Performance Evaluation of Secure and Privacy-preserving DNS at the 5G Edge

Yacoub Hanna*, Diana Pineda*, Kemal Akkaya*, Abdullah Aydeger†, Ricardo Harrilal-Parchment*, and Hamdah Albalawi†

*Advanced Wireless and Security Lab, Florida International University, Miami, FL USA 33174
Email: {yhann002, dpine033, kakkaya, rharr119}@fiu.edu

†Dept. of Electrical Engineering and Computer Science, Florida Institute of Technology, Melbourne, FL USA 32901
Email: aaydeger@fit.edu, halbalawi2019@my.fit.edu

***Abstract*—With the improved network performance and efficiency, 5G has been a very appealing alternative for various applications and devices, including but not limited to Industrial IoT (IIoT) applications. However, since IIoT applications require real-time transmission guarantees for time-critical data, optimizing the 5G network performance has attracted much research recently by pushing critical services to the 5G edge. One of such services is Domain Name System (DNS) which is typically offered by ISPs to serve 5G networks. However, due to its heavy role on Internet traffic, DNS has seen many attacks in the past in terms of its authenticity of records and exposure of user requests. Therefore, there have been many variants of DNS protocol such as DNSSEC and DNS over TLS (DoT) to address these security and privacy issues. As such these new protocols need to be integrated into 5G edge and their overhead should be investigated for their potential use in time-critical applications. In this paper, we consider deploying secure and privacy-preserving DNS to the 5G base stations and investigate the feasibility and performance of such an edge computing approach. To this end, we utilize SDN capabilities and forward packets to a local SDN controller for extracting and processing DNS queries. Thus, the DNS packets are served at the 5G edge with optional security and privacy features without being forwarded to remote ISP servers. Our approach is implemented in a virtualized 5G testbed network, and we present various DNS performance results via different metrics and scenarios.**

***Index Terms*—5G, DNS, DNSSEC, DoT, SDN, Edge computing**

I. Introduction

Following 2G, 3G, and 4G/LTE, 5G is the latest development in wireless networking technology. With capabilities such as faster speeds, reduced latency, and enhanced coverage and security, it represents a huge advance in terms of speed, connectivity, and capacity and has the potential to revolutionize data communication as well as the interaction with technology completely. Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communications (URLLC), and Massive Machine-Type Communications (mMTC) are just a few of the numerous new applications that are enabled by 5G revolution.

The above characteristics make 5G an appealing choice for enabling real-world solutions supporting a wide range of time-sensitive applications [1] in various domains such as IIoT, Cyberphysical Systems (CPS), and Metaverse [2]. These time-sensitive applications require any additional delays to the communications to be minimized while being secured against cyber attacks. One such factor in almost all communications is the Domain Name System (DNS) service which is a hierarchical system that maps domain names to IP addresses. DNS resolvers are responsible for querying the authoritative servers to obtain the IP address associated with a requested domain name [3]. DNS queries that are generated and passed to a 5G network are normally forwarded to a remote DNS resolver hosted by a Telecom operator or ISP, which might be located miles away within the base station. Thus, a closer DNS resolver has been very helpful in reducing network latencies while using the minimal computing power at the edge [4].

While it is an interoperability challenge to integrate DNS service at the edge within the 5G protocol stack, this is not enough as we experience new challenges regarding the security and privacy of this widely used service. For instance, there is a widespread DNS vulnerability known as DNS spoofing that enables an attacker to alter DNS responses while they are being sent to the users. Changes made by an attacker to a single server's DNS tables will spread throughout the Internet to various users via such a vulnerability. Furthermore, there have been recent concerns over the confidentiality of the DNS responses since the traditional DNS packets have been transmitted over clear text. Any network router on the route of the DNS request or response would obtain information about the websites that a user is visiting; exposing his/her privacy. Therefore, authenticating data origin and protecting data integrity and confidentiality are necessary for DNS security. Fortunately, the problems listed have already been investigated by other researchers in the past, and there are several popular protocols available for improving DNS security, including DNS Secure (DNSSEC) and DNS over TLS (DoT).

Nevertheless, there has not been much consideration of their integration and analysis for 5G networks, especially for time-sensitive communications occurring on 5G channels. In addition to their advantages and disadvantages, DNS security

protocols vary in performance based on network conditions and hardware resources. Hence, in this paper, we propose integrating secure and privacy-preserving DNS protocols, specifically DNSSEC and DoT, into the DNS resolver at the 5G edge to investigate the performance of DNS in this context and demonstrate their feasibility to be used in specific applications. As opposed to existing studies that recommend DNS services at the edge but do not offer any actual implementation and testing, our work also deals with engineering challenges of such integration.

It is important to note that integrating the DNS resolver to the 5G edge is not a straightforward approach due to how the packets are handled and tunneled via the 5G network. In other words, there is a need for an intervention mechanism to be able to recognize the content of the packets as early as possible after they leave a User Equipment (UE) device. To this end, we chose the base-stations as our deployment medium for DNS service. However, to be able to utilize the DNS resolver, we needed to employ an SDN service at the 5G Base Station. A local on-site SDN Controller is specifically used for unwrapping the packets and responding to any DNS query as long as it is in the cache of the server.

Our main contributions in this work are as follows: (1) We propose a DNS resolver to the 5G edge, which is capable of the transmissions of DNSSEC and DoT protocol packets; (2) We consider SDN Controller that is employed at the 5G base-stations, which facilitates the DNS packet handling (i.e., responding to a query locally); (3) We implemented the proposed approaches into a virtualized 5G testbed with open source tools and 5G RAN and Core codes; (4) We compared the performance of our local DNS approaches against traditional DNS service and provided a summary of what kind of network traffic (i.e., time-sensitive) the 5G environment would be able to support, considering the delay requirements.

The rest of the paper is organized as follows. The next section provides the related work that focuses on services to the 5G edge. Section III describes the preliminaries and background on the use of concepts throughout the paper while also defining the problem. In Section IV, we explain our approach in detail. Section V is dedicated to the performance evaluation of the approach and the presentation of the results. Finally, Section VI concludes the paper and discusses our planned future work.

II. Related Work

This section summarizes the research efforts that investigated secure DNS integration and placement within the infrastructure of 5G.

The main work reported in this context [5] describes a Multi-Access Edge Computing (MEC) solution developed by a mobile network operator for private 5G networks. Unlike our work, their system is situated between the mobile base station and the mobile core network, providing low-latency processing, high-bandwidth connectivity, and efficient resource utilization for real-time applications. The system is suitable for both non-standalone (NSA) and standalone (SA) 5G networks. Moreover, in this work, the authors point out that a local DNS module was implemented and the mapping between a domain name and its IP address can be learned automatically. Nonetheless, there was no type of security or privacy considered for DNS.

Another empirical study [6] discusses Customer Edge Switching (CES) and its dependency on the DNS. A lack of DNS encryption and authentication makes the current implementation of CES vulnerable to man-in-the-middle (MitM) attacks. The paper suggests implementing DNSCrypt and DNSSEC in CES to address this issue. Experimental results indicate that DNSCrypt and DNSSEC are effective at preventing attacks, with only minor delays in DNS exchanges. However, there was no actual implementation using secure DNS and DNSCrypt in a 5G environment.

Another closely related work is reported in [7], which proposes enhancements to DNS to address the needs of a distributed MEC environment. The authors present three scenarios for using a distributed environment in MEC systems and outline three requirements: selecting the optimal service, minimizing interruption time, and providing enhancements during deployment. The paper proposes various solutions to facilitate connectivity in a distributed MEC environment, including Enhanced DNS, HTTP redirection, UE and Device application interfaces, Virtual IP, and Edge DNS server solutions. These solutions aim to enhance the current DNS support to meet the needs of a distributed MEC system and improve the Quality of Service (QoS) and Quality of Experience (QoE) for users. However, there was no implementation and evaluation of any versions of DNS towards the MEC environment.

As summarized, there are several works that attempted to utilize edge services for improved DNS performance. However, none of these works have successfully integrated DNSSEC or DoT within a gNB and evaluated their performance in a real 5G testbed environment. Therefore, to the best of our knowledge, this is the first work to successfully combine these technologies and security/privacy to create an operational framework that other researchers can use.

III. Preliminaries

In this section, we give a short description of the fundamental concepts and technologies that we use in our work.

A. 5G

A few advantages of 5G include increased connectivity, lower latency, quicker transmission speeds, more connected devices, and virtual networks. Direct access to files, programs, and remote applications from the cloud is made possible by the transmission speeds, which can approach 15 or 20 Gbps, without the need for internal memory or multiple CPUs [8]. The monitoring of industrial machinery, logistics, medical treatments, and remote transportation can all be improved by lowering latency and enabling real-time execution of remote tasks. Smart cities will be able to function independently in the future to the rise of connected devices. Additionally, network slicing can build sub-networks that are suited to particular

requirements and priorities, enabling emergency connections to be given priority over other users and preventing network overloads.

B. 5G User Plane Handling

When a UE IoT device queries the DNS on 5G networks, the request is enclosed in an IP packet and transmitted to the DNS resolver. Therefore, it is essential to comprehend the fundamental elements of 5G networks, especially the User Plane (UP). The UP domain connects many network functions to enable its core role, which is to carry and deliver data between the UE and the Data Network (DN). Mobile devices can connect to the 5G network through the initial UP connection between the UE and the gNB. The following link is made via the N3 reference point from the gNB to the UPF in the core network. Using the N9 reference point, it is possible that there will be more hops made between UPFs in the core network. The final link runs across the N6 reference point between the DN and UPF. The GTP - User Plane (GTP-U) tunnel carries User Plane data between the N3 and N9 reference points as shown in Fig. 1. A reference point, which is used in 5G networks, defines the logical connection between network services and provides interoperability and consistency across several mobile network generations.

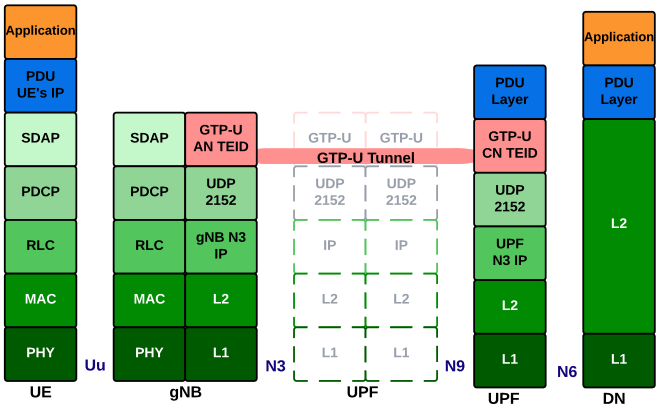

Fig. 1. User Plane Protocol Stack for 5G [4]

C. DNS

DNS is the system that maps human-readable domain names (e.g., www.mywebsite.com) to IP addresses (e.g., 192.168.0.1). It works by having a hierarchical structure of servers, where each server is responsible for a portion of the domain namespace. When a client wants to resolve a domain name, it sends a query to a DNS resolver, which then contacts authoritative servers from the top of the hierarchy down until it finds the IP address for the requested domain name, as shown in Fig. 2. To this end, a stub resolver, which runs on a network node, searches its local DNS cache and, if needed, generates and forwards the DNS query to a recursive resolver. A recursive resolver, usually run by the Internet Service Provider (ISP), provides address resolution service and can officiate a level of control over the traffic of its client base.

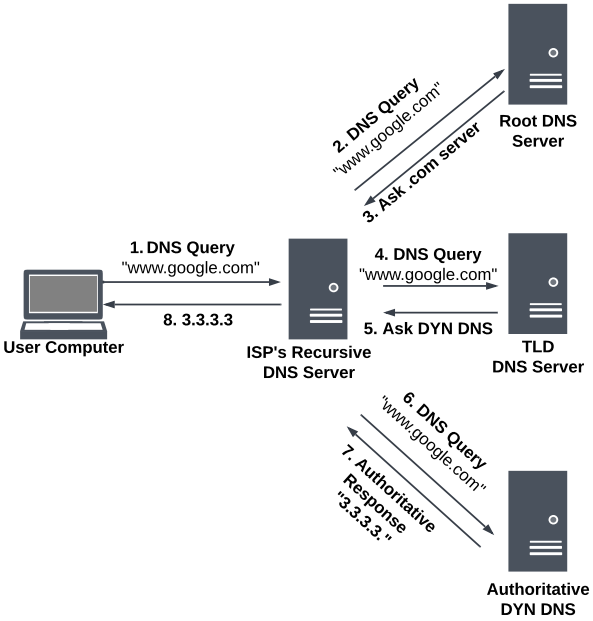

Fig. 2. DNS protocol steps

If the requested domain name is not found in the DNS recursive resolver's cache, a DNS query is forwarded to one of the 13 Root servers [9]. The Root server provides the address of TLD name servers in which the location of the information about the domain name server will be obtained. As the last step, the DNS query will be sent to the Authoritative name server for the requested domain, where the DNS response will be generated and sent back to the stub resolver eventually.

In 5G networks, the DNS query from a UE is tunneled through the base station directly to the UPF, which then routes the query to the target DNS server. Once received by the DNS server, if the requested address is not in the cache, the query response time varies depending on the location of the authoritative name servers of the requested domain. Even if the network core may employ a DNS cache locally at the UPF, this still requires access to UPF, which is typically far away from the UE devices.

D. DNS Security

DNS was not originally designed with security in mind, and some security issues were discovered in 1990 regarding the authenticity of the responses received from DNS servers. In 1997, DNSSEC was introduced to address these issues and updated in 2005 to further ensure DNS accuracy and authenticity. In DNSSEC, digital signatures are used to verify the accuracy of DNS records, allowing users to verify the correctness of information supplied by DNS servers. Domain owners sign DNS records with private keys to establish a trust chain between them and the DNSSEC hierarchy's root key. Using this chain of trust, DNS resolvers verify the digital signature of requested DNS records, ensuring that the data has not been modified. DNSSEC's steps are displayed in Fig. 3.

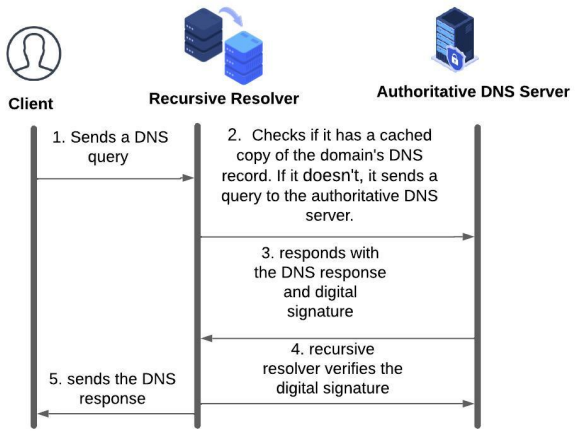

Fig. 3. DNSSEC Message Exchanges

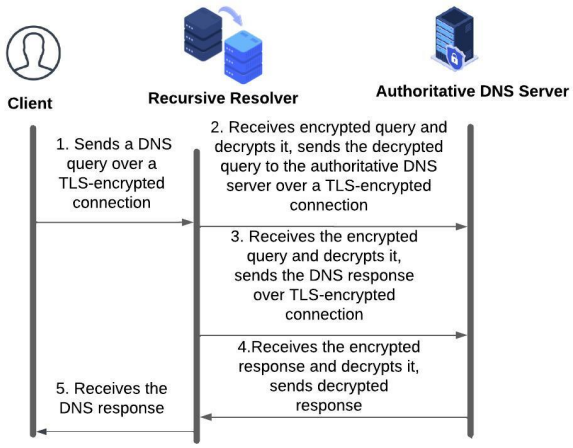

Fig. 4. DoT Message Exchanges [12]

E. Privacy-preserving DNS

Another main issue with the DNS service was the privacy of the users. Even though a DNS service might be secure against any authentication attacks, it still exposes the IP address of the user and the websites accessed by the user. This created a lot of concern among users lately since ISPs could use or sell this information to other third parties for advertisement purposes. To this end, different privacy-preserving solutions were considered.

For instance, Transport Layer Security (TLS) protocol encrypts data when it is transmitted over the Internet to establish a secure communication channel and enable authorized parties to confirm their identities. In 2016, the DoT protocol was introduced to encrypt DNS queries and responses by adding a TLS encryption on top of the User Datagram Protocol (UDP) which is typically used for DNS queries. To establish a connection, the client initiates a TCP connection to a designated DoT-enabled resolver port TCP/853, followed by a typical TLS handshake. Nevertheless, DoT faces practical challenges, such as being less recognized by the security community and less resilient to packet losses. Despite these limitations, DoT remains an essential candidate and is implemented and tested [10]. Moreover, one of the widely used open-source DNS software that provides domain name resolution services as well as supporting DoT is Berkeley Internet Name Domain (BIND). BIND allows to generate or obtain a TLS certificate and key for the DNS server and it uses these credentials to establish an encrypted connection with DNS clients [11].

IV. Proposed Approach

In a 5G network simulation, our approach uses SDN capabilities to extract GTP-U-encapsulated DNS queries from the UE. Then, instead of depending on the UPF to route the request to an external DNS server (e.g., Google DNS, Cloudflare DNS), we forward the DNS query to our local DNS server. as illustrated in Fig. 5, we placed the SDN controller at the gNB to allow real-time GTP-U traffic analysis. The primary purpose is to use network function virtualization to do quicker processing locally at the gNB.

We are able to process the GTP-U network traffic in real-time using the centralized and southbound application capabilities of the SDN to gather the necessary data from a UE DNS query, send it to the local DNS server, and forward the response back to the UE by adequately configuring the rules at the SDN controller. Therefore, installing the SDN switch at the gNB and routing the packets to the SDN controller is required to enable this operation. The general architecture for this method is illustrated in Fig. 5.

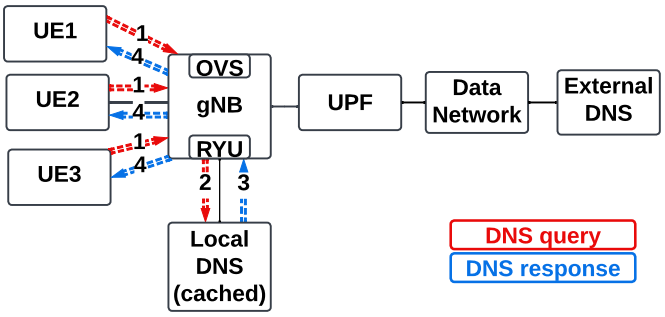

Fig. 5. General architecture and approach for processing DNS queries locally.

A. SDN Switch Phase

The initial step in our strategy is to create the following three flows to the gNB's SDN switch:

- A flow to send GTP-U traffic to the SDN controller for further analysis (outbound UE DNS queries);
- A flow to send DNS responses from the edge/local DNS server to the controller for handling (inbound DNS responses from the edge);
- A flow to forward all other traffic normally.

Using OpenFlow, it is possible to send the frames to the controller as `packet_in` events for further analysis and also to send the new frames to the SDN switch as `packet_out` events for retransmission.

It is important to note that the current technique is not optimal for real-world implementation because an additional overhead is associated with delivering the frames to the controller, having the controller process them, and instructing the SDN switch. In the ideal scenario, the SDN switch would be able to handle these frames directly, thus reducing the overall latency/overhead experienced. However, this would require more advanced programming capabilities at the SDN switches, which are expected to become more accessible. Nevertheless, as the studies proved, keeping everything local with gNB would save us considerable time, as shown in the experiments.

B. SDN Controller Phase

At the SDN controller, we further analyze the received GTP packets or packets from the local DNS server, extract the necessary information, and redirect the traffic toward its destination. As shown in Fig. 6, there are three main stages that constitute the controller's overall workflow:

- A *learning* stage in which the controller examines GTP-U traffic to build a database of linked UEs and corresponding tunnels;
- A *query* stage in which the controller separates the DNS query from the GTP-U packet and sends it to the edge DNS server;
- A *response* stage, where a DNS response coming from the edge DNS server is encapsulated in a GTP traffic by the controller and injected into the corresponding GTP tunnel between the UE and the UPF.

Following are the details on these stages:

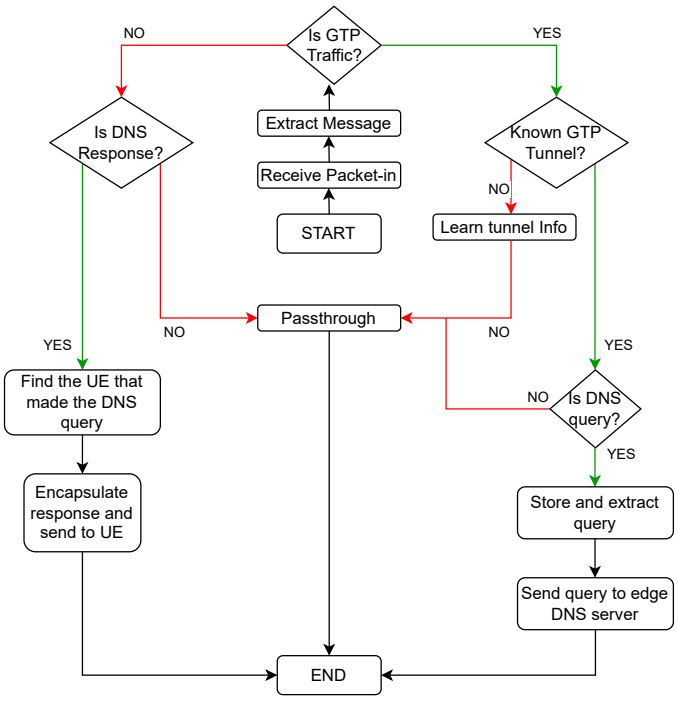

Fig. 6. SDN Controller Workflow

1) Learning stage: When a GTP-U packet arrives as a `packet_in` event at the SDN controller, the controller divides the packet into individual layers: the GTP-U header, GTP-U extension headers, tunneled IPv4 header, subsequent transport layer protocol, and final payload. The extracted layers are analyzed to specify the key parameters, such as the UE IP address, GTP Tunnel Endpoint Identifier (TEID), and GTP tunnel type. The controller utilizes the gathered data to create an internal database that lists the UEs, their corresponding tunnels, and the type of communication (uplink or downlink)

2) Query Stage: After collecting adequate data to identify the downlink tunnel for a specific UE, if the controller later recognizes a DNS query originating from that UE in the uplink tunnel, it extracts the query and keeps any critical parameters, such as the requesting UE and DNS transaction ID (As shown on the left branch in Fig. 7). Then, the controller crafts new transport, network, and Ethernet layer headers for the query and instructs the SDN switch to forward the frame to the local edge DNS, as shown in Fig. 7. Finally, the DNS server then accepts the query and processes it normally, serving out a cached answer if one is already available or requesting and caching an authoritative response in the absence of it.

3) Response Stage: In this stage, the DNS response is extracted and compared to the stored query by the controller when it arrives at the controller (illustrated on the right branch of the flow diagram in Fig. reffig:flowchart). The information in its database will be used to correctly wrap the response directed toward a particular UE. Then, the controller generates new transport, network, and GTP-U with associated extension headers and an extra set of transport, network, and Ethernet layer headers for the response. Subsequently, the controller instructs the SDN switch to deliver the frame to the gNB, which forwards the message to the corresponding UE.

C. DNS Security

After establishing our different stages to managing DNS queries at the edge, we also implemented DNSSEC and DoT as two additional security measures for our local DNS server with the primary goal of providing a DNS infrastructure that is secure and trustworthy.

1) DNSSEC: We implemented DNSSEC in our deployed edge DNS server to improve the security and integrity of the DNS infrastructure. The principal objective is to ensure that DNS responses are protected from tampering and that DNS data can be verified as authentic. Our proposed DNSSEC approach consists on the following steps:

- Generate two cryptographic key pairs: Key Signing Key (KSK), which will be used for signing the bulk zone, and Zone Signing Key (ZSK) to sign the DNSKEYs.
- Sign the zone (signing the resource records) using the generated private key. This process ensures that the records' integrity is maintained and any alterations can be detected.
- Make the public keys accessible to DNS resolvers that want to access them. These public keys are represented

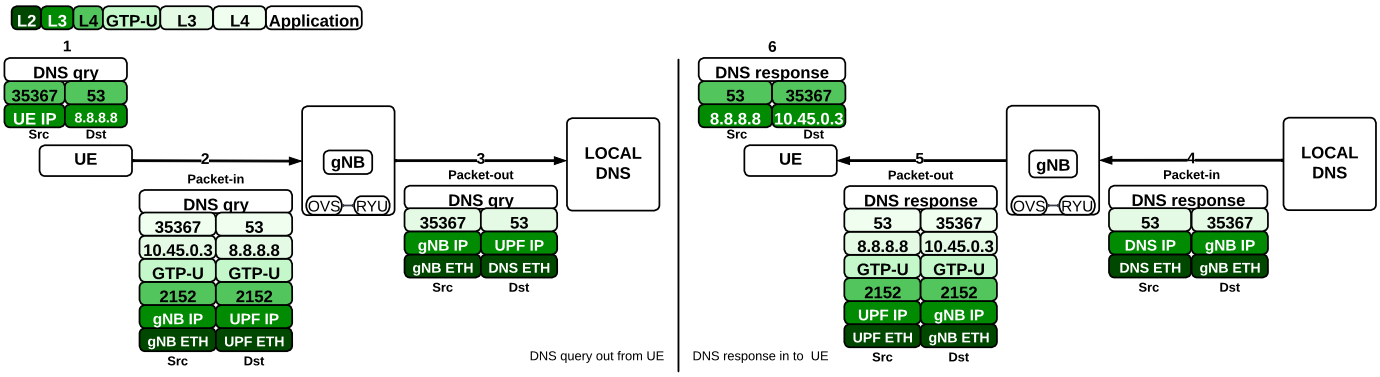

Fig. 7. Implementation of Local DNS - Query and Response [4]

as DNSKEY records, which resolvers can access when making DNS requests.

- Configure the delegation signer (DS) records in the parent zone to create a chain of trust from the root zone to each individual DNS zone. The hash values of the DNSKEY entries are included in these DS records, allowing parent zones to check the reliability and authenticity of the child zones. legitimacy and integrity of the child zones.

By deploying DNSSEC to our approach, we are ensuring that the received DNS responses are secure and can be verified as authentic. This enhances the security and dependability of our DNS infrastructure by preventing DNS spoofing and other harmful operations.

2) DoT: In addition to DNSSEC, we have added DNS over TLS (DoT) as an additional security feature for our local DNS server. By tunneling DNS traffic via a TLS (Transport Layer Security) connection, DoT offers encryption and anonymity to DNS requests and answers, which can be important to the 5G network subscribers. This guarantees the confidentiality and security of the information sent between the UE and the DNS server. Our implementation of DoT involves the following steps:

- Generate certificates using three different cryptographic algorithms: RSA, ECDSA, and EDDSA. These certificates are intended for our DNS server to establish secure TLS connections and also measure the variance in latency introduced by implementing each of them.
- Configure the edge DNS server to listen for incoming DoT connections on our specified port, which is 853.
- Enable TLS support in our DNS server by configuring the private key and the corresponding certificate.
- Validate the client certificates to ensure that only authorized clients can establish a secure connection to our DNS server.
- Use the TLS connections that are encrypted for communication with DNS clients, where DNS queries and responses are encapsulated within the TLS tunnel.

By using DoT, we ensure that all DNS communication between the UE and the local DNS server is encrypted and secured against unauthorized access or tampering. As a result, there is less chance of DNS spoofing, DNS hijacking, and other attacks that try to snoop on or alter DNS traffic.

V. Performance Evaluation

This section summarizes the experiment setup, metrics, and performance evaluation results.

A. Setting up the 5G testbed Environment

The proposed approach has been deployed and tested in a 5G-SDN open-source testbed, which is a modification of the found approach in the documentation provided in [13]. This testbed includes different open-source projects that replicate the Control Plane, User Plane, gNB, and UE components of the 5G networks and a local DNS server. These open-source projects simulate the 5G networks and DNS edge environment are Open5GS as the Core Network, UERANSIM for the 5G UE and gNB, and Bind9 as the DNS server in addition to OpenvSwitch [14] as the SDN switch and Ryu [15] as the SDN controller.

Our virtual setup is distributed into two machines: A Dell Laptop and a Dell Precision Workstation, with VMWare ESXi as a bare-metal hypervisor. Within the ESXi environment, we created three VMs, including a gNB, UE, and DNS server. Additionally, the gNB VM contains an individual OpenvSwitch instance to intercept frames before they enter the 5G RAN and extend SDN capabilities to the gNB, thus completely containing our edge functions at the gNB, and an RYU SDN controller for real-time traffic analysis. At the same time, on the Dell laptop, we deployed the CP and UP components of the 5G core network without utilizing Virtual Machines, as shown in Fig. 8.

B. Metrics and Benchmarks

We considered two metrics for the evaluation of DNS overhead:

- **Query Latency:** This is the time it takes to get a response for a DNS query that comes from the caches.
- **Response size:** This metric indicates the size of the response in terms of bits.

TABLE I
AVERAGE LATENCY RESULTS FOR DNS RESPONSES UNDER DIFFERENT APPROACHES.

	Average Latency (ms)							
	Local DNS - With SDN				Public DNS - W/O SDN			
	DNS	DNSSEC	DoT	DNSSEC + DoT	DNS	DNSSEC	DoT	DNSSEC + DoT
internetsociety.org	5.6	15.3	68.8	69	33.5	55.7	105.1	129.9
dnssec-tools.org	6.07	16.7	66.6	71	118	192.2	198.1	210.9
dnssec-deployment.org	5.47	16.1	67.4	68.9	35.4	59.5	101.9	163.2
kumari.net	5.33	14.6	67.9	69.9	77.2	142	174.1	239.9
huque.com	5.13	15.4	67.7	68.2	110.7	177	170.3	184.3
ortzmeyer.org	5.27	14.8	68.8	69.9	188.8	257.4	266.3	456.5
afnic.fr	5.3	15	67.2	68.2	95.4	108	185.1	330.3
netfuture.ch	5.3	14.9	68.4	70.4	560.2	689.5	739	974
AVERAGE	5.43	15.35	67.85	69.44	152.40	210.16	242.49	336.13

We considered all the approaches discussed previously as our benchmarks with respect to a regular DNS query. Specifically, we considered various DNS security options at the edge as follows: 1) **DNSSEC**; 2) **DoT**; and 3) **DNSSEC + DoT**.

We also compared these with the option where DNS server is at a remote (default) location. Additionally, we ensure that that the DNS queries have been cached already by DNS servers by running multiple queries and reporting the final one. Note that if a DNS record is not available at the local cache, it will trigger a DNS lookup for the first time which happens only rarely.

C. Experiment Setup for DNS

To measure each scenario's latency and response size, we used the `kdig` command to send DNS queries to our local DNS server. The source IP address for these queries was set to the UE IP address using the "-b" option. Subsequently, we made DNS queries to domains that have DNSSEC enabled to ensure the authenticity and integrity of the responses. If the resolver supports DNSSEC validation, it would output the "Authentic Data" (AD) flag in the response's flags section. After considering this aspect for our experiments, we selected the following domains from diverse regions for the experiments: `internetsociety.org`, `dnssec-tools.org`, `dnssec-deployment.org`, `kumari.net`, `huque.com`, `bortzmeyer.org`, `afnic.fr`, and `netfuture.ch`.

We then created four Bash scripts that run different variations of the `kdig` command, binding them to the UE IP address for both scenarios (with and without the RYU script). The main purpose of the scripts is to measure and compare the latency of DNS queries considering the effect of caching. The difference between each script is described in the following:

- The first script uses the "+dnssec" `kdig` option. By including this option, it allows to measure the latency experienced during DNSSEC validation.
- The second script includes the "+tls" option inserted in the `kdig` command. By using TLS, it allows to assess the latency when the DNS queries are sent over a secure TLS connection.
- The third script combines "+dnssec" and "+tls" options in the `kdig` command. Combining both options, it allows for measuring the latency when both DNSSEC validation and TLS encryption are applied to the DNS queries.
- The last script does not utilize any DNS security options.

The code that contains the 5G setup environment including a secure or privacy-preserving DNS is available at GitHub[1].

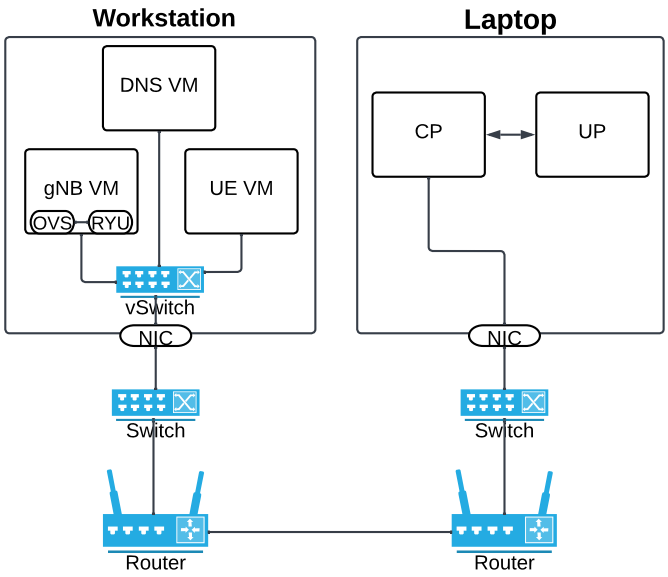

Fig. 8. 5G Network Experiment Environment where the UE and gNB sits on a workstation and the core network is on a separate laptop.

D. Performance Results

In this section, we provide the experiment results based on the established metrics for the DNS queries using several public DNS Servers such as `OpenDNS` While there are many other public servers such as `Google, Cloudflare DNS`, they are dispersed to different geographical locations and the responses may come from different servers each time, and thus it may be difficult to do controlled experiments. Therefore, we chose to go with OpenDNS which provides stable and consistent responses for research experimentation.

The analysis of the results reveals several key findings regarding DNS queries with and without an SDN controller.

[1]https://github.com/adwise-fiu/ADWISE-5G-DNS-Security

TABLE II
DNS PACKET RESPONSE SIZE UNDER DIFFERENT APPROACHES (BYTES)

	Response Size (Bytes)				
	DNS	DNSSEC	Local DNS (DoT)	Local DNS (DNSSEC + DoT)	Public DNS servers (DoT, DNSSEC + DoT)
internetsociety.org	69	195	80	195	468
dnssec-tools.org	98	221	109	221	468
dnssec-deployment.org	71	199	82	199	468
kumari.net	44	161	55	161	468
huque.com	80	386	91	386	468
ortzmeyer.org	48	233	59	233	468
afnic.fr	42	157	53	157	468
netfuture.ch	46	165	57	165	468

As shown in Table I, the average latency for our local DNS server through SDN are significantly lower compared to that of the OpenDNS server (e.g., 5.43 ms vs. 152.4 ms). This translates to a huge saving already (i.e., close to 30 fold) for time-critical applications.

When it comes to security, the local DNS approach with SDN still brings significant latency reductions despite the fact that it increases compared to DNS with no security. For instance, DNSSEC latency in our approach is almost 15 fold reduced. This applies to DoT and DNSSEC+DoT as well, for which the reduction is around 10 fold. All in all, a local DNS makes a big impact on latency and keeps it below 70ms for all examples. Any real-time applications with a few secs latency guarantees may greatly benefit from our approach.

The results are also interesting in terms of security and privacy considerations. It seems that DNSSEC and DoT performance is similar in case of remote DNS but that is not the case for our approach. This can be attributed to the fact that the records are already coming from the cache and thus signature verification times are excluded in our approach, making it behave close to a non-secure DNS. When DoT and DNSSEC are run together, the impact of DNSSEC overhead is very little and thus we can claim that our approach offers security and privacy at the same time without any major increase in the overhead.

The next metric to evaluate is the response packet size in bytes, as shown in Table II. The response packet size is a crucial factor to consider as it directly impacts the efficiency and effectiveness of the DNS infrastructure. For our analysis, we combined the results for DNS and DNSSEC queries, as they produced the same values for both the local DNS and remote public DNS infrastructures. However, different results were obtained for the DoT and DNSSEC + DoT experiments. In terms of the DNS response packet size, both local DNS and public DNS values range from 42 to 98 bytes. These values were relatively smaller than the other results, where we added DNS security. Regarding DNSSEC, the packet size was raised to a range of 157 to 386 bytes. This increment can be ascribed to introducing cryptographic signatures and keys, where packet size varies depending on the domain. When it comes to DoT, the local DNS exhibited smaller packet sizes, spanning from 53 to 91 bytes. In contrast, the DoT results for public DNS servers displayed a fixed packet size of 468 bytes. The fixed packet size for DoT in public DNS servers may be due to optimization and standardized implementation within their infrastructure. These fixed-size results not only apply to DoT but also to the combination of DNSSEC and DoT. In comparison, the local DNS server with SDN showed an incremental packet size, ranging from 157 to 386 bytes, when DNSSEC and DoT were combined. Despite this increment, the packet size remained smaller than that of the other public DNS servers. These results indicate that the local DNS server offers a smaller response packet size than popular public DNS servers, which also helps in reducing the latency.

E. Post Quantum Security Considerations

All the current digital signatures that are in use by Internet standards (e.g., RSA, ECDSA, EDDSA) rely on mathematical problems that are hard to solve using the available classical computers. Nevertheless, quantum computers have the possibility to solve these problems quickly and render these current cryptographical schemes ineffective. In recent years, post-quantum (PQ) signature algorithms are being actively researched and developed as a response to these challenges [16]. They explore mathematical problems and cryptographic techniques that can resist attacks from quantum computers in which we can future-proof our digital signature systems and ensure the user's security and integrity.

In our previous experiments, all communications using DNSSEC were secured based on RSA signatures which are not PQ secure. If we were to replace this algorithm with a PQ alternative, one of the options that can offer similar or better performance when acquiring DNSSEC query is Falcon-512 which was reported to provide even less delay compared to RSA [17] [18].

Thus, we also implemented a PQ DNSSEC in our deployed 5G-edge DNS server using Falcon-512 signature and compared it with different signature algorithms such as RSA, Elliptic Curve Digital Signature Algorithm (ECDSA), and Edwards Curve (ED). For this PQ DNSSEC implementation, we relied on the UPF network function to forward the queries to the DNS Server at the gNB. This approach differs from the previous DNSSEC implementation, where all the DNS queries are handled at the gNB, as shown in Fig. 5 The

reason for this modification was that the available DNSSEC implementation with Falcon-512 was only an emulation of the protocol behavior, impacting the deployment of the prior SDN approach. Without using the SDN approach, the UPF can receive DNS queries coming from the UE and then forward them to the local DNS server at the gNB, which also follows the DNS response. The only additional propagation delay here is the forwarding of packets among gNB and UPF, but the DNS server still sits at the edge. It is important to highlight the benefits of the SDN approach, where it forces the DNS queries to be handled at the edge and might be beneficial in case the RAN and the UPF have a higher distance than our current case. Also, note that we could not test the DNSSEC Falcon version with public DNS servers (such as Google) since there has not been any support for PQ capabilities. To be able to conduct experiments, we created local DNS records and retrieved those records from our local environment.

As shown in Table III, the average latency of DNSSEC with Falcon-512 was the lowest compared to the rest of the signatures with a latency of 3.7 ms. This is interesting because typically PQ algorithms bring more delay and increased packet size. DNS without any security is faster as expected but since the used implementation is an emulation and there is no SDN processing at the edge, the latency values are even less compared to Table I. With these experiments, we have shown that replacing RSA certificates with PQ signature algorithms in our 5G setup environment will not affect DNS latency (indeed Falcon-512 reduces it) and will make DNSSEC even more resistant to quantum attacks.

TABLE III
AVERAGE LATENCY RESULTS FOR DNS RESPONSES FOR DIFFERENT SIGNATURES ALGORITHMS

	Average Latency (ms)	
	Local DNS	
Zones Example	**DNSSEC**	**DNS - No Signatures**
RSA-SHA256	5.3	1.0
RSA-SHA512	5.8	1.3
ECDSA-256	4.2	1.5
ED-25519	6.0	1.4
Falcon-512	3.7	1.1

VI. CONCLUSION

In this paper, we proposed an approach that utilizes SDN capabilities to allow DNS service at the edge for 5G networks. In addition to DNS, we enabled DNSSEC and DoT to enhance the security and privacy of DNS operations for 5G networks. Our implementation focused on integrating these technologies at the base station in a real 5G testbed, utilizing SDN to facilitate DNS services while ensuring enhanced security and privacy. Our study revealed that the local DNS server, deployed at the edge using our proposed approach, performed better than public DNS servers like OpenDNS regarding latency and response packet size. These results imply that bringing the DNS server closer to the edge can significantly reduce the latency associated with DNS resolution, resulting in faster and more efficient network operations. As a result, organizations can guarantee secure and effective DNS operations within their 5G network architecture by utilizing this approach.

ACKNOWLEDGMENT

This research was funded by a National Centers of Academic Excellence in Cybersecurity grant (H98230-21-1-0324 (NCAE-C-002-2021)), which is part of the US National Security Agency and National Science Foundation under the grant #2147196. This research also utilized some funds from the Florida Institute of Technology's Institutional Research Incentive.

REFERENCES

[1] S. K. Rao and R. Prasad, "Impact of 5G technologies on industry 4.0," Wireless personal communications, vol. 100, pp. 145–159, 2018.
[2] Z. Huang, C. Xiong, H. Ni, D. Wang, Y. Tao, and T. Sun, "Standard evolution of 5g-advanced and future mobile network for extended reality and metaverse," IEEE Internet of Things Magazine, vol. 6, no. 1, pp. 20–25, 2023.
[3] M. M. Nesary and A. Aydeger, "vDNS: Securing DNS from amplification attacks," in 2022 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom). IEEE, 2022, pp. 102–106.
[4] R. Harrilal-Parchment, D. Pineda, K. Akkaya, A. Aydeger, and A. Perez-Pons, "Bringing DNS service to 5G edge for reduced latencies in mMTC applications," in 2023 IEEE International Conference on Industrial Technology (ICIT), 2023, pp. 1–6.
[5] L.-C. Kao and W. Liao, "5G intelligent a+: A pioneer multi-access edge computing solution for 5G private networks," IEEE Communications Standards Magazine, vol. 5, no. 1, pp. 78–84, 2021.
[6] S. Nowaczewski and W. Mazurczyk, "Securing future internet and 5G using customer edge switching using DNSCrypt and DNSSEC," J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl., vol. 11, no. 3, pp. 87–106, 2020.
[7] M. Suzuki, T. Miyasaka, D. Purkayastha, Y. Fang, Q. Huang, J. Zhu, B. Burla, X. Tong, D. Druta, J. Shen et al., "Enhanced DNS support towards distributed MEC environment," ETSI White Paper, no. 39, 2020.
[8] L. Shrama, A. Javali, and S. K. Routray, "An overview of high speed streaming in 5G," in 2020 International Conference on Inventive Computation Technologies (ICICT). IEEE, 2020, pp. 557–562.
[9] IANA, "Root servers," Available at https://www.iana.org/domains/root/servers [Accessed: Mar. 07, 2023].
[10] M. Lyu, H. H. Gharakheili, and V. Sivaraman, "A survey on DNS encryption: Current development, malware misuse, and inference techniques," ACM Computing Surveys, vol. 55, no. 8, pp. 1–28, 2022.
[11] T. Jinmei and P. Vixie, "Implementation and evaluation of moderate parallelism in the BIND9 DNS server." in USENIX Annual Technical Conference, General Track, 2006, pp. 115–128.
[12] K. Bumanglag and H. Kettani, "On the impact of dns over https paradigm on cyber systems," in 2020 3rd International Conference on Information and Computer Technologies (ICICT). IEEE, 2020, pp. 494–499.
[13] D. Pineda, R. Harrilal-Parchment, K. Akkaya, A. Ibrahim, and A. Perez-Pons, "Design and analysis of an open-source sdn-based 5G standalone testbed," to appear, proceedings of INFOCOM Workshop on Computer and Networking Experimental Research using Testbeds (CNERT 2023).
[14] "Open vswitch," 2016. [Online]. Available: http://www.openvswitch.org/
[15] "Ryu SDN framework," 2017. [Online]. Available: https://ryu-sdn.org/index.html
[16] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, C. Miller, D. Moody, R. Peralta et al., "Status report on the third round of the NIST post-quantum cryptography standardization process," US Department of Commerce, NIST, 2022.
[17] J. Bozhko, Y. Hanna, R. Harrilal-Parchment, S. Tonyali, and K. Akkaya, "Performance evaluation of quantum-resistant TLS for consumer IoT devices," in 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC). IEEE, 2023, pp. 230–235.
[18] "Post-quantum DNSSEC," 2022. [Online]. Available: https://github.com/nils-wisiol/dns-falcon