

A Framework for Named Networking on Edge AI

Fritz Kevin Flores and Marnel Peradilla

Advanced Research Institute for Informatics, Computing, and Networking
De La Salle University

Manila, Philippines

Email: fritz.flores@dlsu.edu.ph, marnel.peradilla@dlsu.edu.ph

Abstract— Edge AI is an integration of Edge Computing and Machine Learning algorithms to address concerns with how Machine Learning is currently used on the Cloud by reducing network latency, providing faster response and near-real-time classification, better control of data security and privacy, among others. However, one of its limiting factors is due to its tight integration with the Internet Protocol, which presents various concerns due to address range limitations, lack of context on the address identifiers, but more so on its host-centricity, which is a paradigm that no longer reflects how the world uses the Internet today. Because of these limitations, this study investigates the use of a different approach in communication, through Named Networking; a subset of the proposed future Internet architecture, combining concepts of Named Data Networking, Named Function Networking, and Machine Learning algorithms, to create a framework for use in Edge AI.

Keywords— *Edge AI; NDN; NFN; Named Networking*

I. INTRODUCTION

Advancements of the Internet paved the way for Cloud Computing, allowing various devices and technologies to transmit and share data through interconnected networks, store a variety of data on storage equipment, and process them on high-performance computing servers which are distributed throughout the Internet. Because of this, technologies such as the Internet of Things (IoT) and Big Data were realized, enabling users to process various amounts and diverse data for use in either artificial intelligence (AI), business intelligence, analytics, and others. However, this cloud-centric architecture assumed that the data must be transferred to the cloud, prior to processing, thus introducing latencies as well as additional network resource consumption [1].

To address those concerns, Edge Computing was created to bring the data processing capability closer to the origin or source of the data, instead of the cloud, solving concerns on traditional cloud-based applications. These include reducing network latency, since data no longer needs to be uploaded to the cloud before processing, resulting in faster response and near-real-time classification, as well as better control of data security and privacy since data is only transmitted and processed locally which decreases the risk of data being exposed to untrusted networks, among others. Because of this development, AI-based applications became more accessible and practical for use in a variety of domains, terming this integration of Edge Computing and AI as Edge AI.

Although Edge AI is able to address issues in latency and security, further improvements may be realized by decoupling its tight integration with the Internet Protocol (IP); of which generally most computing resources, such as compute, storage, security, and networking operate on. The concern stems from the knowledge that the IP was developed during the 1980's to address the issues of host-centricity at that time, during the era of the telephony, which is different from how the world uses the Internet today. Over the years of using the Internet, multiple concerns have been realized, such as with the address range limitations of IPv4 and such. Additionally, majority of the Internet traffic nowadays is focused on the exchange of information or content rather than its source or destination, hence the Internet today is more data-centric rather than the previous host-centric. Although many issues of IP were solved with the introduction of IPv6 and many application layer protocols, it is still built on the same host-centric paradigm.

Proposed future Internet architectures were introduced such as the Information-Centric Networking (ICN), an approach to move from a host-centric to a data-centric paradigm, giving more importance to the data being transmitted than the sender who is transmitting. Additionally, ICN enables data to become independent from the end-device or technology, allowing data to be retrieved using multi-access communications and caching, thus reducing network latency and bottlenecks. ICN design foundations typically focus on using a variant of the publish-subscribe system as well as names to identify certain data or function. One of the notable implementations of ICN is Named Data Networking (NDN) which follows a new architecture that is independent on IP, following a named-data approach where names are used to identify data, as well as uses an Interest/Data packet approach for communication.

NDN solves many of the concerns of IP and has matured to the point of already having simulation tools as well as a variety of deployment scenarios. Further extending the named-data approach of NDN is Named Function Networking (NFN), which introduces the use of named-functions to enable in-network processing of data, allowing data to be processed prior to transmission, thereby reducing network bandwidth usage as well as off-loading application tasks to the network. The introduction of NDN and NFN introduces many possibilities in improving application-specific implementations by improving data transmission as well as with in-network processing. With these, the study investigates the possibility of creating a framework for use in Edge AI, incorporating concepts of NDN, NFN, and Machine Learning algorithms.

II. BUILDING BLOCKS FOR THE PROPOSED FRAMEWORK

The architecture and design principles [2] which made the Internet, focusing on a host-centric paradigm for end-to-end connectivity, enabled point-to-point telephony technology to computers, allowing them to converse with one another. Despite the original intentions for the Internet [3] as a communication network, its unexpected growth in the areas of social media, e-commerce, and other innovations, has evolved its function to more of a distribution network [4], focusing more on content distribution and consumption, than what was originally intended. This in-turn presents some restrictions but also possible research opportunities from the end-to-end addressing and nature of the Internet Protocol, such having a limited address space and its focus on a single-source communicating to a single-destination, despite the data possibly existing in multiple locations. This paved the way to explore the possibilities of transitioning from the current Internet architecture from a host-centric to a data-centric or information-centric architecture, such as that of NDN.

A. Named Data Networking

NDN is a novel architecture whose design principles are based on the Internet, but generalizing the architecture to use hierarchically structured names to identify data, objects, or named content chunks, transitioning from the host-centric network architecture of IP, to data-centric architecture [2][4]. Communication in NDN is based on the exchange of two packet types: (1) the *Interest* packet is sent by a consumer to request data; (2) the *Data* packet is replied by a node that contains the requested data. NDN Routers perform forwarding [4] using three data structures: (1) *Pending Interest Table* (PIT) is used to store the Interests with its incoming and outgoing ports, to identify the originating port as well as reduce redundant Interests; (2) *Forwarding Information Base* (FIB) forwards Interests using a forwarding strategy, such as the longest match, allowing a router to determine the outgoing interface an Interest needs to take; (3) *Content Store* (CS) is a Data cache, allowing a router to cache Data packets to speed up retrieval as well as satisfy future requests.

NDN also follows a hierarchical naming scheme which can be adjusted for different implementations, for scalability, while being independent from the network [4]. Hierarchical naming can be somewhat similar to a Uniform Resource Identifier (URI) or Uniform Resource Locator (URL), separating names, groups, or objects into a hierarchical sequence.

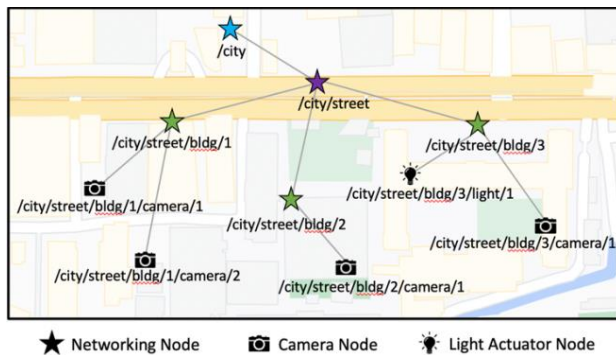


Fig. 1. Smart City Street Camera and Lights

Such an example of an Smart City implementation with node identifiers using hierarchical naming is shown on Fig 1., where `/city/street/bldg/1/camera/` may refer to the metadata or details of all cameras on an identified location; `/city/street/bldg/1/camera/1/1100` may refer to an hour of video footage from a selected camera at the identified location from 11:00AM; and `/city/street/bldg/1/camera/1/cctv.mov` may refer to the full video footage from a selected camera. The transition of using names to reference data or objects instead of fixed numerical addresses, would remove address range limitations as well as provide context in the address identifiers without the need of an additional upper layer, such as that of the currently used Domain Name System (DNS). The innate forwarding data structures of NDN would also enable in-network data caching, which could improve overall network performance, for multiple requests of the same data.

B. Named Function Networking

NFN is an extension NDN, such that it does not only support naming of data and objects, but also function definitions and application to data as well [6]. NFN complements the information retrieval of NDN, with the information processing done on Edge or Cloud Computing, effectively removing both the *locality-of-storage* and *locality-of-execution*. NFN orchestrates the interaction of the functions with the data on behalf of the user, allowing compute jobs to be distributed across different nodes in the network [3]. NFN nodes may be selected for compute jobs using either of the mechanisms [3]: (1) Proactive approach sends periodic messages containing its functions and resource utilization; (2) Reactive approach information is only sent when it is requested by a consumer. Compute jobs may also use smart deferral schemes to enable a more effective selection of the node to perform the execution, such as the node with the lowest resource utilization [4].

NFN follows the same hierarchical naming of NDN and is called by appending the content as parameters during the function call. The main difference however is that while NDN focuses of name resolution or lookup, NFN focuses on the expression or processing of the data [5]. Such examples of named-functions in Smart City applications using Fig. 1, can be `/get/size(/city/street/bldg/1/camera/1/cctv.mov)` where the named-function `/get/size()` may refer to requesting the file size of the data `/city/street/bldg/1/camera/1/cctv.mov`. The transition of data processing capabilities from the application layer to the network layer, may significantly reduce network bandwidth requirements since data to be requested can be pre-processed before being sent to the network.

C. Edge AI

With the growing trend of Edge Computing, more specifically Edge AI, applications that use a variety and large amounts of data to be processed, are now implemented closer to the origin of the data, thus reducing latency, and allowing for near-real-time response. Edge AI implementations with traditional computing resources, IoT devices, or even using MANETs, may generally use TCP/IP or other host-centric communication protocols, however, are still subject to the same limitations of the end-to-end communication paradigm.

The capabilities of both NDN and NFN enables Edge AI to have a different perspective in implementation, which allows the communication protocol to focus on the data rather than the destination. Additionally, NDN and NFN paves the way for Named AI Networking [7], which enables every node in the network to contribute to the AI workflow, allowing data collection, training, and inferencing to be done using named data and functions. This enables different devices in the network to collectively share resources, balance computing capabilities, distribute training tasks, create complex workflows, and others. Combining data-centric protocols of NDN, NFN, and Named AI may reveal certain advantages in Edge AI application environments.

Implementing Edge AI with NDN, NFN, and Named AI, enables both training and inferencing of basic AI processing to be performed at the edge nodes, while leaving the larger processing requirements to be done in the Cloud, creating a fully self-functioning edge computing network design, which performs the full end-to-end process of data collection through sensing, networking, storage, processing, training, inferencing or prediction, to actuation, and is thus termed by this study as Named Networking, focusing its application on Edge AI.

III. FRAMEWORK DESIGN CONSIDERATIONS

The idea of Named Networking in this study, is a blanket terminology which combines named-data, function, and AI among others that takes advantage of using names as identifiers of computing resources in the networking or communication paradigm. NDN allows nodes as well as data or content chunks to be identified as names used in routing and forwarding; NFN enables data processing jobs or compute functions, such as data normalization, scaling, data fusion, and others; Named AI provides capabilities including native federated learning, distributed inferencing, and cache results; among others.

With these, the proposed Named Networking Framework would need to be able to support the following design considerations, in order to address the existing Internet, or more specifically, Edge Computing requirements.

A. Extensibility

One of the main advantages of how the current Internet functions is that it follows a layered approach, enabling the Network or Internet layer to focus solely on delivering the data from end-to-end. This allows a variety of applications and functions to be implemented above the IP layer, while also enabling flexibility to the lower layers to use different interfaces and transmission media, utilizing the universal data forwarding and routing capabilities of IP. The layered approach would be advantageous in designing this framework, since it allows each layer to be improved and developed independently from the others, without compromising their functions.

B. Data-Centricity

The use of the Internet nowadays focuses more on content distribution and consumption, instead of the intended end-to-end communication, thus introducing inefficiencies in network communications. Although majority of the existing higher-layer communication protocols and applications rely on IP and

its universality, some of these protocols may have potential which are not fully realized in an end-to-end communication framework. By transitioning to a network that is designed to focus on the data rather than the destination, certain improvements may be observed, as certain architectural advantages may be considered, such as in-network caching and multicasting. With a data-centric architecture, capabilities such as security, processing, and caching among others, can be tightly integrated in the routing and forwarding processes of the usual network layer. This would mean that instead of relying on additional overhead and processes of the higher layers, these functionalities may be configured in a way that it addresses current issues, mostly on the lower layers of communication.

C. Scalability

The Internet is home to around 21 billion devices by 2025 [8] and is expecting a further exponential increase in the coming years. As the number of devices increase, so does the amount of data being transmitted, and as such, considering a transition from a numerically limited addressing space to virtually unlimited name-based identifiers as well moving from a host-centric to data-centric paradigm, would allow this increase to be accommodated. This increase not only considers traditional computing resources but as well as devices that are related to IoT, which are heterogeneous and pervasive. Thus, in order to address this possible influx of devices as well as consider possible futureproofing, the proposed framework must be capable of accommodating growth.

D. Interoperability

The Internet is home to heterogeneous devices that use IP as somewhat of a middleware, allowing interoperability between the different devices. With the sudden rise of IoT devices and Wireless Sensor Actuator Networks (WSAN), more devices are being created as well as network interfaces and protocols are continuously being improved, which further expands the already heterogeneous Internet. The Edge, being a subset of the Internet or Cloud, may also experience the same concerns with heterogeneity; where different IoT devices with different sensors, collect data and transmit them via various communication media, to then be aggregated and processed, before sending them back for actuation. This would mean that the Edge environments, would need to be capable of addressing heterogeneity by having a middleware or some flexibility in the layered design in order to accommodate such requirements, thus the proposed framework must consider interoperability with the different existing platforms and communication media allowing seamless integration with existing network setups.

E. Adaptable

Lastly, Edge environments are developed and deployed near the data producers or sources, to support the intended application or domain requirements, reducing latency for near-real-time application scenarios. Such examples could be for tracking vehicle traffic in a certain intersection or highways for automated traffic management; another may focus on managing environmental conditions in multiple indoor farming setups for automated agriculture; or even using computer vision and IoT for improving processes on smart retail.

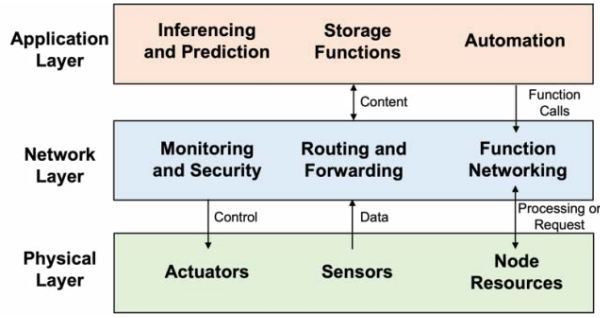


Fig. 2. Proposed Named Networking Framework for Edge AI

There are many different possible applications in Edge Computing and having the capability to adapt to these different kinds of application or domain implementations, would result in easier migration and adoption of a new framework. This would mean that the proposed framework should be capable to certain general functionalities that are widely and commonly used across different applications, but still be capable of addressing and support specific domain requirements.

IV. PROPOSED NAMED NETWORKING FRAMEWORK

The proposed framework in Fig. 2, is divided into three layers, namely: (1) Physical Layer, which represents the different hardware components, interfaces, and resources of the devices or nodes; (2) Network Layer where the routing and forwarding, as well as in-network functionality are located; (3) Application Layer focusing on the functionality of the nodes and that of the application. The layers of the proposed frameworks are designed to be compact, such that it increases the responsibility and functionalities of each layer as well as to be able to take advantage of processing at the lower layers.

A. Physical Layer

The Physical Layer represents the node itself and the hardware components, which includes sensors, actuators, storage, compute, memory, among others. Nodes may generally range from resource-constrained to resource-rich nodes. Resource-constrained nodes may be described as nodes that are battery operated, have volatile-only memory, limited sensing capabilities, low data-rate transmission, among others, which may be typically attributed to microcontrollers and even single-board computers. Resource-rich nodes on the other hand, may be attributed to nodes with dedicated continuous power, high-speed primary and secondary storage, high-performance computing capabilities, among others, which may even be attributed to traditional computers or even servers, who may even be configured with certain special functions or roles.

B. Network Layer

The Network Layer represents the networking functionality of the framework, where Named Networking is implemented with its various functionalities: name resolution, routing, forwarding, and processing of packets. Additionally, in-network functions, monitoring, and security are also present, in order to support the network, by incorporating these as native capabilities of the layer.

Since Named Networking, enables the use of a semantically rich naming scheme to reference resource, such as data, nodes, functions (i.e., data processing, model training, inferencing, predicting) [7] and others; actions in the network, such as retrieval of data, orchestrating of jobs, performing load balancing or federated learning, etc. are enabled because of network communication. Environments with heterogeneous nodes, with each having different constraints and capabilities, may also have their nodes take on certain node roles, such as those for routing and forwarding, function processing, monitoring, and others.

C. Application Layer

Lastly, the Application Layer represents the higher-level functions of the framework, which are implemented by resource-rich nodes. As the network layer references, forwards resources, and functions of the network, the application layer is responsible for the functions of the nodes such as sending instructions to retrieve data from other nodes, implementing various data processing algorithms, before being available for retrieval, as used in Edge Computing environments, or more specifically Edge AI, computational tasks.

V. USE CASES

The proposed framework may be implemented through the following situations in Smart Agriculture and Smart Cities.

A. Smart Agriculture

Smart Agriculture is an approach for managing agricultural resources, such as farms, through the use of wireless communication, sensors, actuators, and AI among other technologies. Certain agricultural environments are known to have very little to no Internet connectivity, hence a possible solution is to have an in-house computing capabilities or nodes, to address the data processing and computational requirements. These applications can typically have various resource-constrained nodes scattered all throughout the farm area, with each having its own set of sensors to detect the various environmental information, such as temperature, humidity, moisture, and others, which forward their data to a cluster head or network routing node, for forwarding, or even directly to the node sink or base station, storing the information collected.

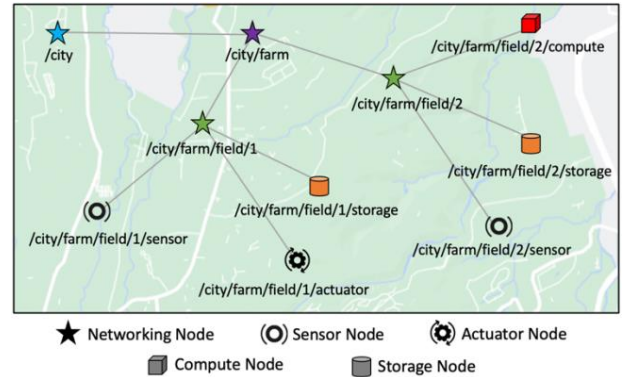


Fig. 3. Smart Agriculture Field and House Nodes

Some implementations may not even need to have their nodes send their data to the sink immediately after sensing or based on a predefined time interval but may also send it upon an on-demand user request. Special function nodes or nodes with specific roles, may perform additional functions such as duplicate data removal, data aggregation, data summarization, and such, before sending data to the sink. Such implementation of a Smart Agriculture environment can be seen on Fig. 3.

A separate resource-rich node can have the role of collecting the information or content received by the sink node and perform inferencing or prediction, based on the requirements of the agriculturalist, resulting into automated functions for notifications, alerts, as well as actuation commands forwarded to the nodes. These are in order to provide proper and timely intervention, such as opening exhausts for temperature regulation, activation of irrigation or fertigation among others.

With large scale environments, such as having multiple farms or farm areas, several resource-rich nodes which have a compute role, tasked to process data as well as perform inferencing and prediction, may even perform federated learning. This allows different compute nodes, even if they are comprised of heterogeneous devices or if they are scattered and distributed in the environment, to communicate and collaborate together on training as well as improving a shared model independently from each other, aggregating their small incremental updates as adjustments in weights of the model.

Other resource-rich nodes may also adopt roles, such as for orchestration, by allowing autonomous monitoring of the resource and workload of other nodes in the environment to balance load, redelegate tasks, or even provide alerts to indicate overall node health. These orchestration nodes may also be contact points of systems or applications, allowing user messages and requests to be sent to the orchestrator, which would then coordinate the assignment of processing tasks to the nodes, depending on resource availability or even prioritization. This in turn may provide better resource usage as well as balancing of load between nodes.

In a traditional IoT setup for smart agriculture, different considerations and possibly even issues may be observed. Each node would be assigned with unique numerical addresses to identify them, then have a mapping scheme in the higher layers, to establish the relationship between the node and its description; similar to the function of DNS for ease in identifying the node and its data from the others. This introduces increased overhead on different layers in the network design, as well as may introduce added protocol formats and packet types. Moreover, nodes are focused on sending their data to generally a single device, the sink node, which in certain situations may be congested, introducing possibilities of packet loss among others. Additionally, when data is requested by the user, the destination node or storage node needs to be identified before the request would be forwarded, however in Named Networking, the emphasis of the network is on the data, hence users do not need to identify the node addresses but instead focus on named identifiers such as locations in the farm, types of sensor data, timestamp of data as well as even processed data such as average temperature.

B. Smart City

In a Smart City environment, various technologies are deployed to collect environmental data, process the data into usable information, then convert it into actionable insights to improve city functions and operations. These environments are known to expect high data rate networks due to a dense number of sensor nodes per area, while experiencing generally congested network connectivity accesses from mobile Internet services. Additionally, some of these nodes may be expected to be mobile or moving, such as nodes attached to public vehicles for monitoring and telemetry, hence increasing the complexity of deployment. Furthermore, in relation to security, there are countless malicious individuals and threats which are usually present in a city, hence protecting data against theft, misuse, tampering, and such are also a much-needed consideration.

Applications in smart cities may require real-time calculations hence having the need to have lower latencies as well as high network bandwidth, together with the presence of local storage and local compute facilities, which are coincidentally present in Edge Computing environments, are something which needs to be considered on these environments. Such applications would include automating control of traffic lights based on the obtained vehicle and pedestrian traffic, vision-based driving violation with timely notifications to the traffic officers, among others.

Smart city nodes can vary from resource-constrained battery-operated nodes which sense the environment for air quality, vehicle traffic, resource consumption, and such to high-resource nodes which collect images, audio, and even videos, with such implementation seen on Fig. 1. Due to a dense population of wireless devices in a city, nodes would typically use different methods for communication, combining capabilities of both wired and wireless on different nodes and deployments to support the data forwarding requirements. Typical wireless protocols for use may include Wi-Fi, Bluetooth, as well as even low data rate protocols such as 6LoWPAN, LoRaWAN, ZigBee, or even the simple IEEE 802.15.4 among others. Depending on the type of protocol or data rate needed, smart city applications may choose to use one or even multiple of these protocols together, to enable data transmission from one node to another for either processing or storage needs.

Implementations for storage in smart cities may be flexible as well, such as using a cloud-based storage where nodes forward their data directly to the storage node or the sink as it is collected, while others may choose to have a local storage in the network or even an in-node storage and have the data processed locally, such as those implemented for Edge Computing. The advantage of having a local storage and following an Edge Computing deployment design, is that data may be collected and processed near the deployment area, hence providing timely results as well as decreasing the needed bandwidth to send the raw data to the Internet. In these deployments, resource-rich nodes may be deployed in the vicinity, to process the collected data, be it statistically or through machine learning models or algorithms, to provide near-real-time automation or timely intervention as needed.

Certain applications domains of smart environments in a smart city, areas in the city of which are highly populated or are priority areas for the city, may have denser node deployments, relating to possibly having higher volume and velocity of data. To support these scenarios, deployment of nodes can be distributed into clusters or zones, where each cluster can be assigned to a particular subset of the area, as well as allowing each cluster to each have its own dedicated storage and compute nodes. This would enable each cluster to function on its own with regards to collecting and preprocessing data, while still enabling federated learning or collaboration on shared model between different clusters. Additionally, this design would reduce the reliance of needing to have a dedicated high-speed Internet connection, as data being sent by the clusters may be in the form of processed data, incremental model updates, telemetry, among others, instead of the traditional setup where raw data is being transmitted from the node directly to the sink or the dedicated cloud storage.

With the proposed Named Networking framework, smart cities can take advantage of delegating processes and functions on resource-rich nodes, includes nodes with medium to high resource capabilities, while leaving the sensing capabilities to the resource-constrained nodes, all through coordination within the network layer, reducing the needed communication bandwidth on lower data-rate channels, such as that of the medium congested mobile broadband. Sector or cluster-based deployments in parts of a city, such as major intersections, may also take advantage of being identified with descriptive names instead of a numerical identifier which would require another layer of abstraction as opposed to providing a descriptive name as an identifier directly, reducing dependencies as well as can generalize data collected on an area instead of pinpointing and identifying specific nodes. In-built monitoring would also allow for nodes to be assigned with specific roles, depending on the existing resources, as well as ensure that communication from nodes have in-built encryption and mechanisms to ensure authenticity of data and its sources. Lastly is that high-resource nodes, particularly in terms of compute capabilities, may perform in-network training, inferencing, and prediction from collected data, before transmitting the results back to the user applications, all on the edge, reducing overall network traffic on the Internet.

VI. USE CASES

This paper presents the ongoing situation with how the Internet is currently used as means for content distribution and consumption rather than the original intention for end-to-end communication, such as that of the telephony. This presents opportunities for proposed future Internet architectures and implementations, giving emphasis on the content and data

rather than the destination, which opens plenty of possibilities to look for possible solutions to these concerns. The study aims to continue the work by investigating the use of named identifiers for computing resources, as Named Networking, and develop a usable framework for Edge AI.

As a future work, the framework would be implemented and evaluated, with all the different layers and modules. Certain components from NDN, NFN, mathematical, and statistical models would be incorporated, existing TCP/IP and Application Layer protocols would be translated to adapt to the framework and named-resource paradigm, and new protocols would be developed in order to address the lacking areas as well as possible improvements to the existing protocols. This is also due to the fact that there are plenty of application and transport layer protocols built on top of IP, that may be beneficial when reconfigured with a named-resource design as well as using a data-centric paradigm.

These developments would enable the evaluation of the Named Networking Framework to take place, in order to determine its viability as a framework as well as its usability as a proposed future Internet architecture under ICN.

REFERENCES

- [1] Lovén, L., Leppänen, T., Peltonen, E., Partala, J., Harjula, E., Porambage, P., ... & Riekkki, J. (2019). EdgeAI: A vision for distributed, edge-native artificial intelligence in future 6G networks. *The 1st 6G Wireless Summit*, 1-2.
- [2] Zhang, L., Estrin, D., Burke, J., Jacobson, V., Thornton, J. D., Smetters, D. K., ... & Papadopoulos, C. (2010). Named data networking (ndn) project. *Relatório Técnico NDN-0001*, Xerox Palo Alto Research Center-PARC, 157, 158.
- [3] Sifalakis, M., Kohler, B., Scherb, C., & Tschudin, C. (2014, September). An information centric network for computing the distribution of computations. In *Proceedings of the 1st ACM Conference on Information-Centric Networking* (pp. 137-146).
- [4] Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., Claffy, K. C., Crowley, P., ... & Zhang, B. (2014). Named data networking. *ACM SIGCOMM Computer Communication Review*, 44(3), 66-73.
- [5] Park, J., Samarakoon, S., Bennis, M., & Debbah, M. (2019). Wireless network intelligence at the edge. *Proceedings of the IEEE*, 107(11), 2204-2239.
- [6] Wang, X., Han, Y., Wang, C., Zhao, Q., Chen, X., & Chen, M. (2019). In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning. *IEEE Network*, 33(5), 156-165.
- [7] Campolo, C., Lia, G., Amadeo, M., Ruggeri, G., Iera, A., & Molinaro, A. (2020, October). Towards Named AI Networking: Unveiling the Potential of NDN for Edge AI. In *International Conference on Ad-Hoc Networks and Wireless* (pp. 16-22). Springer, Cham.
- [8] Ong, A. V., & Peradilla, M. (2021, August). An IoT Framework Based on SDN and NFV for Context-Aware Security. In *2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN)* (pp. 167-172). IEEE.