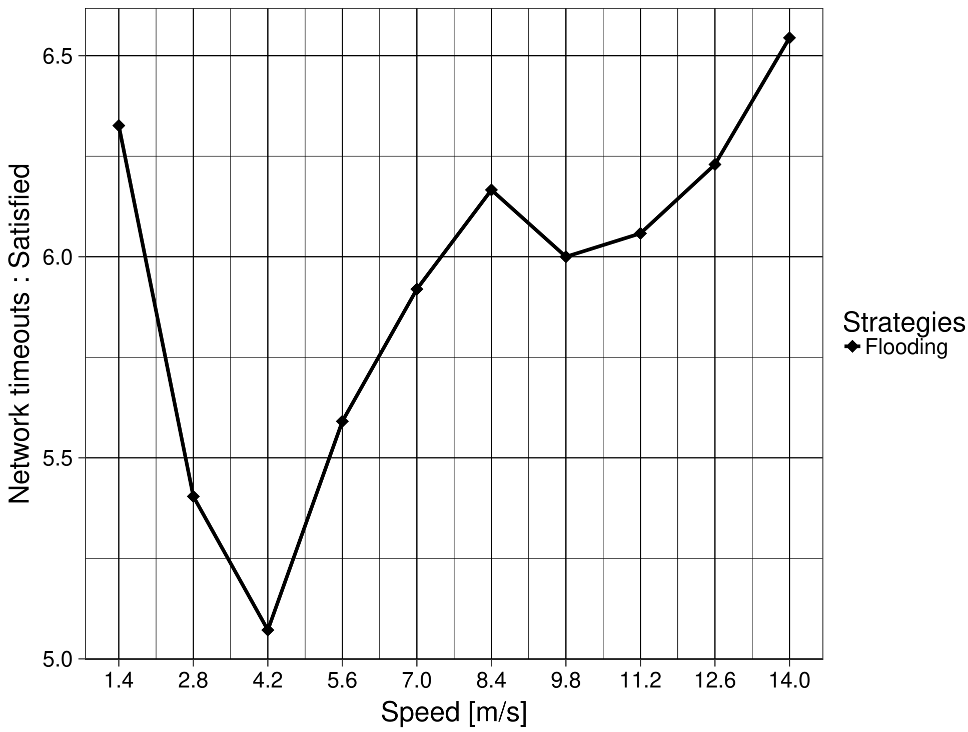

Figure 6.3: Timed out Interests to satisfied consumer Interests ratio for mobile consumer and producer using Flooding forwarding strategy

SF MM, SF 3NC, SF 3NP and SF 3ND. Although there is a clear difference between the forwarding strategies other than SF 3ND, the difference is minimal. SF 3ND clearly outperforms all the other forwarding strategies by a large margin.

The ratio of network timed out Interests vs satisfied consumer Interests for the five forwarding strategies are shown in Figure 6.6. Here SF 3ND outperforms all the other forwarding strategies by obtaining a stable ratio that is close to 0.1. SF 3NP, is not very far behind but clearly has a slightly higher network timed out ratio than SF 3ND. SF MM has a ratio that is always above 2 which increases as the mobile node speed gets higher to over 3. This forwarding strategy is the only one that sees a worsening of the ratio as the mobile nodes increase speed. SF 3NC and SF obtain ratios above 4, but stay below 4.6, also showing a slight tendency to worsen as mobile node speed increases.

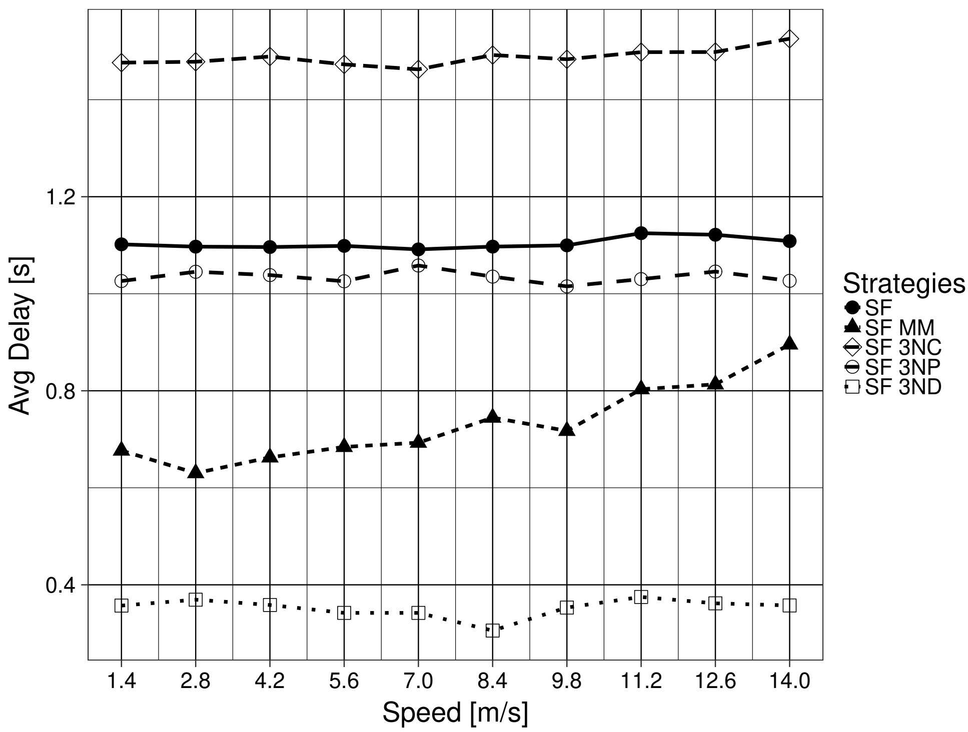

Figure 6.4: Mean network delay for consumer for mobile consumer and producer using forwarding strategies

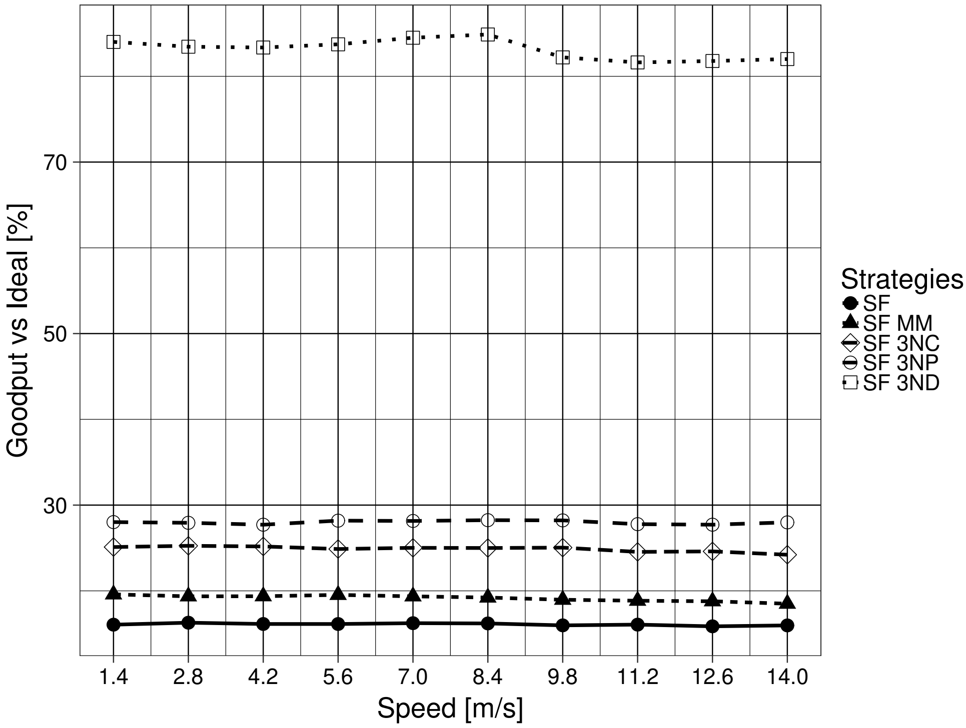

Figure 6.5: Mean goodput percentage of ideal for mobile consumer and producer using forwarding strategies

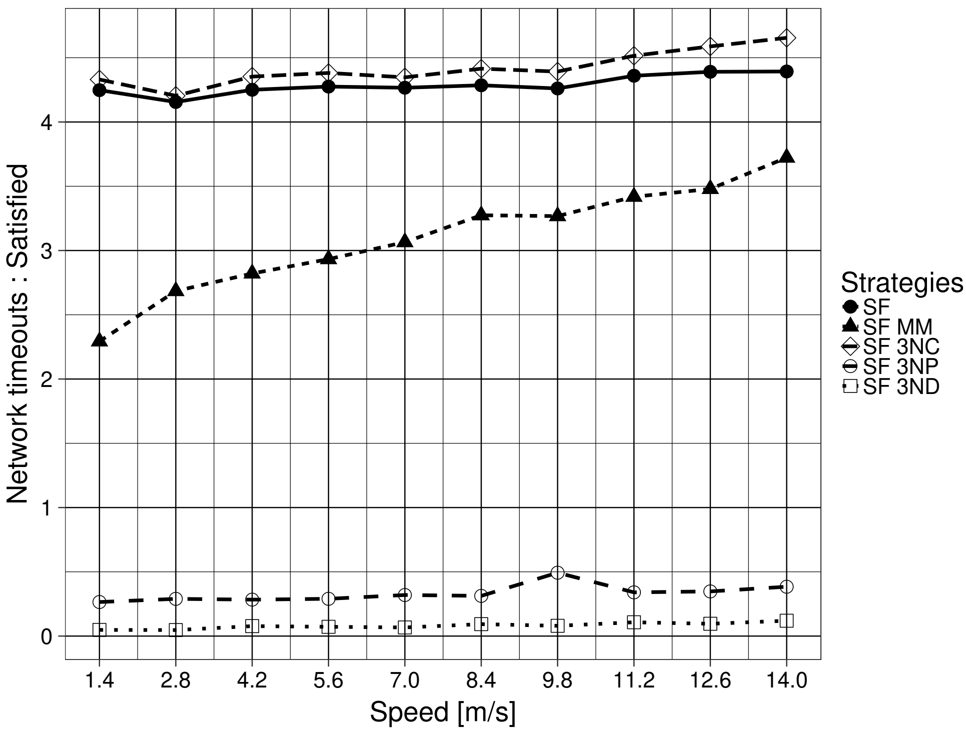

Figure 6.6: Timed out Interests to satisfied consumer Interests ratio for mobile consumer and producer using forwarding strategies

6.1.1 Discussion

The information that the consumer is requesting is initially always at the producer. Leaving aside retransmitted PDUs and the cache system present in ICN architectures, this means that for all intents and purposes, the requested ICN name, our figure F, is at the node, our ground G. This is true for the producer and is also true for the consumer, whose application, that has no name in the ICN architecture, is also at a fixed node. For the communication methods that do not use 3N names, although the G may be the node, the only identifiable network component is the network interface. This interface is completely local. The problem here is that, although the interface on the node doesn't change because the node only has one interface, the connecting interface to reach the node's interface changes whenever the node moves. Both the consumer and the producer are moving at the same speed, meaning that both the consumer and the producer's interface connection to the wired network is always changing at the same rate. Without any other way to redirect PDUs when the producer or the consumer changes the interface to which their interface connects, a PDU is trapped in the initially created FIB/PIT chain, ensuring that a lot of PDUs will never reach their intended targets. From a place specification perspective, what is happening is that the interface, the G, being picked when heading to the consumer or the producer is always slightly outdated.

In the ICN architecture, the MapMe implementation, described in Section 2.5.2, was created specifically to update the intermediate interfaces so that the route to the producer would not be so outdated. This fits in very well with the idea proposed in question 3. As the results in the previous section show, there is in fact a reduction in delay, and the network timed-out Interest to consumer satisfied Interest ratio. More importantly, there is a slight increase in overall throughout, signaling that consumer Interests seem to be reaching the producer and returning to the consumer in a more efficient manner than with Flooding of SF. The use of MapMe also decreases the mean network delay and the network timed-out Interests to consumer satisfied Interest ratio when compared to SF.

What is interesting, is that the improvement is not higher in our simulations.

The 3N name based communication methods are extremely interesting. SF 3NC, which attempts to ensure the reachability of the consumer outperforms SF and even SF MM. This implies that in ICN, there is some benefit to tracking consumers for performance purposes. There is an understanding that this may be seen as a trade-off, particularly when anonymity is desired. The results for delay also point to there being an unintended cost in increased delay as information might have to travel longer, particularly when taking into account the actions of the DEN and INF mechanism PDUs from Section 4.3.1. SF 3NP further improves the goodput, and drastically improves the network timed-out Interests to consumer satisfied Interest ratio with only minor gains to mean delay when compared to SF. Again the improvement in goodput is not big, bordering 30% of the ideal goodput, but this result seems to imply that arriving at the mobile producer positively affects overall overall goodput and delay.

This makes sense when one remembers that any information that is sent to attempt to satisfy an Interest will be cached within the wired network. However, if the producer is never reached, then there is no information to be cached or information sent.

SF 3ND far outperforms all the forwarding strategies, having the best performance of all forwarding strategies tested, regardless of node speeds. What is happening in this particular scenario is that ICN names are not being used to find the consumer or the producer, except for the first initial contact. The Interest request is still used and should the network have that particular ICN name related information available it can quickly respond to the petition. However, the search for the producer and the consumer is utilizing the the 3N namespace, a namespace that can be made to fit the definition of a metric space (Section 4.7.2) and the search for a ICN name and a consumer can be simplified to what 3N name is the node using. When using SF 3ND for scenarios with mobile consumers and producers, the questionnaire in fact can be answered positively for question 1, 2 and 3 because we are completely avoiding the use of the ICN namespace. Remember that the frames of reference proposed by Levinson (Section 2.2.2) do not limit the definition of F and G to be in different namespaces, making this sort of technicality extremely interesting.

6.2 Mobile push producer with mobile consumer

The mean network delay for Flooding is shown in Figure 6.7. The average delay is always under 0.5 seconds, which, taking into consideration how Flooding works, does not seem extremely bad. When one looks at the goodput, it becomes clear that Flooding is not working as well as it could be. The mean goodput compared to the ideal goodput of 2500kbit/s for Flooding is shown in Figure 6.8. The graphs has it's best goodput at 59% of the ideal with all other results at a lower rate. What is of interest is that at highest speed tests, Flooding still obtained results above 53% of the ideal. The final graph related to Flooding is the ratio of network timed out Interests vs satisfied consumer Interests and is shown in figure 6.9. It should come as no surprise that in order to get one satisfied PDU, the forwarding mechanisms introduces more than 5 timeouts. Since Flooding has no memory of where it previously decided, it uses all 6 branches of the mobility scenario's hexagon to attempt to search for the producer. Although a goodput of more than 50% may be acceptable, having all branches of our mobility framework occupied is unacceptable. Flooding demonstrates that a network naming scheme that doesn't follow the questionnaire from 2.4 is completely possible with some trade-offs, such as higher overall bandwidth usage and a high ratio for timeouts.

The mean network delay for SF, SF MM, SF MM PR and SF MM PR 3N are shown in Figure 6.10. For these results, there is a clear difference between making a network naming scheme answer positively to question 1. Although the difference between SF and SF MM (question 1 = $\times$) and SF MM PR and SF MM PR 3N (question 1 = $\bigcirc$), is not large before the nodes start moving below $5.6m/s$, the delay increases by more than double afterwards. SF MM PR 3N outperforms all other strategies. The difference in delay between SF and SF MM is negligible until the nodes start moving faster. The

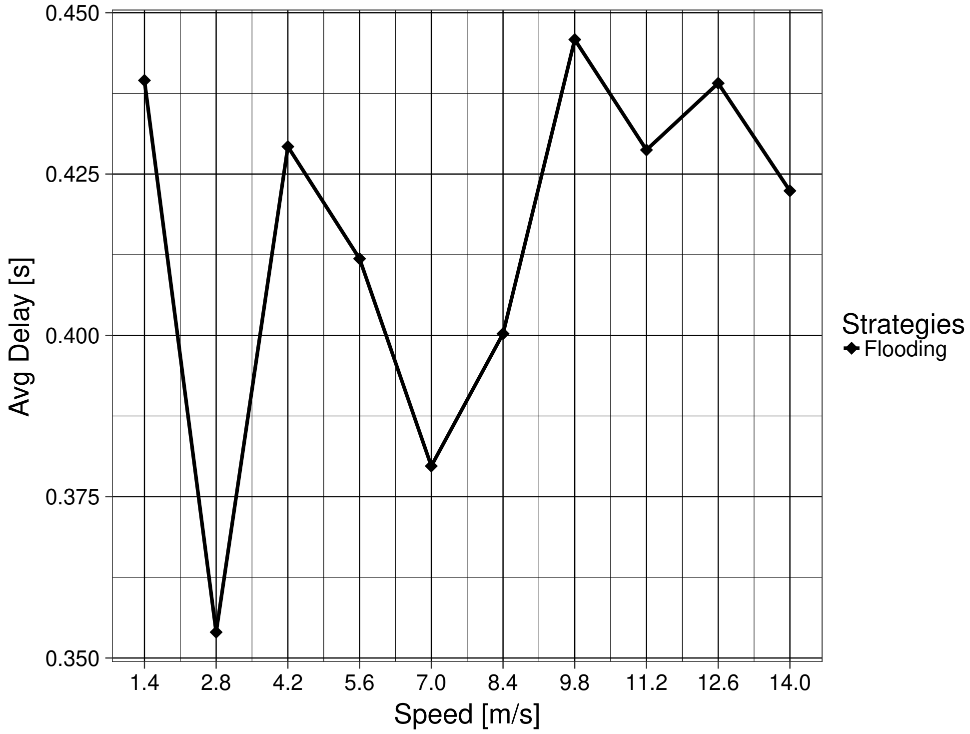

Figure 6.7: Mean network delay for consumer for mobile consumer and push producer using Flooding forwarding strategy

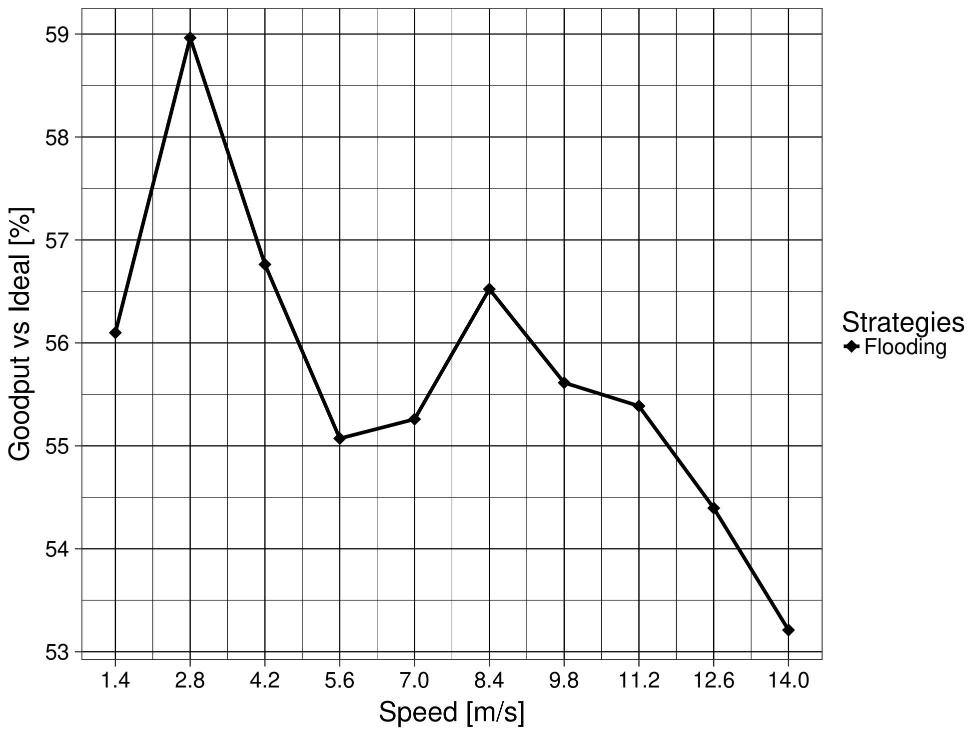

Figure 6.8: Mean goodput percentage of ideal for mobile consumer and push producer using Flooding forwarding strategy

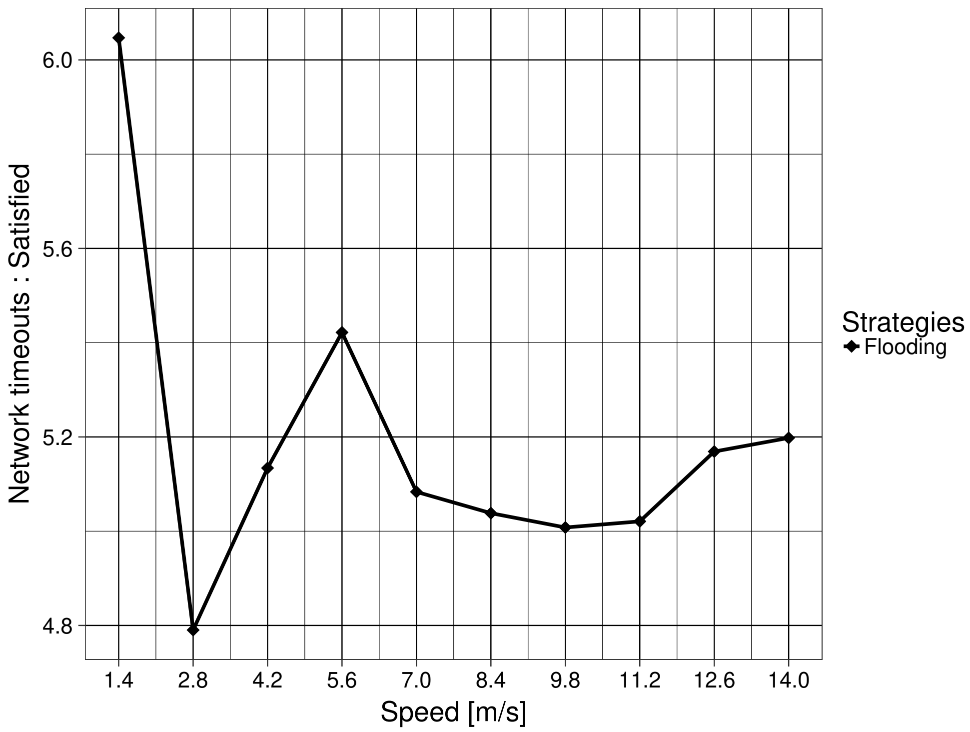

Figure 6.9: Timed out Interests to satisfied consumer Interests ratio for mobile consumer and push producer using Flooding forwarding strategy

maximum tested speed of $14m/s$ shows a remarkable difference in delay of about 300%.. It is interesting to note that the delay for all forwarding strategies is always lower than $0.38s$.

The mean goodput compared to the ideal goodput of 2500kbit for the four forwarding strategies are shown in Figure 6.11. Once again a clear difference in making a network naming scheme answer positively to question e1 is observed. SF MM PR 3N is always more than 3% away from the next best forwarding strategy, which is SF MM PR. For both SF MM PR 3N and SF MM PR, the goodput is maintained above 85% of the ideal, regardless of the moving speed of the nodes. For both SF and SF MM, the goodput starts above 82% at $1.4m/s$ and fluctuates, then quickly decreases to below 83% of the ideal goodput when the nodes start moving faster than $8.4m/s$.

The ratio of network timed out Interests vs satisfied consumer Interests for the four forwarding strategies are shown in Figure 6.12. There is once again a large difference between making a network naming scheme answer positively to question 1. SF MM PR

3N has the lowest ratio of timed out Interests vs satisfied consumer Interests which is relatively stable a close to zero, regardless of the node movement speed. SF MM PR is extremely close to SF MM PR 3N and is relative stable with a ratio under 0.1. SF and SF MM both start with a low ratio, only slightly worse than SF MM PR and SF MM PR 3N, but it continues to worsen as the node speed increases. SF and SF MM starts with a ratio slightly lower than 0.1, reaching a ratio above 0.6 with SF MM being at every point worse than SF MM.

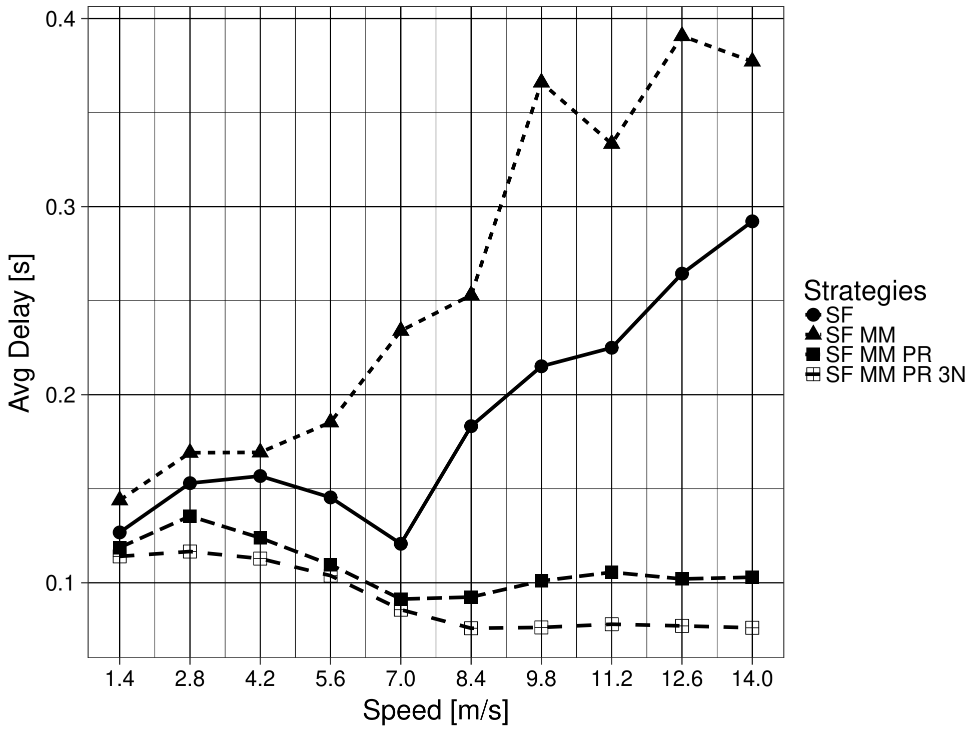

Figure 6.10: Mean network delay for consumer for mobile consumer and push producer using forwarding strategies

6.2.1 Discussion

For the push producer scenario, as was explained in Section 4.7.1 and Section 4.7.2, the issue with a push producer is that the relevant information related to one particular ICN name is dispersed throughout the network. Thus one is no longer searching for a single ICN name, one is searching for the Parts related to an ICN name. This changes the focus

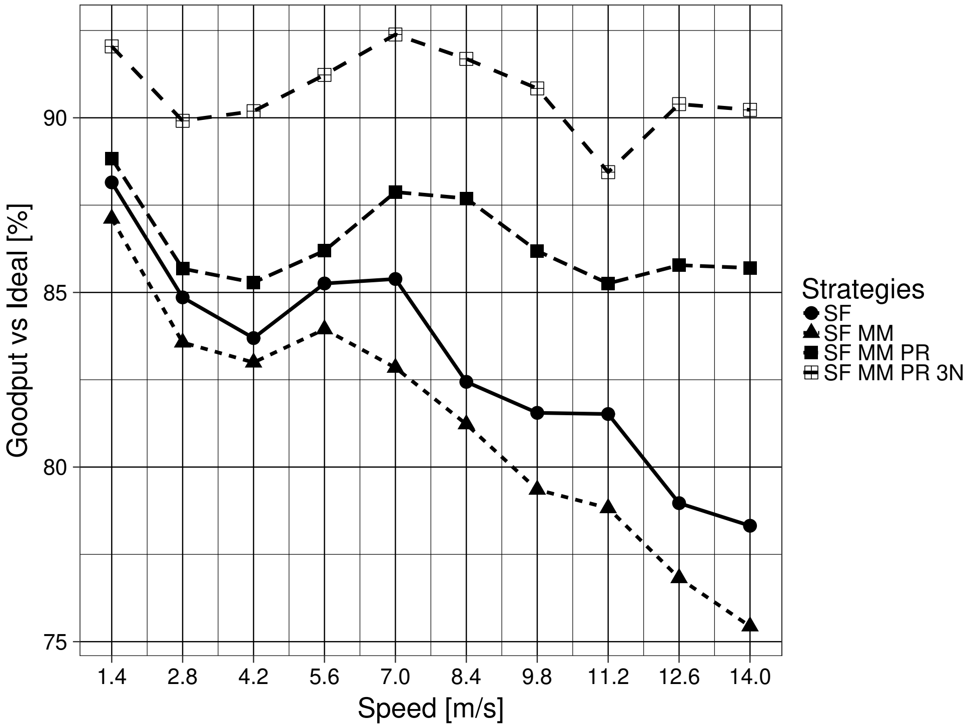

Figure 6.11: Mean goodput percentage of ideal for mobile consumer and push producer using forwarding strategies

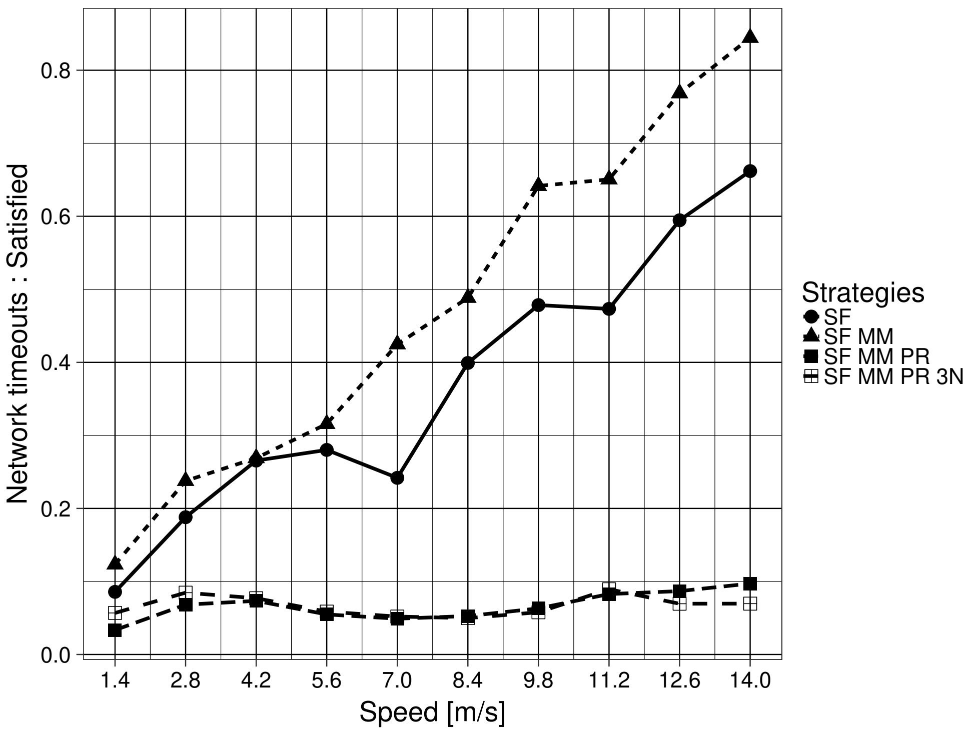

Figure 6.12: Timed out Interests to satisfied consumer Interests ratio for mobile consumer and push producer using forwarding strategies

of the questionnaire to the Parts.

In our scenario, the push producer is only pushing to the edges, ensuring that in 83% of the cases, the consumer is not connected to the edge that has the Part of the ICN name that is required. As part of the scenario setup, the information being requested is ensured to be somewhere within the network. This means that if all branches of the hexagon graph are searched, the requested information can be found.

This last point is very important because it explains why Flooding has a goodput that is above 50% regardless of node speed. This is not a great result, but it shows what a simple forwarding strategy can do if the information is know to be within the area that is flooded. However, the cost in delay and the network timed-out Interests to consumer satisfied Interests ratio, which is always above 4.8, is still very high. This is even more true when one remembers that these results are produced by a single consumer node.

SF does better than Flooding, because when it does find part of the information, it focuses all the PDUs to that given interface (G) until that interface, which is a node on the edge of the graph of the network, doesn't have the Part that SF is looking for. Due to timeouts, SF is forced again to flood the network. This strategy works well at first, but as the speed of the producer increases, and at the same time the amount of continuous Parts that are in a edge node decrease, SF sees a decrease in goodput and an increase in network timed-out Interests to consumer satisfied Interests ratio.

SF MM performs worse than SF because SF MM ties the ICN name to the interfaces that lead to the push producer node. If the push producer was pushing the exact Part that the consumer was requesting when the MapMe enhancement updated the interfaces, the results would probably be better than when using SF. In this scenario, the push producer is given 20 seconds to push content before the consumer starts requesting information in order to ensure that the information push is always in front of the information request. However, it is known that figure F is the ICN name related Parts, so forcing the network to send the information to the push producer is misguided. This results in a worse overall goodput and a higher network timed-out Interests to consumer satisfied Interests ratio than SF.

SF MM PR, detailed in Section 4.7.1, fixes the issue with SF MM by being able to answer the question of where a Part is located. Although the Part is mapped to an interface, because there is no motion, the results show a considerable increase in goodput. The lack of motion for where the Part originally was doesn't mean that the information is not cached. In the graph of the network, binding a Part to an interface is not much different to binding the Part to a node because the graph has only one path from the central node to every WAP. This is a feature of the graph of the chosen network which will also come into consideration when discussing the SF MM PR 3N results.

SF MM PR 3N out performs SF, SF MM and SF MM PR 3N, regardless of node speed. SF MM PR 3N mirrors the results of SF MM PR, which is due to the location of a Part not being influenced by binding the G to a 3N name, like in SF MM PR 3N, or to an interface, like in SF MM PR. This is due to the nature of the graph of the network in the scenario that has a single set of paths for both names SF MM PR 3N achieves a clear an extra $> 4\%$ increase in goodput which is realized by the use of the underlying metric space of 3N names and the mechanism PDUs that keep the NNPT updated, allowing PDUs to be returned to the consumer, regardless of motion. A similar increase was seen in goodput performance when using SF 3NC compared to SF in Section 6.1.

From the evaluation results, it is clear that a forwarding strategy which answers positively to question 1 has a significant improvement in the delay, goodput and Interest timeout to satisfied consumer Interest ratio. Specifically, SF MM PR and SF MM PR 3N outperform SF and SF MM in all the considered metrics. It is also interesting that SF outperforms SF MM regarding delay, goodput and Interest timeout to satisfied consumer Interest ratio in the push producer scenario. Although SF MM seems to answer question 3 positively in our questionnaire, it is not answering the place specification for the right figure F. Another observed improvement is for a forwarding strategy which answers positively to question 2. Although SF MM PR has the same tendency as SF MM PR 3N, SF MM PR 3N outperforms all the other forwarding strategies, having the overall lowest delay, best goodput and lowest Interest timeout to satisfied consumer Interest ratio regardless of node movement speed.

Chapter 7

Conclusions and recommendations

Network naming is a theme that is generally overlooked in network research. The current body of work in ICN attempts to do away with naming altogether. The network architecture on which society relies, the IP network architecture, has not changed the fundamental naming structure since its inception.

Utilizing the research by Levinson from the field of linguistics it has been shown that place specification has a formal definition that requires that one formally define what one is looking for, the figure F, and the search domain G for where one expects to find that figure. The names that are given to these object should, in the worst of cases, map to topological notions in order to be able to organize and categorize the objects in a graph. If that is done and a method to keep updates of this relator of F and G over time, then support for tracking of objects over time can be realized.

The information is condensed into a questionnaire with 3 questions. The questions are extremely simple but the implications and the interpretations are nuanced. An interpretation of this questionnaire was applied to the ICN architecture and to our own 3N architecture, to both demonstrate the architectures' weaknesses and strengths. The questionnaire was then utilized to enable the 3N architecture to answer to support all the requirements to answer to the questionnaire in a positive manner.

Both the ICN architecture and the 3N architecture were completely implemented the in open source discrete event simulator ns-3 and a framework for mobile producers with mobile consumers as well as mobile push producers with mobile consumers was produced and released to the networking community.

Our results have shown that networking naming can have a huge effect in goodput, delay and network time-outs. In the mobile producer with mobile consumer scenario, the use of the knowledge of place specification and metric spaces, permits the best 3N architecture forwarding strategy to outperform ICN forwarding strategies by over 400% in goodput regardless of the node movement speed. In the mobile push producer scenario, the same knowledge permits the best 3N architecture forwarding strategy to outperform ICN forwarding strategies by 115%, at the highest node movement speed. In both cases, the delay and network timed-out Interest to consumer satisfied Interests ratio is the lowest utilizing the best 3N architecture forwarding strategy that leverages the knowledge of place specification and metric spaces.

The simple ideas presented in this document and the detailed explanations given about how place specification is defined and how to implement it in a given network architecture can be of extreme benefit to improving current network architectures, or starting from a more beneficial position should a clean slate network architecture approach be taken.

Bibliography

[1] Gnu general public license. http://www.gnu.org/licenses/gpl.html.

[2] Afanasyev, A., Moiseenko, I., and Zhang, L. ndnSIM: NDN simulator for NS-3. Technical Report NDN-0005, NDN, October 2012.

[3] Augé, J., Carofiglio, G., Grassi, G., Muscariello, L., Pau, G., and Zeng, X. Map-me: Managing anchor-less producer mobility in information-centric networks. *CoRR abs/1611.06785* (2016).

[4] Azgin, A., Ravindran, R., and Wang, G. Mobility study for named data networking in wireless access networks. In *2014 IEEE International Conference on Communications (ICC)* (Sydney, NSW, Australia, Jun 2014), IEEE, pp. 3252–3257.

[5] Bourbaki, N. *General Topology Chapters 5-10.* Springer-Verlag Berlin Heidelberg, New York, 1989.

[6] Bourbaki, N. *General Topology Chapters 1-4.* Springer-Verlag Berlin Heidelberg, New York, 1995.

[7] Clark, D., and Cohen, D. A proposal for addressing and routing in the internet. IEN 46, IEN, June 1978. `https://www.ietf.org/rfc/ien/ien46.txt`.

[8] Clark, D. D. *Designing an Internet.* The MIT Press, Cambridge, Massachusetts, 2018.

[9] Cunche, M. I know your MAC Address: Targeted tracking of individual using Wi-Fi. In *International Symposium on Research in Grey-Hat Hacking - GreHack* (Grenoble, France, Nov. 2013).

[10] Day, J. *Patterns in Network Architecture: A Return to Fundamentals*. Pearson Education, Boston, MA, USA, 2007.

[11] Developers, N. Nfd - named data networking forwarding daemon. Available Online `https://named-data.net/doc/NFD/current/overview.html`. Accessed: 2019-01-20.

[12] Donatiello, L., Marfia, G., Bujari, A., and Palazzi, C. E. A simulation model for event goodput estimation in real-time sensor networks. In *2017 IEEE/ACM 21st International Symposium on Distributed Simulation and Real Time Applications (DS-RT)* (Oct 2017), pp. 1–7.

[13] Droms, R. Dynamic host configuration protocol. RFC 1531, IETF, October 1993. `https://tools.ietf.org/html/rfc1531`.

[14] Ericsson, T. L. M. Ericsson mobility report: November 2018. Available Online: `https://www.ericsson.com/assets/local/mobility-report/documents/2018/ericsson-mobility-report-november-2018.pdf`. Accessed: 2019-01-19.

[15] Fang, C., Yao, H., Wang, Z., Wu, W., Jin, X., and Yu, F. R. A survey of mobile information-centric networking: Research issues and challenges. *IEEE Communications Surveys Tutorials 20*, 3 (thirdquarter 2018), 2353–2371.

[16] Feng, B., Zhang, H., Zhou, H., and Yu, S. Locator/identifier split networking: A promising future internet architecture. *IEEE Communications Surveys Tutorials 19*, 4 (Fourthquarter 2017), 2927–2948.

[17] Fuller, V., and Li, T. Classless inter-domain routing (cidr): The internet address assignment and aggregation plan. RFC 4632, IETF, San Jose, CA, USA, Aug 2006. http://tools.ietf.org/html/rfc4632.

[18] Fuller, V., Li, T., Yu, H., and Varadhan, K. Classless inter-domain routing (cidr): an address assignment and aggregation strategy. RFC 1519, IETF, September 1993. https://tools.ietf.org/html/rfc1519.

[19] Greenstein, S. *How the Internet Became Commercial: Innovation, Privatization, and the Birth of a New Network.* Princeton University Press, 2015.

[20] Ieee standards association. http://standards.ieee.org/faqs/regauth.html, Jan. 2015. Accessed: 2015-01-02.

[21] Information Science Institute, U. o. S. C. Internet protocol - darpa internet program - protocol specification. RFC 791, IETF, September 1981. https://tools.ietf.org/html/rfc791.

[22] Ishakian, V., Akinwumi, J., Esposito, F., and Matta, I. On supporting mobility and multihoming in recursive internet architectures. *Comput. Commun. 35*, 13 (July 2012), 1561–1573.

[23] Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., and Braynard, R. L. Networking named content. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies* (Rome, Italy, 1-4 Dec, 2009), CoNEXT '09, ACM, pp. 1–12.

[24] Karrakchou, O., Samaan, N., and Karmouch, A. A survey of routing mechanisms in icns. In *2018 6th International Conference on Multimedia Computing and Systems (ICMCS)* (May 2018), pp. 1–6.

[25] Kozierok, C. *The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference.* No Starch Press Series. No Starch Press, 2005.

[26] Levinson, S. C. *Space in Language and Cognition: Explorations in Cognitive Diversity*. Language Culture and Cognition. Cambridge University Press, Cambridge, UK, 2003.

[27] Lin, Z., and Pearson, S. An inside look at industrical ethernet communication protocols. http://www.ti.com/lit/wp/spry254b/spry254b.pdf, 2018. Accessed: 2019-01-06.

[28] López, J., Arifuzzaman, M., Zhu, L., Wen, Z., and Takuro, S. Seamless mobility in data aware networking. In *Proceedings of the 2015 ITU Kaleidoscope: Trust in the Information Society, K-2015 - Academic Conference* (Barcelona, Spain, 1 9-11 Dec, 2016), IEEE.

[29] López, J., and Sato, T. Seamless mobility in icn for mobile consumers with mobile producers. *IEICE Transactions on Communications E100B*, 10 (10 2017), 1827–1836.

[30] López, J. E., and Sato, T. nnnsim 3n based icn simulator. Available Online: https://bitbucket.org/nnnsimdev/nnnsim. Accessed: 2019-06-28.

[31] Lu, Q., Huang, X., and Wang, Y. Leetar: A new hierarchical-structural topology generator for the internet. In *2011 3rd International Conference on Computer Research and Development* (March 2011), vol. 1, pp. 241–245.

[32] Mailland, J., and Driscoll, K. *Minitel: Welcome to the Internet*. The MIT Press, Cambridge, Massachusetts, 2017.

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

[34] Mockapetris, P. Domain names - implementation and specification. RFC 1035, IETF, November 1987. https://tools.ietf.org/html/rfc1035.

[35] ns 3 developers. ns-3. Available Online: http://www.nsname.org. Accessed: 2019-06-28.

[36] Nsf named data networking project. http://www.named-data.net, June 2014. Accessed: 2014-06-07.

[37] of Adelaide, T. U. The internet topology zoo. Available Online: http://www.topology-zoo.org/index.html. Accessed: 2019-06-28.

[38] Ohlman, B. From id/locator split to icn. In *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)* (Las Vegas, NV, USA, Jan 9-12 Jan, 2015), IEEE, pp. 256–261.

[39] Ortiz Jr., S. Internet researchers look to wipe the slate clean. *Computer 41*, 1 (Jan 2008), 12–16.

[40] Peters, B. *How Not to Network a Nation: The Uneasy History of the Soviet Internet*. The MIT Press, Cambridge, Massachusetts, 2016.

[41] Rekhter, Y., and Li, T. An architecture for ip address allocation with cidr. RFC 1518, IETF, September 1993. https://tools.ietf.org/html/rfc1518.

[42] Rexford, J., and Dovrolis, C. Future internet architecture: Clean-slate versus evolutionary research. *Commun. ACM 53*, 9 (September 2010), 36–40.

[43] Riley, G. F., and Henderson, T. R. The ns-3 Network Simulator Modeling and Tools for Network Simulation. In *Modeling and Tools for Network Simulation*, K. Wehrle, M. Güneş, and J. Gross, Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, ch. 2, pp. 15–34.

[44] Rossi, R., Fahmy, S., and Talukder, N. A multi-level approach for evaluating internet topology generators. In *2013 IFIP Networking Conference* (May 2013), pp. 1–9.

[45] Saltzer, J. H. On the naming and binding of network destinations. North-Holland Publishing Company, pp. 311–317.

[46] Saltzer, J. H. On the naming and binding of network destinations. RFC 1498, IETF, Cambridge, MA, USA, August Aug, 1993. `http://tools.ietf.org/html/rfc1498.txt`.

[47] Shoch, J. F. Inter-network naming, addressing, and routing. In *In Proceedings of IEEE Computer Conference, COMPCON* (Jan. 1978), pp. 72–79.

[48] Sunshine, C. A. Addressing problems in multi-network systems. IEN 178, IETF, April 1981. `https://www.ietf.org/rfc/ien/ien178.txt`.

[49] Takeda, K. User identification and tracking with online device fingerprints fusion. In *Security Technology (ICCST), 2012 IEEE International Carnahan Conference on* (Oct 2012), pp. 163–167.

[50] Tang, J., Zhou, H., Liu, Y., Zhang, H., and Gao, D. A source mobility management scheme in content-centric networking. In *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th* (Jan 2014), pp. 176–181.

[51] Tozzi, C. *For Fun and Profit: A History of the Free and Open Source Software Revolution.* The MIT Press, 2017.

[52] VNI, C. Cisco visual networking index: Forecast and methodology, 2016-2021.

[53] Watson, R. W. The delta-t transport protocol: features and experience. In *[1989] Proceedings. 14th Conference on Local Computer Networks* (Oct 1989), pp. 399–407.

[54] Xiao-hui, K., Min-huan, H., Yong-gang, C., Jin, L., and Fei, X. Large-scale network topology generation for emulation environment. In *2010 First International Conference on Networking and Distributed Computing* (Oct 2010), pp. 392–396.

[55] Yi, C., Afanasyev, A., Moiseenko, I., Wang, L., Zhang, B., and Zhang, L. A case for stateful forwarding plane. *Computer Communications 36*, 7 (2013), 779 – 791.

[56] Yuan, F., Liu, F., Huang, D., Liu, Y., and Luo, X. A high completeness pop partition algorithm for ip geolocation. *IEEE Access* (2019), 1–1.

[57] Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., claffy, k., Crowley, P., Papadopoulos, C., Wang, L., and Zhang, B. Named data networking. *SIGCOMM Comput. Commun. Rev. 44*, 3 (July 2014), 66–73.

[58] Zhu, Z., Wakikawa, R., and Zhang, L. A survey of mobility support in the internet. RFC 6301, IETF, Los Angeles, CA, USA, Jul 2011. `http://tools.ietf.org/html/rfc6301`.

Publication Achievement

[1] J. López, M. Arifuzzaman, L. Zhu, Z. Wen, and S. Takuro, "Seamless mobility in data aware networking," in *Proceedings of the 2015 ITU Kaleidoscope: Trust in the Information Society, K-2015 - Academic Conference*, (Barcelona, Spain), IEEE, Jan 2016.

[2] J. López and T. Sato, "Seamless mobility in icn for mobile consumers with mobile producers," *IEICE Transactions on Communications*, vol. E100B, pp. 1827–1836, Oct 2017.

[3] J. López, Q. N. Nguyen, Z. Wen, K. Yu, and T. Sato, "Using linguistic properties of place specification for network naming to improve mobility performance," *Sensors*, vol. 19, no. 13, 2019.