Towards Edge Computing Over Named Data Networking

Abderrahmen Mtibaa*, Reza Tourani*, Satyajayant Misra*, Jeff Burke†, Lixia Zhang†
*New Mexico State University, †University of California at Los Angeles

***Abstract*—This paper highlights the architectural opportunities of enabling edge computing over Named Data Networking (NDN). We use a simple Augmented Reality (AR) use-case to illustrate NDN's architectural advantages in edge computing support, and to discuss specific solutions for efficient resource discovery, compute re-use, mobility management, and security. We also elaborate different design options and tradeoffs in addressing these questions.**

I. Introduction

In the IP-based architecture, edge computing client applications rely on centralized entities (*e.g.,* proxies and/or SDN controllers) or direct communication with all edge computing devices to make informed task-offloading decisions. These solutions require the creation of additional infrastructure in the network, introduce overheads–requests routed to the proxies or the controllers and all information from compute resources are also routed to these entities. The links connected to these entities thus become bottlenecks. The network just operate as dumb forwarders with all decisions being made at the application layer (especially with the proxy).

Information-Centric Networks (ICN) provides an alternative to traditional host-centric IP architecture which may make it a perfect candidate for provider-agnostic distributed computation such as *fog and edge computing*. In Named Data networking (NDN), which is a realization of the ICN vision, computation can be executed remotely by any node in the network which supports the required compute capabilities which makes the client and the network agnostic of the list of providers and their corresponding IP mapping.

In this paper, our aim is to discuss the feasibility of a generic NDN edge computing paradigm that leverages NDN networking and forwarding features to seamlessly offload computation efficiently. We focus on in-network compute execution in NDN and we highlight the open issues for enabling *secure*, *efficient* and *seamless* edge computing functionalities over NDN. Named Function Networking (NFN) [1] and Named-Function as a Service (NFaaS) [2] are two designs that use function naming to locate remote compute resources and perform in-network computation over ICN/NDN. However, these designs raise new challenges such as efficient resource discovery, leveraging compute re-use, mobility management, and security. We consider an augmented reality (AR) application as a use-case to highlight the following NDN edge computing high-level design challenges: (Section IV) efficient *resource discovery* in order to make accurate in-network offloading decisions, (Section V) enabling *compute re-use* for optimal resource utilization, (Section VI) efficient *mobility management*, and (Section VII) enabling *secure and private* computation for a distributed and trusted execution environment. We identify and highlight these design challenges, discuss the different options to tackle these challenges, and where possible provide solution examples.

II. Background & Related work

a) Mobile Edge Computing: Various solutions for computation offloading to more powerful surrogate machines [3], known as cyber-foraging [4], [5], have been proposed. Researchers have pointed out the significant impact of large RTT's on energy consumed by mobile devices while offloading to distant clouds, which has pushed research efforts into bringing computational resources, known as cloudlets [6] closer to mobile devices. Edge computing research has been focused on the following main challenges; task scheduling [7], and energy efficient computing [8].

While this area of research is rapidly growing, most of the existing IP based edge computing research focus on applications and services to optimize response time and energy consumption. These edge computing solutions are agnostic of the networking problems used. Problems include complex mapping between application names and IP addressess without leveraging the available network conditions. NDN, however, solves the above problem, by marrying app names to network forwarding. In this paper, we discuss leveraging the NDN in-network features that can be used for seamless and efficient edge computing. We discuss how NDN can enable seamless computing resource discovery, task forwarding, and compute re-use in a distributed fashion without relying on a centralized entity while incurring a minimum overhead. Our approach can aid the optimization envisaged today with proxies and SDN controllers and may also be strong enough to be leveraged alone.

Next, we present an NDN primer to describe its key design foundations, and the Named Function Networking (NFN) which we will build upon to discuss research challenges for NDN edge computing.

b) Named Data Networking (NDN): Different from IP networks that use IP addresses to identify where packets should move, the fundamental idea of Named Data Networking (NDN) architecture [9], [10] is to retrieve the named pieces of information (named network-layer packets), from any node that can provide it.

In NDN, routers are equipped with a content store (CS), a pending interest table (PIT), and a forwarding information

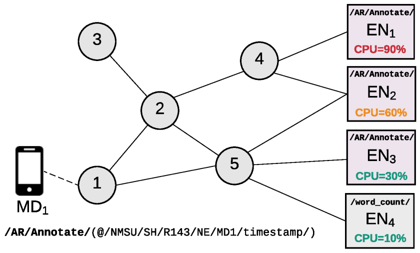

Fig. 1. Illustration of a scenario consisting of a mobile device requesting remote annotation. Three of the four ENs implement the annotation service, but EN_3 is the leat loaded edge computing node.

base (FIB).The FIB (similar to the forwarding table in IP routers) gets populated using a routing algorithm. Any node that receives an interest for a content, performs a CS lookup on the content name. If the content is not available in the CS, the node performs a lookup in its PIT to check whether there is an existing entry for the requested content. If the PIT lookup is successful, the router adds the incoming interest's interface to the PIT entry (interest aggregation) and drops the interest. If no PIT match is found, the router creates a new PIT entry for the interest and forwards the interest using the FIB to an upstream router in the direction of the data source(s).

Contents take the interests' reverse-path back to the requester. Upon receipt of a content chunk, a router forwards the chunk along the interfaces on which it had received the corresponding interest(s).

Named Function Networking (NFN) [1], [11] uses the ICN function naming to locate remote compute resources. NFN supports the resolution-by-name for functions as well as data [11]. In NFN, interests are expressions consisting of two components; a routable prefix and an appended expression. This inetrest will be first routed, as in NDN, to reach the data described by the routable prefix, then the expression can be extracted and processed to perform further computation or data fetching [12]. NFaaS, a framework which was proposed to extend NFN to support dynamic execution of code using lightweight VMs [2]. However, most of these existing solutions do not use the NDN networking features such as naming and aggregation to improve users QoE. This paper will fill this gap to build on the general concepts of NDN and NFN in order to enable efficient resource discovery and task aggregation.

III. A Typical Edge Computing Application

In this paper, we consider an augmented reality (AR) app as a use-case to highlight the design challenges and potential solutions resulting in enabling edge computing over NDN.

We consider an NDN network of nodes–some have edge computing capabilities. A mobile user who carries a mobile device (MD) aims at creating an AR view (*e.g.,* annotated view, overlaid images) of a scene, also referred to as the Field-of-View (FoV). If the compute intensity of the operations in the AR application are expensive, the MD schedules the current FoV as a compute task, which will be sent to neighboring edge compute nodes (ENs) for remote processing.

Fig. 1 depicts a scenario where a mobile device (MD_1) is requesting remote annotation of its current FoV. MD_1 is requesting the service `/AR/annotate` of the FoV `/NMSU/SH/R143/NE/MD1/timestamp/` captured at Room 143 of the Science Hall (SH) building at NMSU, by MD_1 while facing north east (NE) at time *timestamp*. Note that while the network has 4 edge nodes $EN_{i\in\{1\cdots4\}}$, only 3 ENs support the `/AR/annotate` service, and each of these three have different loads (*e.g.,* CPU usage). MD_1 must leverage NDN's naming feature to route its task to the best EN (among the ENs supporting the requested service).

In the following, we will discuss how to enable seamless task forwarding? What constitute the best EN candidate? What are the design and engineering challenges for enabling edge computing over NDN?

IV. Resource Discovery For Edge Resources

Resource discovery (RD) is one of the main challenges in edge computing. In particular, ENs often have a rapidly changing available compute resources (high compute loads, concurrent clients, and limited resources). These resource availability statuses must be discovered by intermediate nodes and/or MDs to make efficient task forwarding decisions.

As shown in Fig. 1, ENs must disseminate (1) the services they support (*i.e.,* `/AR/Annotate/` and `/Word_count/`), and (2) their resource utilization (*e.g.,* CPU, GPU, memory, storage, and energy). One way to disseminate this information is to make them part of routing message exchange (*e.g.,* using NLSR [13]). In our scenario (Fig. 1), intermediate nodes in the network will set EN_3 to execute the service `/AR/Annotate/` as per its lowest CPU utilization.

We identify three approaches to perform neighboring resource discovery (RD); proactive, reactive, and passive RD approaches. (i) **Proactive RD:** In the proactive approach, ENs will proactively advertise, every time period δt, their resources using an NDN routing protocol (*e.g.,* NLSR). This approach can guarantee a unified view of the entire network and a timely access of service forwarding entries at any node. However, depending on the period set, this approach can lead to the use of stale information (*i.e.,* if δt is large), or a large overhead in the network (*i.e.,* if δt is too small). We can propose a dynamic period setting based on the load at each EN. (ii) **Reactive RD:** In the reactive approach, MD_1 sends requests for the service `/AR/Annotate/` to all ENs. On receiving these requests, ENs reply with their load, which will update the FIB entries at the nodes in the reverse path (*i.e.,* Nodes 1, 2, 4, and 5 in Fig. 1). This approach generates an overhead proportional to the load in the network making it suitable only for low frequency tasks. While NDN (unlike TCP) does not have a notion of sessions, one can set a time-based (*e.g.,* all sessions last 10s) or a context-based sessions based on the location of the device; for instance when FoV names change from `/NMSU/SH/R143/*/` to `/NMSU/SH/R144/*/`. (iii) **Passive RD:** A passive approach rely on NDN forwarding plane to send negative acknowledgment (NACK) when an EN receiving a given task is overloaded. On receiving such NACKs, in-network routers avoid sending more requests to overloaded EN, and re-route the task to another EN candidate.

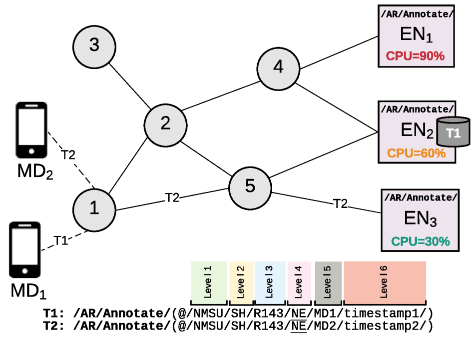

Fig. 2. Scenario highlighting a potential compute reuse if T_2 can be routed to edge node EN_2

V. Compute Reuse or Task Aggregation

In the context of the image annotation service, users connected to the same access point (*i.e.,* a periphery NDN node) will likely share overlapping FoVs; for instances, users attending the same event (*e.g.,* music concert or visitors attending the Louvre museum). If these overlapping FoVs will be forwarded to different edge computing nodes, a potential compute reuse and considerable speed-up in task execution time will be unused. Fig. 2 highlights a scenario, where MD_2 sends a potentially overlapping FoV to the best edge computing node, EN_3 which has the lowest CPU usage. Applying the resource discovery mechanism, described in the previous section, nodes will be oblivious of the existence of a potentially overlapping task, T_1, at EN_2.

In order to take advantage of compute reuse, we propose a new forwarding strategy algorithm that stores fingerprints (*e.g.,* hash of the task names) of the tasks at the routers for a given freshness period Δt. In addition to the PIT and FIB tables, this forwarding strategy requires the implementation of a forwarding task base (FTB) table which consists of task name, say T_1, and the interface used to forward T_1, say i_1. When a new task arrives at a router v, v performs FTB lookup and computes an overlapping score. For instance, in Fig. 2, on receiving T2, Node 1 finds that the similarity score between T1 and T2 is high, and forwards T2 through i_5 to Node 5, which will also forward the task to EN_2. Note that nodes forwarding the task will also set entries in their PIT tables as described in Section II to set the reverse path for the results to reach the requesting MD.

The similarity score can be measured as the longest prefix match of the task names. For instance in Fig. 2, the similarity score between T1 and T2 will be the highest matching level, *i.e.,* 4, as both task names share the longest prefix `/NMSU/SH/R143/NE/`. This similarity score can be combined (using an objective function) with the cost set at the FIB table to make a more efficient forwarding decision. Therefore Node 1 and all nodes in the path may choose to forward T2 towards E_2, which offers a good tradeoff between cost and compute reuse instead of forwarding to E_3, which has the lowest CPU usage. Appropriate and autonomous naming scheme is required to efficiently leverage compute-reuse in this way. Mobile applications can leverage the sensors in the MDs to get accurate indoor localization and orientation.

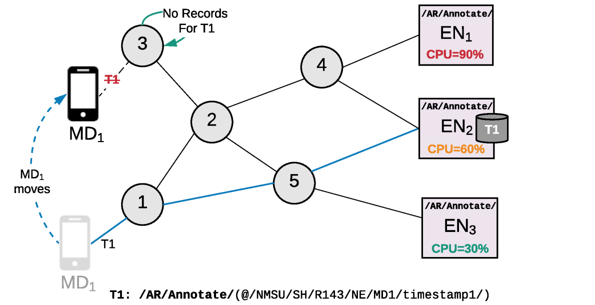

Fig. 3. Scenario highlighting the mobility management challenge; MD_1 moves to a new location before receiving the task execution results. .

The EN that receives a potentially overlapping FoV feed compares it with other potential overlapping candidates using image processing features, such as background removal [14], object detection [15], image alignment, and stitching [16]. Therefore, ENs extract non-redundant subtasks for annotation (*e.g.,* a moving object within a pre-annotated background scene), and perform stitching and aggregation upon receiving all results from all dependent tasks.

This method has the following limitations: (i) if Δt is large, the FTB lookup can be slow resulting in a slow down of the total task completion time, and (ii) this method can be used by malicious nodes for DoS attacks and/or FTB poisoning. We believe that (i) can be mitigated using hashing and dynamic Δt assignment. We will discuss (ii) in Section VII.

VI. Mobile Device Mobility Challenge

In NDN, clients are responsible for re-requesting data upon timeouts. In the context of edge computing, tasks can be compute intensive and execution times can be hard to estimate, resulting in longer timeouts. How can we manage MD mobility and minimize the task response time?

Fig. 3 illustrates a scenario in which MD_1 looses connectivity with Node 1 before receiving the result of its task T1. In NDN, data packets will travel across the reverse path to the destination. Moreover, MD_1 can request the data results of its task from the new neighbor Node 3, however node 3 will not find records for a pending task T1 or which edge computing node is responsible to execute the task.

One solution to this mobility management challenge can result in a two steps proactive approach consisting of: (i) the MD keeps track of the link breakage with its corresponding periphery node; (ii) if the MD looses connectivity, it re-sends the pending interest. As per our FTB tables design, intermediate nodes closer to the edge computing nodes will find an exact match of the task and forward this new request to the EN currently executing this task or to any cache node in the reverse path that has the result for the task (in case this task is already executed). However, if the NDN network is sparse and ENs are scattered across the network, the task may not reach the task executing EN. Therefore, we can add a flag to the interest which will be set if the corresponding interest is a re-sent. This flag will be checked at the intermediate nodes; (1) if an intermediate node does not have a full match of the task in its FTB then it broadcasts it on all interfaces, and (2) if an intermediate node has the exact task in its FTB, then it will forward it to the executing EN candidate.

This solution which involves exchange of broadcast messages can lead to; (i) large overhead in the network, and (ii) malicious use of the broadcast messages. In order to reduce such overhead, we can limit the TTL for these messages to one or two hops assuming that MDs will move to adjacent locations which can be directly connected to the previous periphery node.

VII. Access Control & Security

The security challenges include client privacy, Distributed Denial of Service (DDoS) vulnerability, and effective access control enforcement. In this section we discuss two of these challenges, naming client privacy and DDoS vulnerability, which are more relevant in the context of edge computing. As for the access control enforcement, we refer the readers to a comprehensive survey [17] on this matter.

a) Privacy Threats: In NDN, a client requests content by explicitly expressing the content's name in the interest packet. In our design, the information exposure is not limited to the content name; a request also exposes the location information and the computational operations that should be executed on the content. The state-of-the-art in NDN anonymous communication suggests using proxy-based approaches, in which clients privately interact with a network of proxies (by encrypting the discerning names) to evade censorship and information exposure. The proxies de-cloak the names for leveraging the in-network caching and request aggregation. This secure communication using encrypted requests between the clients and proxies, if applied to the EC scenarios undermines task aggregation as well as leveraging of results in caches. It remains an open challenge how to design a mechanism the allows network entities to leverage NDN features while preserving clients privacy. One approach to explore is location abstraction to prevent fine-grained location identification.

b) DDoS Attack: As discussed earlier, malicious nodes can leverage the use of broadcast messages and/or the FTB tables to perform DoS or DDoS attacks. For EC, we divide the DDoS vulnerabilities into two categories. First vulnerability arises from a set of users requesting various content (either existing or fake content) at high rates with the objective of exhausting the available resources (*e.g.,* CPU or storage resources) at the intermediate routers. The second DDoS attack occurs when a set of users request computationally-demanding tasks from ENs with the objective of exhausting the available resources at ENs. While the former attack has been extensively discussed in the NDN community, the latter is specific to EC.

One naive solution to the latter vulnerability requires the routers to consider the ENs' residual resources in their forwarding decisions. More specifically, a router may perform an implicit rate limiting approach by not forwarding tasks to an overloaded EN, instead, forwarding it to another EN with more available resources. While this approach does not completely solve the DDoS attack on ENs, it limits its snowballing effect.

VIII. Concluding Remarks

In this paper, we have discussed leveraging the Named Data Networking (NDN) features to facilitate edge computing functionalities. We have presented the steps needed and the challenges that arise when using NDN edge computing. Challenges include efficient resource discovery, enabling compute reuse, mobility management, and ensuring secure and trustworthy task offloading. This work represents one of the first attempts to use the NDN networking features to improve computing in a data communication-driven NDN design. Additional issues and design challenges remain unexplored such as scalability, robustness/resilience, resource poverty, energy efficiency, and load balancing, and energy consumption which will be considered in our future work.

References

[1] C. Tschudin and M. Sifalakis, "Named functions and cached computations," in *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*, pp. 851–857, IEEE, 2014.

[2] M. Król and I. Psaras, "NFaaS: named function as a service," in *Proceedings of the 4th ACM Conference on Information-Centric Networking*, pp. 134–144, ACM, 2017.

[3] J. Flinn, "Cyber foraging: Bridging mobile and cloud computing," *Synthesis Lectures on Mobile and Pervasive Computing*, vol. 7, no. 2, pp. 1–103, 2012.

[4] R. Balan, J. Flinn, M. Satyanarayanan, S. Sinnamohideen, and H.-I. Yang, "The case for cyber foraging," in *Proceedings of the 10th workshop on ACM SIGOPS European workshop*, 2002.

[5] R. K. Balan, D. Gergle, M. Satyanarayanan, and J. Herbsleb, "Simplifying cyber foraging for mobile devices," in *ACM MobiSys*, 2007.

[6] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *Pervasive Computing, IEEE*, vol. 8, no. 4, pp. 14–23, 2009.

[7] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *Proceedings of the sixth conference on Computer systems*, EuroSys '11, (New York, NY, USA), pp. 301–314, ACM, 2011.

[8] A. Mtibaa, A. Fahim, K. A. Harras, and M. H. Ammar, "Towards resource sharing in mobile device clouds: Power balancing across mobile devices," in *Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing*, MCC '13, pp. 51–56, 2013.

[9] "NDN." http://named-data.net/.

[10] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, *et al.*, "Named data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, 2014.

[11] M. Sifalakis, B. Kohler, C. Scherb, and C. Tschudin, "An information centric network for computing the distribution of computations," in *Proceedings of the 1st international conference on Information-centric networking*, pp. 137–146, ACM, 2014.

[12] H. Zhang, Z. Wang, C. Scherb, C. Marxer, J. Burke, L. Zhang, and C. Tschudin, "Sharing mhealth data via named data networking," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, pp. 142–147, ACM, 2016.

[13] A. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang, "NLSR: named-data link state routing protocol," in *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, pp. 15–20, ACM, 2013.

[14] V. Mazet, C. Carteret, D. Brie, J. Idier, and B. Humbert, "Background removal from spectra by designing and minimising a non-quadratic cost function," *Chemometrics and intelligent laboratory systems*, vol. 76, no. 2, pp. 121–133, 2005.

[15] S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed, "Moving object detection in spatial domain using background removal techniques-state-of-art," *Recent patents on CS*, vol. 1, no. 1, pp. 32–54, 2008.

[16] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International journal of computer vision*, vol. 74, no. 1, pp. 59–73, 2007.

[17] R. Tourani, T. Mick, S. Misra, and G. Panwar, "Security, privacy, and access control in information-centric networking: A survey," *IEEE Communications on Surveys and Tutorial (accepted)*, 2017.