Performance Evaluation of Quantum-Resistant IKEv2 Protocol for Satellite Networking Environments

Ahmet Mutlugun*,Yacoub Hanna†, Kemal Akkaya†
*Department of Computer Science, San Jose State University, San Jose, USA
Email: ahmet.mutlugun@sjsu.edu
†Advanced Wireless and Security (ADWISE) Lab, Florida International University, Miami, USA
Email: {yhann002, kakkaya}@fiu.edu

***Abstract*—With the introduction of post-quantum cryptography (PQC) algorithms, there are ongoing efforts to not only standardize the proposed solutions but also consider integrating them into existing network applications and evaluating their performance. This evaluation is especially critical for security-sensitive applications such as satellite communications, where network limitations such as high packet loss and propagation delay pose unique challenges. In this paper, we consider the integration of PQC with one of the widely used security protocols, namely IPSec, by focusing on its key exchange protocol IKEv2. Specifically, we evaluate how PQ key exchange and digital signatures impact the latency compared to existing classical crypto solutions. We demonstrate through simulation how such integration triggers fragmentation that needs to be handled by IKEv2 and quantify the performance overhead due to packet loss or delaying of such fragments when used under satellite networking applications. The results indicate that while higher packet losses pose significant overheads that may hinder the adoption of PQ-based IPSec solutions, this may be subsidized when propagation delays are much higher in satellite networks.**

***Index Terms*—IKEv2; IPsec; VPN; Satellite communication; Post-Quantum Cryptography; Key Encapsulation Mechanism**

I. Introduction

As quantum computing technology develops, current encryption standards may not hold up to the unconventional architecture of quantum computers. When sufficiently powerful quantum computers become available, they may be able to use Shor's algorithm [1] to compute prime factors of large integers. This capability poses a significant threat to public key cryptography, as their security relies on the difficulty of factoring large integers. Most notably, RSA encryption and the Elliptic Curve Diffie-Hellman (ECDH) may be affected, as both are widely used and are vulnerable to attacks using Shor's algorithm. As a safety measure, agencies like the National Institute of Standards and Technology (NIST) are taking steps to develop quantum-resistant algorithms, which we refer to as post-quantum cryptography (PQC). NIST has standardized Kyber, Dilithium, and Falcon to be used in security-critical applications [2].

While quantum computers are not powerful enough to crack current encryption standards, there is an urgency to secure sensitive data from future attacks by implementing PQC. This need is particularly critical in satellite communications, where the sensitivity of the transmitted data makes them especially vulnerable to future decryption attempts [3]. Satellites present additional challenges due to their extended operational lifespans, often exceeding a decade. The longevity of satellites, combined with the limitations in their ability to receive significant software or hardware updates once deployed, creates a critical vulnerability to future cryptographic threats.

This scenario underscores the imperative of implementing PQ measures before launch. Retrofitting satellites with new cryptographic protocols can be technically challenging or even infeasible, potentially exposing valuable data to future decryption attempts throughout the satellite's entire service life. While PQC algorithms offer enhanced security against future quantum attacks, their implementation presents challenges in satellites. The physical distance of the satellites results in high propagation delays and packet loss. With PQ algorithms producing larger public and private keys, Internet protocols have to be suitable to transfer more data, which may be impacted due to constrained network conditions.

One of the protocols that can create a reliable and authenticated communication channel among devices across the Internet, particularly within the frameworks of Virtual Private Networks (VPNs) and Internet Protocol Security (IPSec) implementations, is Internet Key Exchange (IKEv2). It enables security features like confidentiality, integrity, and secure data exchange for IPSec-based VPNs. The message exchanges in IKEv2 are designed to offer key agreement and mutual authentication with four message exchanges. However, fragmentation issues arise under certain network conditions, particularly in high-latency or constrained networks. Fragmentation occurs when the message size exceeds the underlying network's Maximum Transmission Unit (MTU). This is due to the increased size of signatures and the use of larger public keys when PQ algorithms are integrated into IKEv2. This can cause delays, packet loss, or even complete failure of key exchanges.

Therefore, in this paper, we evaluate the performance of PQ

key exchange (i.e., key encapsulation mechanism (KEM) and authentication within IKEv2 and analyze the classical solution's performance with respect to the quantum-resistant solutions under a simulated satellite communication environment. Specifically, we used Kyber as a PQ KEM while deploying Falcon and Dilithium digital signature in the authentication phase. The results indicate that PQ-based IKEv2 implementations face significant performance challenges, particularly in high packet loss scenarios. With higher packet losses, the gap between classical and PQ setups increased. However, we also observed that higher propagation delays in satellite applications can subsidize this gap, making Falcon and Dilithium comparable in some cases.

The rest of the paper is organized as follows: First, we present state-of-the-art works on IKEv2 integration with PQC in Section II. In Section III, we present the background on IPsec, IKEv2, and PQ algorithms. We present our objectives and motivations in Section IV. In section V, we presents details of the implementation that we used to perform the experiments. In Section VI, we evaluate the performance of our implementation in various settings and discuss the results. Finally, we summarize the conclusions in Section VII.

II. Related Work

IKEv2 performance/overhead has been studied in various context. For instance, Lee and Kim [4] presented the performance of IKEv2 with classical encryption in mobile IPv6 networks. They analyzed the authentication key resetting and re-keying of the IKEv2 protocol focusing on the effects of limited bandwidth on key exchange. Their experiments showed that initializing time in IKEv2 may be challenging in existing wireless infrastructure due to limited bandwidth.

However, IKEv2 integration with PQC has just started to receive attention. For instance, in [5], the authors comprehensively examine the challenges and proposals for integrating PQC into network protocols such as IPSec and IKEv2. The paper reviews several PQ key agreement proposals, including efforts by the Open Quantum Safe (OQS) project and integrating Liboqs into popular cryptographic libraries like OpenSSL. However, the paper did not provide any actual evaluations. Similarly, Bae et al. [6] evaluated the performance of key exchange in IPSec in terms of latencies and packet sizes. Their work provides valuable insights into computational performance in IPsec but does not consider any constrained network conditions in terms of packet loss and long propagation delays. The work also does not integrate PQ signature schemes.

PQ overhead has also been studied under TLS protocol, particularly for IoT settings. The authors measured the latency of TLS handshakes over IP over Bluetooth connections [7] and 5G authentication scenarios [8]. However, unlike TLS, which operates over TCP, IPsec utilizes UDP, making it more susceptible to packet loss and potentially affecting the performance of PQ algorithms differently.

Our work considers a comprehensive PQ integration for IPSec with key exchange and signatures and offers a thorough analysis under satellite networking conditions with packet loss and longer propagation delays.

III. Background

IPsec is commonly used as a VPN protocol to enforce secure communication between two parties. IPSec comprises three sub-protocols: Internet Key Exchange Version 2 (IKEv2), Authentication Header, and Encapsulating Security Payload. IKEv2 uses public key encryption to set up a secure channel. This is achieved by exchanging the required signatures, certificates, and cryptographic keys between two parties to establish a shared secret key.

A. IKEv2

Internet Key Exchange (IKE) is the protocol used within IPSec to set up secure communication. It is responsible for negotiating and establishing Security Associations (SAs) between communicating parties by agreeing on the cryptographic suite, establishing shared secret keys, and authenticating them to each other. IKEv2 is the latest protocol version, offering enhanced performance, reliability, and security. IKEv2 typically involves exchanging a series of messages between two parties, known as the initiator and the responder. The first message exchange is IKE_SA_INIT, where the two parties agree on the encryption and authentication standards. The messages also contain both the initiator and the responder's nonces to prevent replay attacks. With traditional authentication methods, both parties share their Diffie-Hellman public keys in this step to compute a shared secret key for further communication. If no additional key exchange methods are used, a final IKE_AUTH message is exchanged to authenticate both parties. Certificates and signatures are used to transmit identities and prove knowledge of secrets. In total, two initiator messages and two response messages are sent as seen in Fig. 1.

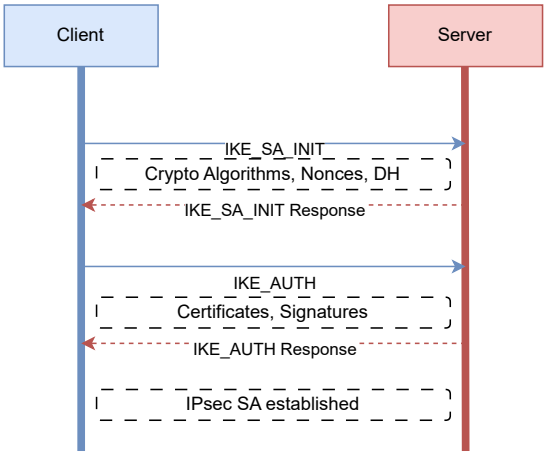

Fig. 1. IKEv2 Initiation Message Exchanges

B. Post-Quantum Cryptography

The emergence of quantum computing has revealed significant vulnerabilities inherent in conventional encryption algorithms. For instance, Shor's algorithm can effectively break down the widely employed discrete logarithm and integer

factorization challenges. Consequently, post-quantum cryptography offers methodologies for formulating public-key cryptosystems that demonstrate resilience against quantum computational threats [9]. In this study, we focus on the two signature scheme families and one KEM:

Signature Schemes: PQ signature schemes aim to provide alternatives resistant to quantum attacks. Therefore, we will be evaluating *Falcon* and *Dilithium*, which have been finalized by the United States National Institute of Standards and Technology (NIST) as the initial post-quantum signature algorithms to be standardized [10]. Dilithium produces public keys that take a minimum of 1,312 bytes at the lowest level, with higher levels taking up to 2592 bytes for the public key. Compared to RSA's 512-byte public keys, Dilithium keys are much larger. Additionally, Dilithium signatures are even larger: a Dilithium5 signature is 4,595 bytes.

KEM: KEM is used for secure key exchange. This mechanism allows two parties to agree on a shared secret key by encapsulating it through asymmetric key cryptography. This process is fundamental for secure communications. Therefore, *CRYSTALS-Kyber* standardized by NIST will be assessed in our performance evaluation [11]. It has three variants with public key sizes changing from 800 to 1568 bytes [12].

IV. Objectives and Research Questions

A. Problem Context and Motivation

Since IKE uses the UDP protocol, without IKE's own fragmentation, large packet transmissions would have to be fragmented at the IP level. Despite going against the current Internet standards, many NATs still do not support IP-level fragmentation due to legacy hardware and security concerns. To address these concerns, IKEv2 had to support fragmentation at the protocol level rather than relying on IP fragmentation. Standardized by RFC 7383, IKEv2 could handle its fragmentation by breaking up messages greater than a configurable size [13].

However, this fragmentation solution was geared for the authentication phase only (i.e., IKE_AUTH messages) since fragmentation can happen once the keys (for encryption and authentication) have been established after the IKE_SA_INIT phase. The problem was that the RFC did not address any potential fragmentation during the IKE_SA_INIT phase that can now happen due to the use of KEMs for supporting PQC where the public key sizes might be higher than that of Maximum Transfer Units (MTUs). Consequently, another RFC (RFC9242) recently came to fill this gap, introducing the IKE_INTERMEDIATE message exchange. With this mechanism, after the initial key exchange (i.e., using classical (EC)DH), one or more IKE_INTERMEDIATE exchanges can be done to accommodate KEMs. As the IKE_INTERMEDIATE exchange is encrypted, the IKE fragmentation protocol (RFC7383) can also be used here, similar to IKE_AUTH. However, with the addition of the IKE_Intermediate step, a total of six messages are sent between the initiator and the responder, which may bring a major overhead, as discussed next.

B. PQC and IPSec Integration

The standard key encapsulation method is CRYSTALS-Kyber, as selected by the NIST competition. Kyber is not only computationally slower than classic DH key exchanges like Curve25519 [14], but it transmits much larger public keys as mentioned before. The increased key size warrants an additional IKE_INTERMEDIATE, with higher Kyber levels requiring fragmentation by exceeding the maximum fragment size of 1280 bytes by default. In addition to Kyber, CRYSTALS-Dilithium and Falcon are used as a post-quantum signature scheme. These PQ signatures also introduce significant size increases compared to classical signatures. For instance, Dilithium signatures range from 2420 to 4595 bytes, while Falcon signatures range from 666 to 1280 bytes as shown in Table II. These larger signature sizes, especially Dilithium, often exceed the default 1280 byte fragment size, necessitating fragmentation during the IKE_AUTH phase.

Such fragmentation due to PQC in IKEv2 presents challenges in constrained networks characterized by noisy wireless channels and limited bandwidth or resources. This is because such limitations result in packet losses, which trigger retransmissions in IKEv2, with fragmented messages requiring all fragments to be resent if any are lost. An example case is shown in Fig. 2.

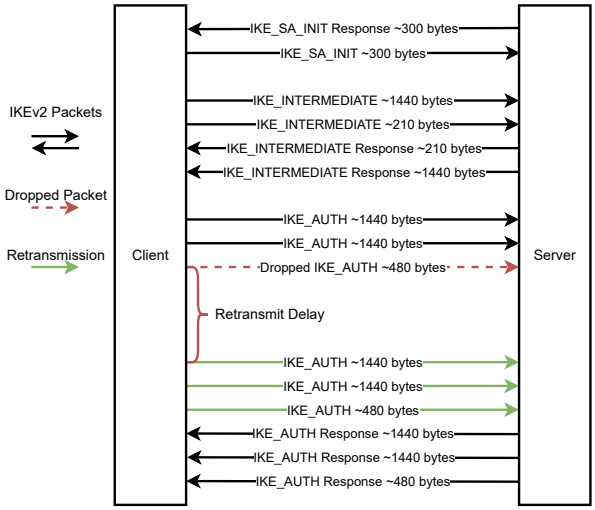

Fig. 2. IKEv2 Initiation With a Re-transmission

As the heavier PQ algorithms increase fragmentation, the probability of packet loss and re-transmissions grows, potentially causing significant delays in establishing IKE SAs. This paper aims to quantify these effects, providing insights for implementing PQ-secure communication under challenging network conditions. Specifically, we strive to answer the following questions:

- How do PQ signature schemes impact the total number of fragmentations in IKEv2 messages?

- How do KEMs affect the number of IKE_INTERMEDIATE packets required?
- What is the impact of fragment losses experienced by IKE_INTERMEDIATE and IKE_AUTH on IKEv2 performance under both classical and PQ-based versions?
- What is the impact of increased propagation delay on the performance of IKEv2 under both classical and PQ-based versions?
- What are the practical implications of performance differences for implementing PQ security under satellite networks?

C. Use-cases

To this end, we focus on satellite communication use cases where not only packet losses are possible, but also propagation delays are much higher than in terrestrial networks. For instance, a use-case might be a control user which directly communicates with a satellite for configuration and management. Another use case might even introduce further propagation delays. In such cases, two users can communicate directly through satellites; thus, their messages will travel through multiple satellites until they reach the destination. Fig. 3 depicts these cases. If these applications are sensitive, such as relating to critical infrastructure, military, or emergency response, their security will be of utmost importance.

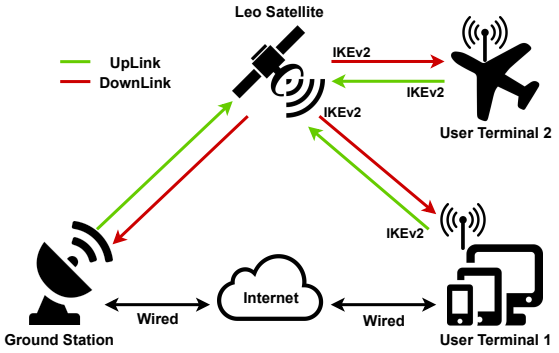

Fig. 3. Satellite Communication Use-cases for deploying IPSec and IKEv2

V. Experimental Setup and Implementation

A. Implementation Environment

We first identified an IPSec VPN implementation library as IKEv2 is part of such an environment. StrongSwan[1] was selected as the preferred implementation library of IPSec due to its open-source status and support for PQC through integration with the Open Quantum Safe (OQS) library.

All implementations and tests were conducted on ARM virtual machines using VMware Fusion. The virtual machines were configured with 2 CPU cores and 2 gigabytes of RAM each, running on Ubuntu 22.04 LTS.

Using a setup script modified from the official StrongSwan post-quantum Docker setup[2], the StrongSwan application was installed, and the OQS library plugins were enabled. The fragmentation size was set to 1480 bytes, as setting it any higher would exceed Ethernet's 1500-byte MTU. The re-transmission settings were kept at default, with the re-transmission delay at four seconds.

Both virtual machines were loaded with bash scripts to update the signature scheme. The script generates three sets of keys and certificates: one for the root certificate authority, one for the initiator, and one for the responder. Both virtual machines are then loaded with their certificates alongside the certificates for the certificate authority.

In the experiments, the initiator (client) opens a connection to the host (responder) and measures the IKEv2 setup time elapsed over 500 asynchronous connections for statistical significance. [3]

B. Simulating Satellite Networks

The two virtual machines are connected with a bridged VMware network connection with no network restrictions. Using the VMware Fusion interface, the network connections can be modified to change network conditions so that they can initiate Satellite communication channels. VMware's VM allows restricting the bandwidth, packet loss and the propagation delay for both incoming and outgoing connections. For our propagation delay, we picked three different values to simulate various satellite communication scenarios:

- 0ms: Baseline measurement with no added delay
- 100ms: One-way delay for MEO satellites or two-way delay for LEO satellites [15]
- 200ms: Two-way delay for MEO satellites

These delay values were chosen to imitate realistic communication scenarios through LEO and MEO satellites. The 100ms and 200ms delays represent typical round-trip times for LEO and MEO satellites, respectively [16], allowing us to evaluate PQ-IKEv2 performance in practical satellite environments. Similarly, several packet loss percentages were selected. 0% was used as a baseline for comparison. 1% and 2% were chosen as they represent typical packet loss rates expected in modern LEO and MEO satellites. Lastly, 5% was included to simulate more challenging or degraded network conditions where higher packet loss may occur.

C. Metrics and Baselines

We used *average runtime* as our performance metric which is defined as the time elapsed from the client sending the IKE_SA_INIT message to the completion of the IKE SA. We recorded the timestamp before and after creating the IKE SA. Therefore, our runtimes reflect the total time elapsed on the client side, which includes the network transmission time and the computational overhead.

The performance of IKEv2 is tested with three setups:

- Key Exchange Tests: We assessed various key exchange protocols including KEM.

[1]https://www.strongswan.org/

[2]https://github.com/strongX509/docker/tree/master/pq-strongswan

[3]https://github.com/adwise-fiu/PQ-IPsec

- Certificate Tests: We evaluated different certificate types while maintaining Curve25519 (x25519) as the consistent key exchange method.
- Classical vs Post-Quantum Comparison: We compared a classical cryptographic setup (RSA + x25519) against two post-quantum configurations (Kyber5 + Falcon1024 and Kyber5 + Dilithium5).

VI. Performance Results and Analysis

To provide an accurate analysis of signature schemes, we recorded their fragmentation characteristics in our setup. Table I shows the number of IKE_AUTH fragments produced by the client and the server.

TABLE I
Fragment Count Comparison for Signatures and KEMs

Algorithm	Label in Figs	Fragments Sent	Fragments Received
ED25519	ed25	1	1
ECDSA	ecd	1	1
RSA-2048	RSA	2	1
Dilithium2	di2	5	5
Dilithium3	di3	7	7
Dilithium5	di5	10	9
Falcon512	fal512	2	2
Falcon1024	fal1024	4	4
ECDH	x25519	-	-
Kyber512	ky1	1	1
Kyber768	ky3	1	1
Kyber1024	ky5	2	2

A. 0ms Propagation Delay

Impact of KEMs: As seen in Fig. 4, all KEMs are impacted significantly when packet loss is especially above 2%. The situation gets worse for PQ solutions beyond 1% compared to classical DH. The gap becomes more than double reaching its peak at 2% while dropping partly at 5%. We also observed that Kyber512 performed closest to Kyber768 since both introduced two IKE_INTERMEDIATE packets in total. Conversely, with Kyber1024, the intermediate packet was fragmented into two parts, increasing the chance of re-transmission. That is most likely the main contributor to the increased runtime of Kyber1024. The main lesson here is that IKEv2 would not be a viable solution if the packet loss starts to exceed 2%.

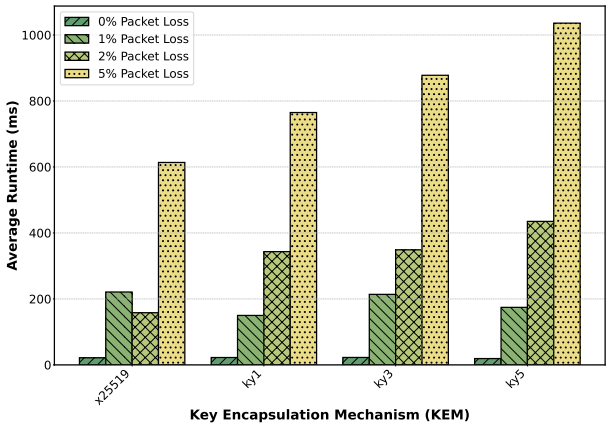

Fig. 4. Avg. Run-times of KEMs With 0ms Propagation Delay

Impact of Signatures: In this test, Curve25519 was used as the DH key exchange. We then proceeded to authentication process. We again observed that quantum-resistant authentication was heavily impacted by packet loss conditions as seen in Fig. 5. The good news is that Falcon512 performed close to RSA. even at high packet losses, such as 2% and 5%, Falcon was 6.7% and 9.3% slower than RSA respectively due to increased number of IKE_AUTH fragments. On the contrary, Falcon1024 was 76.0% slower than Falcon512, and 92.3% slower than RSA. Different variants of Dilithium performed much worse which again made it impractical to be deployed beyond 2% packet loss. These results suggest a similar outcome of KEM experiment except the fact that Falcon512 is comparable to classical solutions.

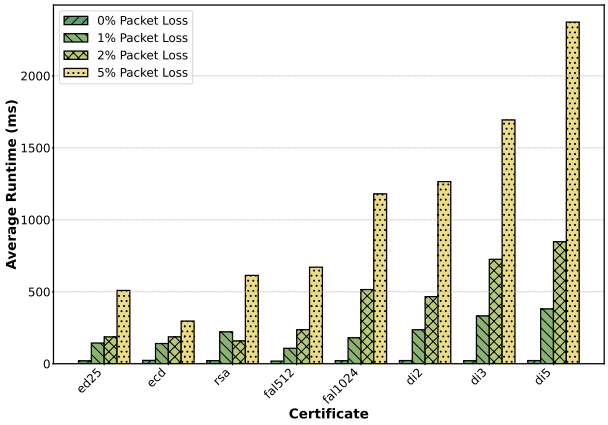

Fig. 5. Avg. Run-times of Signatures With 0ms Propagation Delay

Impact of a full PQ-based System: The final test with 0ms propagation delay compares a fully PQ system (i.e., both KEM and authentication) to a classical instance of IPSec. The classical setup used Curve25519 + RSA and the PQ setups used Kyber1024 + Falcon1024/Dilithium5 as KEM and signatures, respectively. At 1% packet loss, the Falcon setup was 65.3% slower than the classical setup, while the Dilithium setup was 83.2% slower as seen in Fig. 6. The gap between the classical and PQ setups widened at higher packet losses, such as 2% and 5%. At 5%, the Falcon setup was 109.7% slower, while the Dilithium setup was 320.5% slower than RSA. All these results suggest that once the packet loss exceeds 1%, PQ solutions may not be able to offer a good quality of service.

B. 100ms Propagation Delay

We now consider 100ms propagation delay for LEO satellites applications to quantify its impact on the performance compared to 0ms propagation delay.

Impact of KEMs: We observed that the impact on PQ KEMs were lower than that of the prior results as shown in Fig. 7. Once again, there was a trend of slower runtimes with higher Kyber levels. At 5% packet loss, Kyber512 was 24.5% slower than x25519, while Kyber768 was 41.4% slower. Kyber1024 was the slowest, since it's IKE_INTERMEDIATE packet is fragmented in our setup. At 5% packet loss, it was 77.2% slower than x25519.

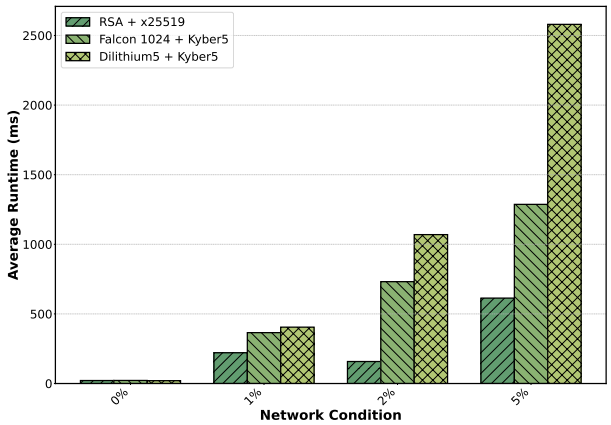

Fig. 6. Classical vs PQ IKEv2 Average Run-times at 0ms Propagation Delay

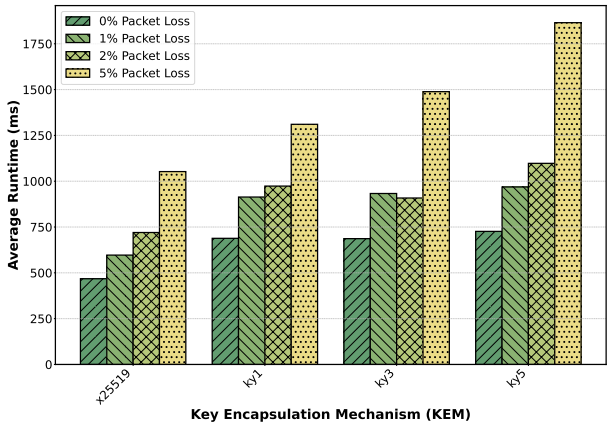

Fig. 7. Avg. Run-times of KEMs With 100ms Propagation Delay

Impact of Signatures: As seen in Fig. 8, Falcon512 demonstrated the closest performance to classical schemes, with only a slight increase in average runtime as packet loss increased. Falcon1024 showed a more noticeable performance drop, particularly at 5% packet loss. The Dilithium variants had the most significant performance deterioration under packet loss conditions. Specifically, Dilithium5 had the most increase in average runtime, from about 470ms at 0% packet loss to 3138.5ms at 5% packet loss, a 567.5% slowdown. Considering RSA's 125% slowdown in the same comparison, we can conclude that PQ signature schemes, particularly those with larger signatures like Dilithium, are significantly more sensitive to high packet loss.

Impact on a full PQ-based System: Under a full PQ setup, we observed that increasing propagation delays bring some advantage compared to prior 0ms propagation delay. At 0% packet loss, the Falcon and Dilithium setups showed nearly identical performance, with average runtimes differing by less than 1% while almost matching the performance of RSA + ECDH. The high propagation delay masked the differences in computational and network latency. However, at higher packet losses, PQ setups were consistently slower than the classical setup due to the additional IKE_INTERMEDIATE step required for Kyber1024. As packet loss increased, the Dilithium setup falls behind the Falcon setup, slowing down by

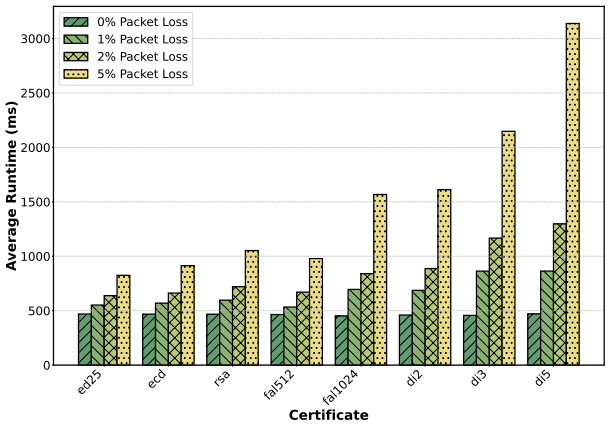

Fig. 8. Avg. Run-times of Signatures With 100ms Propagation Delay

31.2% at 1% and 43.1% at 2% packet loss. At 5% packet loss, Dilithium setup is nearly 51% slower than the Falcon setup. The main observation here is that the gap between 1% and above is not widening dramatically which was the case in 0ms setup. However, we do not see any advantage for 1% packet loss either. Any packet loss above 0% will come at additional overheads when PQ is used in IKEv2 satellite setups.

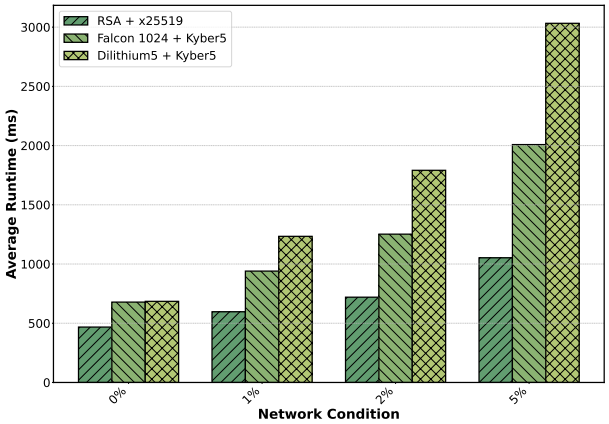

Fig. 9. Classical vs PQ IKEv2 Average Run-times at 100ms

C. 200ms Propagation Delay

With 200ms propagation delay, all KEMs and signatures had increased runtimes compared to lower propagation delays as seen in 10. Due to space constraints, we do not show the results for KEM and signatures and offer separate discussions as the trends are similar to prior cases when comparing PQ and classical approaches. However, it is worth noting that while packet loss still impacts performance, the higher propagation delay dominates the overall runtime, making the relative impact of packet loss less significant compared to lower latency scenarios. In other words, the gap for Dilithium gets smaller when compared to RSA and Falcon. This suggests that increased packet loss may not have a devastating impact on PQ solutions when propagation delays are increasing.

D. Further Analysis and Recommendations

Based on these findings, we recommend Falcon as the signature scheme for scenarios with high propagation delay

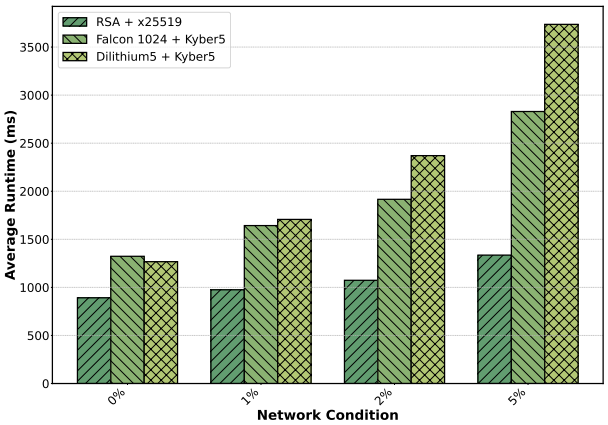

Fig. 10. Classical vs PQ IKEv2 Average Run-times at 200ms

and high packet loss. Dilithium's large transfer sizes were heavily impacted by packet loss, so it should be reserved for security-critical scenarios. For a PQ key exchange, we recommend Kyber768 for performance-critical applications, since its performance was comparable to the less secure Kyber512, even with constrained network environments.

We also provide an interesting observation in Table II which shows the ratio between PQ solutions, Falcon (F) and Dilithium (D) and classical solution, RSA (R). As can be seen, the ratios increase slightly for Falcon with 200ms but surprisingly it decreases for Dilithium. In other words, Dilithium closes the performance gap as the propagation delay is increasing, which is consistent with our observation under 200ms results. For instance, for 200ms under 1% packet loss, Falcon and Dilithium perform very closely. Therefore, it can be an alternative to Falcon if packet losses are guaranteed to be less than 1% in a satellite network environment. We speculate that with even increased propagation delays, Dilithium can even match the performance of Falcon for packet losses greater than 2%.

TABLE II
PERFORMANCE RATIOS OF PQ VS CLASSICAL IPSEC SETUPS UNDER VARYING PACKET LOSS

Packet Loss	0ms Delay		100ms Delay		200ms Delay	
	F / R	D / R	F / R	D / R	F / R	D / R
0%	1.02	0.94	1.45	1.46	1.48	1.42
1%	1.65	1.83	1.58	2.08	1.69	1.75
2%	4.63	6.77	1.74	2.49	1.78	2.21
5%	2.10	4.20	1.91	2.88	2.12	2.80

VII. CONCLUSION

In this paper, we evaluated the performance of PQ algorithms when integrated to IKEv2, an important part of IPSec protocol under various network conditions. We considered use cases of satellite communications where the chances of packet loss are substantial and the propagation delays greatly exceed those noted in terrestrial networks.

Our findings reveal that while Kyber performed comparably to classical key exchange methods in some scenarios, it fell short with high packet losses and propagation delays. Falcon demonstrated performance closest to classical RSA for PQ signature schemes, while Dilithium experienced significant slowdowns with increased packet losses. As network conditions worsened, the performance gap between classical and PQ setups widened, particularly for PQ signature schemes. However, we also found out that increased propagation delays subsidizes this gap and makes Dilithium comparable to Falcon under certain packet loss percentages.

ACKNOWLEDGMENT

This work was supported in part by the US National Science Foundation under the Award No. 2150248 and 2147196.

REFERENCES

[1] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM review, vol. 41, no. 2, pp. 303–332, 1999.

[2] "Nist releases first 3 finalized post-quantum encryption standards — nist," NIST, 08 2024. [Online]. Available: https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards

[3] O. Kodheli, E. Lagunas, N. Maturo, S. K. Sharma, B. Shankar, J. F. M. Montoya, J. C. M. Duncan, D. Spano, S. Chatzinotas, S. Kisseleff et al., "Satellite communications in the new space era: A survey and future challenges," IEEE Communications Surveys & Tutorials, vol. 23, no. 1, pp. 70–109, 2020.

[4] D. H. Lee and J. G. Kim, "Ikev2 authentication exchange model and performance analysis in mobile ipv6 networks," Personal and Ubiquitous Computing, vol. 18, no. 3, pp. 493–501, Mar 2014. [Online]. Available: https://doi.org/10.1007/s00779-013-0669-8

[5] A. Pazienza, E. Lella, P. Noviello, and F. Vitulano, "Analysis of network-level key exchange protocols in the post-quantum era," in 2022 IEEE 15th Workshop on Low Temperature Electronics (WOLTE). IEEE, 2022, pp. 1–4.

[6] S. Bae, Y. Chang, H. Park, M. Kim, and Y. Shin, "A performance evaluation of ipsec with post-quantum cryptography," in Information Security and Cryptology – ICISC 2022. Springer Nature Switzerland, 2023, pp. 249–266.

[7] 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), 2023, pp. 230–235.

[8] Y. Hanna, D. Pineda, M. Veksler, M. Paudel, K. Akkaya, M. Anastasova, and R. Azarderakhsh, "Integrating post-quantum tls into the control plane of 5g networks," in 2024 IEEE 43rd International Performance Computing and Communications Conference (IPCCC). IEEE, 2024, p. to appear.

[9] D. Stebila and M. Mosca, "Post-quantum key exchange for the internet and the open quantum safe project," in International Conference on Selected Areas in Cryptography. Springer, 2016, pp. 14–37.

[10] National Institute of Standards & Technology, "Nist announces first four quantum-resistant cryptographic algorithms," Jul. 5, 2022. [Online]. Available: https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms

[11] D. Rojas Rodríguez, "Kyber: analysis of a nist´ s post-quantum cryptographic standard," 2024.

[12] P. Schwabe, "Kyber," Nov 2020. [Online]. Available: https://pq-crystals.org/kyber/

[13] V. Smyslov, "Internet key exchange protocol version 2 (ikev2) message fragmentation," Internet Requests for Comments, RFC Editor, RFC 7383, 11 2014. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7383

[14] I. Duits, "The post-quantum signal protocol : Secure chat in a quantum world," February 2019. [Online]. Available: http://essay.utwente.nl/77239/

[15] Telarus, "Leo, meo, and geo, oh my! modern satellite connectivity, explained," May 2024. [Online]. Available: https://www.telarus.com/blog/modern-satellite-connectivity-explained/

[16] M. Kang, S. Park, and Y. Lee, "A survey on satellite communication system security," Sensors, vol. 24, no. 9, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/9/2897