ViNET: Connecting the Unconnected using Video over LTE

Manav Mittal, Yogesh Kaushik, Anirudh S Kumar, Mukulika Maity, Sambuddho Chakravarty
Indraprastha Institute of Information Technology Delhi, Delhi, India
{*manav21538, yogesh20163, anirudh21517, mukulika, sambuddho*}*@iiitd.ac.in*

***Abstract*—Internet shutdowns are used authoritarian regimes to suppress communication that end up crippling essential Internet-driven services, besides the obvious silencing of dissent. Traditional tools like VPNs and Tor, dependent on active Internet connections, falter during these blackouts. Earlier solutions, such as Dolphin, delivered meagre bandwidth and weak privacy safeguards, exposing a glaring weakness in the battle against digital oppression.**

ViNET, a system that cleverly repurposes Video over LTE (ViLTE) calls, often operational during shutdowns, into a stealthy conduit for real-time Internet access. By ingeniously embedding network traffic in ViLTE packets, ViNET achieves robust 60 to 400 Kbps transmission rates, matching 2G speeds and surpassing previous solutions like Dolphin by 1500x–4000x, while ensuring end-to-end TLS-based confidentiality and integrity. This performance enables text-based web browsing with page loads in seconds to minutes, 1 MByte file downloads in ≈30s, and seamless messaging over Telegram.

ViNET also outsmarts machine learning-based traffic classifiers, achieving a remarkable false positive rate, at times as high as 40%, when attempting to detect ViNET using SOTA models. With such standout metrics, ViNET emerges as a formidable ally, offering a performant, reliable and privacy-first lifeline, in the face of Internet shutdowns.

***Index Terms*—Internet Shutdown, Video over LTE, Censorship, Mobile Network Operator (MNO).**

1. Introduction

Internet access, now as critical as electricity and clean water, is a UN-declared human right [1]–[3], essential for democracy, communication, education, business, and global discourse [4], [5]. Yet, authoritarian regimes relentlessly impose shutdowns [6], [7], a brutal censorship tool, severing online services in targeted regions.

The motives behind these shutdowns vary. Some governments weaponize them as a means of political control, silencing dissent and disrupting opposition movements, particularly during protests [8], elections, or civil unrest [9]. Others justify them under national security pretexts, arguing that it prevents coordination among extremist groups or criminals [10], [11]. In certain instances, shutdowns are imposed to achieve mundane goals such as preventing cheating during nationwide examinations [12]–[14]. Regardless of the justification, their consequences are severe. Businesses and services (both essential and non-essential) that rely on Internet connectivity suffer substantial financial losses (estimated at $7.69 billion in 2024 alone [15]). Impact on students' education is severe [16], and access to healthcare and essential services are curtailed [17].

Traditional censorship involved restricting access to a particular resource on the Internet, which can be circumvented using tools like Tor [18], VPNs, *domain-fronting* [19], [20], *decoy-routing* [21]–[24], *etc.* But in these extreme measures these tools are rendered ineffective to ameliorate such situations, as they require an active Internet connection in the first place. There are alternative approaches, like *Briar* [25] which rely on establishing peer-to-peer meshes for relaying messages through Bluetooth. However, they work well only in localized networks and lack scalability needed for widespread Internet access. Others like *Starlink* [26], require expensive hardware deployments, besides being commercial ventures that could be controlled by governments.

Given the limitations of these existing circumvention tools, researchers have turned to a more unconventional approach: re-purposing the cellular network itself as a covert data transmission channel. Cellular networks, particularly voice and video calling services, often remain functional even when data services are disabled. This insight led to the development of *Dolphin* [27], a system that exploited the cellular voice channel to transmit Internet data by encoding it into audio signals. Dolphin allowed users in a shutdown region to send requests to a peer outside the affected area, who would then fetch the requested content and relay it back using an encoded voice call. While innovative, Dolphin suffered from several limitations that restricted its practicality.

A significant drawback of Dolphin was its bandwidth constraints. Dolphin transmitted data through modulated audio signals, and the transmission rate (goodput) was limited to approximately 64 bps, making it impractical for real-time applications. Additionally, the system primarily supported one-way communication, enabling users to send emails or tweets with non-realtime latencies, further limiting its utility.

Secondly, Dolphin does not support end-to-end encrypted protocols. Users connect and communicate with a peer, that in-turn fetches the data on the former's behalf. The reliance on a potentially trusted *yet* curious peer risks compromising the confidentiality and integrity of messages for users in shutdown regions.

Thus we propose *ViNET*, an alternative designed to overcome the shortcomings of Dolphin by leveraging a different communication channel that typically remains operational during shutdowns—Video over LTE (ViLTE). ViLTE is the video extension of Voice over LTE (VoLTE), which operates over a packet-switched IP architecture, offering significantly higher bandwidth, lower latency, and better QoS compared to the traditional circuit-switched

telephone systems. Usually during shutdowns, ViLTE channels are not blocked [28]. Most mobile network operators (MNOs) have now transitioned to using LTE based (4G/5G) cellular services [29]. Therefore, turning off LTE would lead to the disconnection of voice calls, and by extension ViLTE, services as well.

Our architecture enables a robust data tunnel between two parties: one confined to an Internet-shutdown zone, the other operating in a region with full Internet access. A ViLTE call is strategically repurposed to channel network traffic from the caller, located in the shutdown region, to the callee, positioned outside. The caller's device (*i.e.*, laptop or desktop), connects to their phone's WiFi hotspot. Network traffic from the client device reaches the phone, where it is encrypted and expertly embedded into outbound ViLTE packets using the `Netfilter` framework on Android (Linux). This process is executed with precision, ensuring the packets' structural and temporal integrity are preserved, for evading detection through pattern analysis.

At the receiver, *i.e.*, the peer outside the shutdown region, the original packets are decapsulated and decrypted from the ViLTE flows, directing them to their intended destination(s). Responses are re-encrypted, embedded into the same ViLTE call, and transmitted back to the client.

This architecture offers distinct advantages over previous designs. Firstly, it establishes a reliable tunnelling framework, supporting seamless encryption, embedding and transmission of complete IP packets between the client and the destination. This facilitates end-to-end TLS-encrypted HTTPS connections between the client and the intended destinations, safeguarding the privacy, confidentiality, and message anonymity against inquisitive peers outside the shutdown region.

Secondly, the ViNET flows can harness the entire ViLTE channel capacity ($\approx$ 60-400 Kbps), which is roughly the speeds accorded to 2G cellular data channels [30], [31]. This is sufficient for browsing websites using text-mode browsers like `lynx` (with webpages loading in 5-70 seconds (depending upon the page size and location)), downloading 1Mbyte files in roughly 30 sec., and comfortably chatting over instant messaging applications like Telegram.

We tested ViNET extensively in the three largest MNOs in India, the country with the highest number of shutdowns [32]. We simulated shutdown by disabling the cellular data and WiFi connectivity. We were able to access sites relevant to critical services that are often impacted during shutdowns.

Additionally, the meticulously crafted ViNET packets preserve packet sizes and timings akin to ViLTE, effectively thwarting large-scale traffic analysis with false positive rates up to 40%, when distinguishing ViNET from ViLTE. To summarize, the work delivers the following contributions:

- ViNET introduces a novel, high-performance approach to bypass Internet shutdowns by cannibalizing ViLTE streams for covert, encrypted data transmission, ensuring robust end-to-end privacy, confidentiality, and message integrity.
- A comprehensive performance evaluation showing that ViNET has significantly higher transmission rates compared to Dolphin, enabling more practical and secure access to Internet services. While Dolphin claims a throughput of 64 bps with low error rates, our measurements for ViNET indicate a throughput ranging from 50 to 400 Kbps, depending on the Internet Service Provider. These results indicate performance improvements of over 1000 times in terms of throughput when compared to Dolphin.
- Finally, a security analysis, showing that our technique is resistant against untargeted detection and blocking mechanisms, even under state-of-the-art classifiers suffer up to 40% false positive rates, making ViNET a viable solution for circumventing Internet shutdowns in highly censored environments.

2. Background and Related Work

2.1. Internet Shutdown

Internet shutdowns refer to the intentional disruption of Internet connectivity within specific geographic regions, ranging from cities to entire countries, implemented by authorised entities under government directive. Frequency of such occurrences has shown a remarkable increase recently, with 296 documented instances reported in the year 2024 [33]. The duration of such shutdowns varies considerably, extending from less than a day to periods exceeding a year, as evidenced by the 472-day shutdown in Chad [7]. It also noteworthy that most internet shutdowns are localised, instead of nationwide [34]. *E.g.* in most cases, in 2023 and 2024, 223 out of 283 and 214 out of 296 shutdowns, respectively, were restricted to specific regions (state or province).

There are a number of initiatives, both global and country-specific projects, which systematically document and analyse these shutdowns. For example, `accessnow` [33] project compiles comprehensive global statistics and incident reports, while there are also some region-specific efforts, such as [35], that maintain detailed records for countries like India, which has consistently recorded the very high number of shutdowns.

Additionally, there are certain initiatives that aim to estimate the economic impact of these disruptions, with organisations such as the Internet Society providing assessments of financial losses incurred due to Internet shutdowns [36].

Besides Internet shutdowns, there are other projects which focus on detecting general Internet outages. For example, the Internet Outage Detection and Analysis (IODA) system [37], employs both active measurements through network probes and passive techniques by analysing anomalies in publicly accessible BGP routing data to infer potential outage events. There are also some proprietary platforms, such as Cisco's ThousandEyes [38], which also contribute to real-time global monitoring of Internet disruptions.

While these platforms and studies provide valuable insights into the occurrence and impact of Internet shutdowns and outages, they largely fall short of offering practical mechanisms to mitigate or circumvent such disruptions.

2.2. ViLTE

ViLTE is short for Video over LTE is an extension of VoLTE (Voice over LTE) to provide person-to-person video call service. It is based on *IP Multimedia Subsystem (IMS)* core network [39]. ViLTE relies on LTE network architecture similar to VoLTE, consisting of the following primary components:

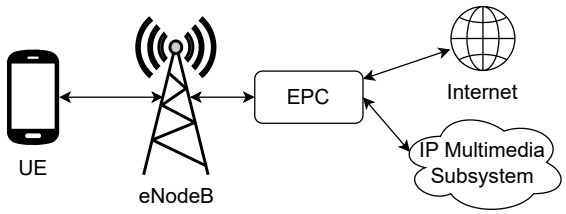

Figure 1: LTE network infrastructure.

- User Equipment (UE): It is the user's endpoint of the connection (e.g. smartphones) for accessing the LTE network.
- Evolved NodeB (eNodeB): It is the radio access component of the network. Which consist of the base stations that control radio communication between the Evolved Packet Core (EPC) and mobile devices. They are responsible for controlling resource allocation in their region.
- Evolved Packet Code (EPC): It is the core network that handles all the IP data traffic. The EPC forwards user plane traffic to the correct packet data network, e.g. to the Internet in case of web-browsing. In case of a ViLTE call, the packet data network is IP Multimedia Subsystem (IMS).
- The IP Multimedia Subsystem (IMS): It is an architectural framework for providing the IP multimedia services for LTE. So it provides IP-based telephone service for LTE.

Protocol Stack. ViLTE uses the same control plane protocol stack as VoLTE [40]. Session Initiation Protocol (SIP) is responsible for session establishment, modifications and termination. The Session Description Protocol (SDP) is embedded within SIP messages, it negotiates the media capabilities between the endpoints. For media transport, ViLTE utilizes Real-time Transport Protocol (RTP), which carries the actual video streams. RTP is designed for end-to-end, real-time data transfer and it works in coordination with the RTP Control Protocol (RTCP), which provides feedback on the quality of service [41]. In our experiments, we observed that ViLTE utilizes RTP for the application layer, UDP for transport, and IPv6 for the network.

Media Plane Implementation. For the media plane, ViLTE uses the H.264 codec to encode and decode the video stream. The video stream is packetised using RFC 6184 [40], which has the following payload format: a RTP header of size 12 bytes followed by the RTP payload. The RTP payload has a 2-byte Network Abstraction Layer Unit (NAL Unit) header, followed by the actual NAL Unit.

Media Plane Security. In ViLTE, the RTP data can be secured via encryption and integrity protection of the SRTP protocol. As per the media plane protection specifications, two scenarios are possible: 1) either the data is protected between the UE and IMS, which provides end-to-access-edge protection. 2) or both UEs protect their data with an end-to-end protection mechanism.

All the cryptographic operations in the case of LTE are handled by the Packet Data Convergence Protocol (PDCP), which is a part of Layer 2 [42]. However, ViNET captures the ViLTE packets at Layer 3, i.e., the network layer. Hence, we get access to the unencrypted RTP data. Thus, any of the above-mentioned implementations by any MNO will not impact the functioning of ViNET. Interestingly, having end-to-end encryption will have an advantage in making ViNET more resistant to the deep packet inspection attacks discussed in Section 6.

2.3. Related Work

We now present relevant prior work. Traditionally, censorship circumvention has explored various mechanism that have relied on proxying/tunnelling [19], [20], [43]–[46], traffic re-routing, *i.e refraction routing* [21]–[24], [47]–[52], and several times covertly hiding data in other applications' protocol messages like VoIP [53], [54] or online video channels [55]–[59].

However, most such solutions won't function during an Internet shutdown (*i.e.*, total disconnection of Internet access) where cellular data (and sometimes even broadband/landline networks) are shut down. Regular voice/video communications and SMS are usually unaffected, as they may aid essential communication [6], [60]–[63]. Previous research has investigated the feasibility of sending data over cellular voice channels, but most efforts have been theoretical, focusing on defining limits, proposing modulation schemes, or analysing specific codecs [64]–[71]. Notably, existing approaches have not attempted to run real Internet applications or addressed the challenges of doing so.

Dolphin [27] enables access to lightweight, delay-tolerant Internet applications (e.g., email, tweets, news) during Internet shutdowns by transmitting data over cellular voice channels. Users in restricted regions connect with peers outside the shutdown zone, who fetch and send content on their behalf. Data is encoded into audio and transmitted via voice calls. To address the unreliability of cellular voice channels, Dolphin implements a TCP-style reliability layer for loss recovery, ensuring stable data transfer across any encoding and modulation scheme. It also provides end-to-end confidentiality (between peers) to prevent eavesdropping and supports automation for seamless operation. Experiments confirm its feasibility, with applications like email and news retrieval functioning in minutes (e.g., a 500-character email transmitted in under two minutes).

However, Dolphin is unsuitable for semi-real-time modern web applications due to the high error rates and low goodput. Further, the solution doesn't work with modern end-to-end message confidentiality and authentication measures like TLS. Persistent trustworthiness of otherwise peers outside the shutdown region may be unfounded.

Additionally, some have explored the possibility of directly exploiting the voice packets of VoLTE [72], [73], which are known to have higher capacity compared

to traditional circuit-switched cellular voice. While such schemes show promise, our initial evaluations prove that as of 2024, modern cell phones do not provide a mechanism to intercept or capture VoLTE packets at the network interfaces of such devices. The VoLTE traffic generation is wholly managed through hardware. The kernel of the Android smartphones (even rooted ones) provides no mechanism to intercept and add data into the outbound packets without implementing some custom patch, which further limits the practicality of the design [74], [75].

In the recent past, several tools like *Anix* [76], *Moby* [77], *Rangzen* [78], *Briar* [25], *Firechat* [79], *Bridgefy* [80], *ASMesh* [81] have been proposed that are for circumventing Internet blackouts/shutdowns. These solutions mostly relied on mesh-based P2P networks, often using Bluetooth and ad hoc (unmanaged) WiFi networks as the communication medium. Most focused on helping users communicate with one another, like messaging apps, focussing on features like message authentication, anonymity, establishing and revoking trust *etc.* These solutions have not been designed to work with existing Internet applications like web browsing in mind.

3. Preliminaries

In this section, we outline the threat model considered in our work and detail the key assumptions made for the proper functioning of ViNET.

3.1. Threat Model

Internet shutdowns are typically enforced by Mobile Network Operators (MNOs) under directives from higher authorities. During these shutdowns, only the access to data network is restricted [62], [63], while leaving SMS, voice (VoLTE) and video call (ViLTE) services operational [28]. These basic services are essential for critical state services, and thus are not usually disabled, evident from recent shutdowns [82]–[84]. Moreover, since most cellular providers have now migrated to 4G/LTE or newer access technologies [29], LTE based services may not be trivial to turn off *without* impacting basic cellular voice calls. Thus, ViNET leverages such channels to transmit network packets between a caller in a shutdown-affected region and a callee located outside.

Hitherto there have been scant studies highlighting ViLTE, including situationslocations where it works or doesn't, to support this claim. But, multiple independent reports confirm that voice and SMS services typically remain functional even when mobile Internet access is restricted [28] and since ViLTE follows the same architecture as voice (VoLTE), it is fair to speculate that it will be operational during the internet shutdowns.

Further, blocking ViLTE during network shutdowns could raise serious ethical concerns, particularly when considering its role in critical services. For instance, video calling is increasingly integral to applications such as telemedicine, which has seen widespread adoption since the COVID-19 pandemic [85], [86]. Additionally, disabling ViLTE may carry regulatory implications. In regions like the EU, there is active promotion of ViLTE integration into emergency services such as 112 to enhance accessibility [87].

Moreover we ourselves verified that ViLTE remains operational during following regional Internet shutdowns in India (*i.e.,* country with most shutdowns [88]).

1) 27 September 2025, Bareilly - Uttar Pradesh, India, Internet Shutdown. ViLTE/VoLTE calls between the same MNOs were working. [89]
2) Feb 2024, Aurangabad - Maharashtra, India, Internet Shutdown. ViLTE/VoLTE call were working [90]

We assume a powerful adversary that can collude with the MNOs to observe the ViLTE traffic. Thus the adversary could employ deep packet inspection techniques to view the contents of the packet. These could involve inspecting the header and payload of ViLTE packets for known byte values/sequences.

The adversary may also employ ML (Machine Learning) based techniques on flow meta-data of ViLTE sessions (*e.g.*, packet sizes, throughput, inter-arrival times *etc.*) to determine if they bear any characteristics akin to those of ViNET.

Additionally, the adversaries may succeed in launching "targetted" attacks to disrupt ViNET calls against known suspects. They may attempt to render video frames of suspects/targets to determine visual distortions across frames, that may hint towards the use of ViNET. But since these type of attacks are logistically very challenging for the adversaries, as they cannot monitor all the ViLTE flows manually. Such scenarios are beyond our threat model.

Finally, large-scale signal jamming disrupting all cellular connectivity, and falls outside our threat model. Thus we posit that the adversary doesn't employ such techniques to disrupt ViNET as collateral damages from total communication disruptions may have severe socio-economic impacts.

3.2. Assumptions

The following assumptions outline the prerequisites for parties intending to use ViNET:

1) **Rooted devices:** Users are expected to possess a rooted Android device that supports ViLTE (Video over LTE) calls. Rooting devices may not be trivial for individual users, besides voiding warranties and preventing period updates/patches. Therefore, users need not use their primary mobile devices for this. A spare device, designated categorically for using ViNET, could be procured and rooted, when not experiencing shutdowns. ViNET's design allows sharing access to such a device between multiple users (*e.g.*, family members).
2) **Pre-shared symmetric encryption keys:** A shared secret key (256-bit in our implementation) must be securely exchanged between the parties before communication. The key exchange is assumed to occur over a secure and trusted channel. Shutdowns are usually restricted to certain regions [34] and occur repeatedly [28]. Therefore, when not experiencing shutdowns, users could get in touch with peers/friends outside such regions, willing to help run ViNET peers for the former and negotiate shared secrets through private/trusted communication channels.
   Additionally, we also propose a key establishment protocol that relies on RSA signed Diffie-Hellman key exchanges, assuming both parties have already exchanged RSA public keys. This is described in section 3.3.

3) **Peers using the same MNO:** Based on our experimental observations, ViLTE calls could only be successfully initiated and maintained between users subscribed to the same mobile network operator. This assumption is rather practical than restrictive. AccessNow data shows that countries with frequent shutdowns typically have concentrated telecom markets: 3–5 providers serve 90–95% subscribers [91]–[94]. This increases the chances of two peers sharing the same MNO, making the requirement relatively easy to meet. While this constraint may partly reflect our testing environment, ViLTE is known to function across carriers within a country [95] and, in some cases internationally [96]. Although ViLTE, as an extension of VoLTE, is theoretically designed to support inter-operator connectivity (as discussed in Section 2.2), we were unable to verify this functionality in practice. Therefore, we assume that both parties possess SIM cards from the same service provider.
4) **Localised internet shutdowns:** This assumption is reasonable since most shutdowns are localised *i.e.* usually happen in a small demographic region of a country(section 2.1) [1]. And since MNOs usually operate at nationwide scale, serving the customers across different regions (*e.g.,* in case of India all the major MNOs provide their services at national scale [98]–[100]), this will make it easy for the user to find a peer outside the shutdown zone.

3.3. Key Exchange Protocol

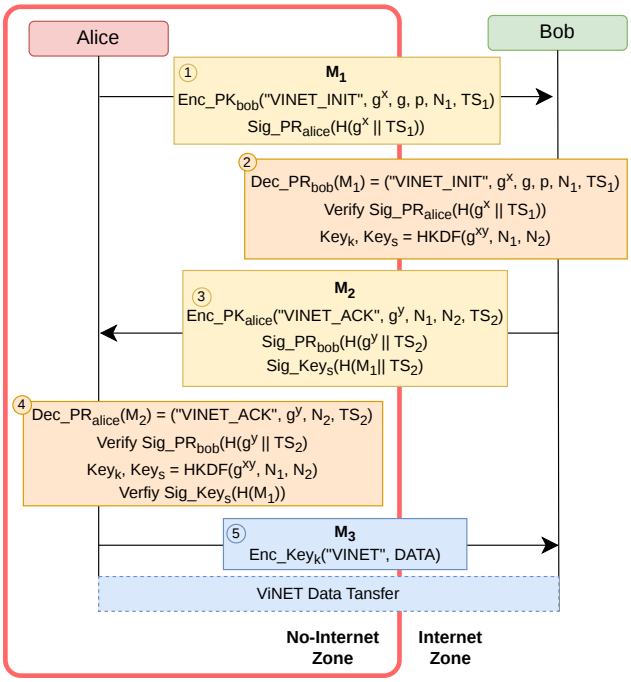

Figure 2: ViNET key exchange between Alice (in the shutdown region) and Bob (outside).

1. AccessNow [97] reports over 75% of shutdowns as localized, making ViNET usable in the majority of cases, as users inside the shutdown region can connect to peers in unaffected areas of the same country/state.

To explain the Key exchange protocol for ViNET, we consider Alice (residing in the shutdown region) and Bob (residing outside), willing to help Alice access the Internet. Alice establishes a ViLTE call with Bob. We assume that the key establishment protocol messages are embedded inside regular ViLTE packets similar to ViNET data packets as explained in subsection 4.2. The key exchange must be done before the actual data communication between the peers. The payloads bearing encrypted messages are derived from random values, revealing no information of the underlying messages. Additionally, we also assumed that Alice and Bob have exchanged their RSA public keys over some channel other than ViNET.

We now describe the protocol, the numbers in the figure 2 show the sequence of steps discussed here.

1) Alice sends $\{"VINET_INIT", g^x, g, p, N_1, TS_1\}$ encrypted with the RSA Public Key of Bob along, with the signed hash of $\{g^x, TS_1\}$ to Bob (denoted as M_1). Then Alice waits for a response from Bob for a fixed interval. If she timeouts then she retransmit the message with an updated timestamp (TS_1) and nonce (N_1).
2) Bob decrypts the incoming packet from Alice using its corresponding Private key and get $\{g^x, g, p, N_1, TS_1\}$. Bob verifies the signed hash using the public key of Alice. Bob also verifies the timestamp (time since epoch) to detect replays. Then Bob chooses a DH parameter y and random nonce N_2 and computes session key Key_k and signature key Key_s using $HKDF\{g^{xy}, N_1, N_2\}$, that are used subsequently for encrypting and signing protocol messages.
3) Bob then sends $\{"VINET_ACK", g^y, N_1, N_2, TS_2\}$, encrypted with Alice's public key, along with the signed hash of $\{g^x, TS_2\}$ signed using RSA Private Key and a signed hash of previous message received from Alice (M_1), using Key_s. Then Bob starts to wait for Alice to send some ViNET data, to confirm that the key exchange was successful. But if he timeouts then he retransmits the message (M_2) with an updated timestamp (*i.e.*, updated TS_2) and new nonce (N_2).
4) Alice decrypts the incoming message (M_2) from Bob, using its RSA Private, deriving $\{g^y, N_2, TS_2\}$. Alice then verifies the received DH parameters and the timestamp (TS_2) using Bob's public key. Once she authenticates that the messages are from Bob, she computes the same session encryption and signature keys as Bob (*viz.*, Key_k and Key_s) using $HKDF\{g^{xy}, N_1, N_2\}$. Using Key_s, Alice verifies that the signed hash of message M_1 that she previous transmitted to Bob (in step 1), confirming that both have now the same set of keys and the handshake messages haven't been replayed.
5) Then Alice will start ViNET data transfer, where data will be encrypted using the shared secret session key K.

4. System Design

To explain the working of ViNET, we consider Alice (residing in the shutdown region) and Bob (residing outside), willing to help Alice access the Internet. Alice

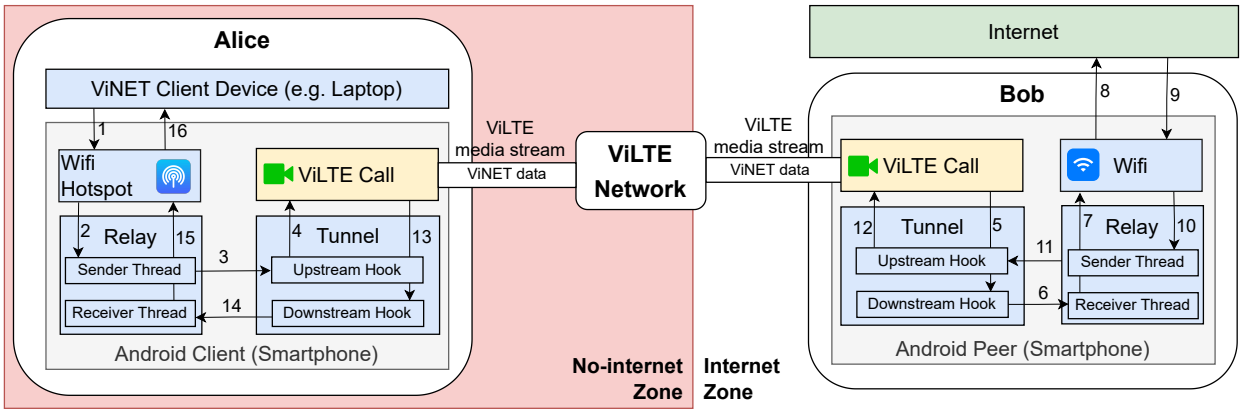

Figure 3: ViNET protocol data flow between Alice (in the shutdown region) and Bob (outside).

establishes a ViLTE call with Bob. Thereafter, Alice's desktop/laptop tethers to the phone over WiFi hotspot, and sends network traffic through it. ViNET application suite, consisting of various entities, intercepts these packets, encrypts them using a pre-shared key (explained in detail in Section 4.2.3), and embeds them in outbound video packets (RTP). The process relies extensively on the Linux Netfilter framework.

Bob's phone then forwards these video packets to an application (that could run on a either Phone or a tethered laptop) that extracts the data and decrypts it. Finally, Bob's laptop forwards these decrypted packets (could be full fledged TCP packets) to the intended destination. The responses from the website follow the exact same mechanism (also relying on Linux Netfilter) for encryption and encapsulation, before sending it back to the Alice's phone that decrypt and decapsulates/extracts them from the video packets and sends it to her client application (*e.g.* browser).

Alice's trusts for Bob doesn't rule out the possibility of Bob from curiously eavesdropping on Alice's data, if not compromising the integrity. Earlier solutions like Dolphin [27] has no provision to support end-to-end message confidentiality and authentication/integrity protocols. But ViNET's design is transport and application layer agnostic and thus natively supports end-to-end TLS handshakes between the client and the actual webserver, achieved using the method described above.

Also, ViNET carefully embeds TCP/UDP/IP packets in ViLTE packets in a manner that neither changes their sizes nor their transmission rates. As shown later in Section 6.1, this clever mechanism helps defend against adversaries by maintaining FPRs over 20%.

4.1. ViNET's Traffic Flow

We now describe in detail how Alice uses ViNET's protocol (Figure 3) for connecting to a website involving a video call with Bob. The numbers in the figure show the sequence of steps discussed here.

1) Alice uses a web client (*e.g.,* `lynx` browser) on her device (*e.g.,* laptop/desktop) and attempts to connect to a website. The generated TCP/IP packets are sent to the Android Client device (*i.e.,* Alice's phone) via WiFi tethering.
2) A *Relay* application, running on Alice's phone intercepts these packets using raw sockets. The application runs two threads-one for sending data to the *Tunnel* (sender thread), and the other for receiving the data from it (receiver thread).
3) Two stream sockets are created connecting the Relay and the *Tunnel* application; one to send the length of the data packets (data_length_socket), and the other to send the data packets (data_socket). After intercepting the TCP/IP packets, ViNET Relay sends the packets and their lengths to the Tunnel application via these sockets.
4) The Tunnel runs two threads—one intercepts (via `netfilter` hooks) the outgoing/upstream traffic, and the other intercepts the incoming/downstream traffic. For every outgoing packet, the Tunnel application inspects the data_length_socket to check for data that be sent. For available data packets, Tunnel checks whether they can be encapsulated in ViLTE packets, without modifying their lengths. The payload of the ViLTE packets is replaced with encrypted packets (discussed in detail in Section 4.2.1), besides updating the underlying packets' UDP checksums. The netfilter hook function then passes on the packets to the kernel, to be transmitted to Bob (*i.e.,* as part of the ViLTE call). In the absence of data packets from relay, the above processing is skipped, and the hooked ViLTE packet is sent back to the kernel.
5) Tunnel application running on Bob's phone intercepts the ViLTE packets sent by Alice and extracts the video payload, that are decrypted assuming they contain ViNET payload. Then it searches for the magic word "ViNET" in the header (Figure 5). Once found, the Tunnel extracts the ViNET data and sends the decrypted ViLTE packet back to the Android Kernel, to be played out. If the "ViNET" signature is absent the the packet is sent back to the kernel for further processing.
6) After extracting the ViNET data, Tunnel sends the data packet and their lengths to the Relay application via the stream sockets.
7) Bob's Relay's receiver thread records the packets' source and destination IPs, before sending them to the WiFi interface of the device, using raw sockets.
8) The packets are source NATed using `iptables`,

while leaving the WiFi interface for the destination. The aforementioned recorded addresses are useful to invert the NATed addresses for the response packets from the requested site.

9) Thereafter, Bob's device receives the response from the destination/website.
10) The sender thread of the Relay intercepts all the packets received from Internet on the WiFi interface, using raw sockets and checks if they are responses to previously sent requests (by Alice). Otherwise, they are assumed to be for the device itself.
11) Thereafter, the Relay inverts the NATed addresses to point back to Alice's IP address, and forwards them, along with their lengths to the Tunnel, using stream sockets.
12) A thread running on the Tunnel, reads the packets sent by the Relay and injects them to the the upstream (outbound) ViLTE traffic (similar to step 4 where Alice's requests are sent). Finally, the modified ViLTE packets, bearing the destination's/website's response embedded in their payloads, are sent back to Alice over the video call.
13) On Alice's phone, a thread receiving the downstream (inbound) traffic, searches for ViNET data in the extracted and decrypted data from ViLTE packets (similar to step 5). These packets are then sent to the Android Kernel.
14) The Tunnel then forwards the extracted ViNET data packet and its length to the Relay application over the shared stream sockets (like in step 6, but on Alice's device).
15) The receiver thread of the Relay reads the packets sent by the Tunnel application and sends them to the WiFi hotspot interface of the Android client device using the raw sockets.
16) The WiFi hotspot treats these packets as responses from the Internet and sends them to the ViNET Client device (*i.e.* Alice's laptop/desktop), where the web client has been waiting for the response from the website.

TCP/IP and TLS packets transmitted (from Alice to Bob and back) via this data flow help establish an end-to-end encrypted connection, resistant to eavesdropping or tampering.

4.2. Implementation and Design Choices

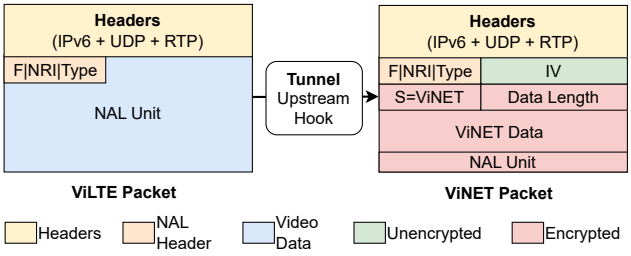

Figure 4: ViNET upstream payload modification.

4.2.1. Modification of upstream ViLTE packets:. While modifying ViLTE packets for sending ViNET data, the Tunnel application intercepts an outgoing ViLTE packet.

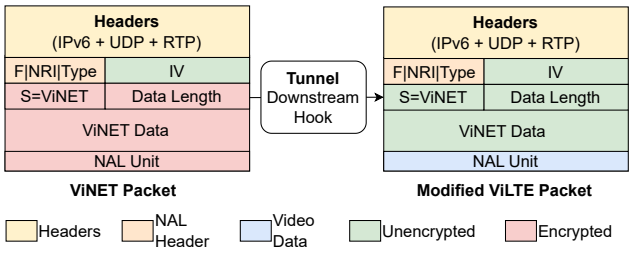

Figure 5: ViNET downstream payload modification.

As illustrated in Figure 4 (left), A ViLTE packet comprises IPv6, UDP, and RTP headers, with the RTP payload containing a 2-byte NAL (Network Abstraction Layer) header followed by a NAL unit that holds the actual video data. The NAL header is critical for decoding the RTP data and is therefore left unmodified. Instead, only the NAL unit is altered to embed ViNET data.

To prepare the ViNET data for embedding shown in Figure 4 (right), ViNET headers are added, which include a 16-byte Initialization Vector (IV) for AES (CTR mode), a 5-byte magic word S=`ViNET`, and a field indicating the length of the ViNET data. The ViNET data, along with this magic word, is encrypted using a pre-shared key between the communicating parties, Alice and Bob, to ensure confidentiality and prevent unauthorized access.

During modification, only the necessary bytes of the NAL unit are replaced with the ViNET data. If the ViNET data is x bytes long, and the NAL unit is of length y (where $y \geq x$), then only the first x bytes are modified, leaving the remaining $y - x$ bytes intact. This minimizes disruption to the original video data, preserves ViLTE channel integrity, and conceals ViNET traffic by maintaining the original packet length.

After embedding ViNET data, its length is adjusted in the header to aid extraction by the receiver. The new payload, comprising the ViNET header (IV, signature, data length), encrypted ViNET data, and remaining NAL unit bytes, replaces the original ViLTE payload. The packet's checksum is recalculated before transmission to comply with ViLTE infrastructure requirements, ensuring secure transmission of ViNET data without disrupting the video stream or exposing the embedded data.

4.2.2. Modification of downstream ViLTE packets:. When the Tunnel application receives a ViLTE packet containing ViNET data, it passes it to the downstream hook (Figure 3) of the Tunnel, extracting the ViLTE payload while preserving all headers (including NAL headers). As Figure 5 (left) shows, the payload is then decrypted using a 16-byte initialization vector (IV) and a pre-shared key between Alice and Bob. If a ViNET magic word (S) is detected, the packet is identified as a ViNET packet, and the relevant ViNET data is extracted using the Data Length field. This extracted data and the Data Length are forwarded to the Relay application for further processing. To minimize bit errors in the ViLTE stream, the encrypted payload is replaced by its unencrypted version. Additionally, to suppress the generation of excessive RTCP error packets, the Forbidden Bit (F) in the NAL header is set to 1, as specified in RFC 6184 [101], advising the decoder that the NAL unit may contain bit errors or syntax

violations, prompting it to discard the unit and conceal lost data, *thereby reducing error RTCP messages*. Thus, it will force the application to ignore the bit errors and not send RTCP error packets to the sender, which helps in preserving the traffic signature of ViLTE. Finally, the modified ViLTE packet, now containing unencrypted data with embedded ViNET data and the F bit set, is returned to the nf_queue for further application processing.

4.2.3. Why encrypting ViNET data:. As mentioned earlier before embedding ViNET data inside the ViLTE packet, the complete ViLTE payload is encrypted with the pre-shared key between Alice and Bob using AES in CTR mode. This is done to prevent the adversary from observing at the payload to inspect plaintext protocol elements like protocol headers, TLS handshake *etc.*, as the presence of such elements will indicate the adversary about the use of ViNET.

4.2.4. Reliability layer of ViNET:. ViLTE operates over UDP and lacks inherent reliability-ensuring mechanisms. Thus, ViNET depends solely on the guarantees provided by the protocols it transports (*e.g.,* TCP). This simplifies the design and avoids modifying the ViLTE protocol, which could otherwise undermine the integrity or make it detectable.

TABLE 1: Comparison of MNO 1, MNO 2 and MNO 3 ViLTE packet sizes.

Metric	MNO 1	MNO 2	MNO 3
Mean	590.69	1070.01	459.56
Std	191.23	105.85	121.53
Min	227.34	612.23	293.19
Max	1102.04	1198.55	983.88

4.2.5. ViNET packet sizes:. To assess complete TCP/IP packet encapsulation in ViLTE packets, we analysed the latter's packet lengths (Table 1). Since notable variations were observed across MNOs, we optimized packet lengths by modifying the WiFi interface's MTU using empirically obtained values (MNO 1: 600 bytes, MNO 2: 1000 bytes, and MNO 3: 500). This restricts the client from sending packets of length larger than specified. These values are added to the MSS value of the TCP header options, of the TCP SYN and RST packets, using `iptables`. Which in-turn, restrains the server from sending packets larger than the specified MSS, forcing TCP packets to be smaller than most ViLTE packets, aiding easy encapsulation and maximizing ViLTE capacity utilization.

4.2.6. Multithreaded tunnel:. The Tunnel uses multi-threading to manage ViLTE traffic, which uses the RTP over UDP, for simultaneous video feed transmission between parties. Sequential processing of upstream and downstream ViLTE traffic in a single nf_queue would reduce throughput and raise processing times, increasing traffic analysis risks. Instead, two threads, each handling a separate nf_queue, process upstream and downstream traffic independently, enhancing performance and resisting traffic analysis.

4.2.7. Using a WiFi-hotspot:. The WiFi-hotspot based architecture allows Alice's multiple client devices to connect to Bob's setup, through the ViNET "tunnel" between Alice's and Bob's devices, to fetch data. At Bob's end, WiFi was chosen for Internet packet forwarding.

5. Experimental Evaluation

We now present ViNET benchmarking experiments, covering test setup and results. To evaluate the performance of ViNET, we performed a variety of tests, both in controlled and over real Internet applications, to determine the robustness and real-world use case of our system.

5.1. Test Setup

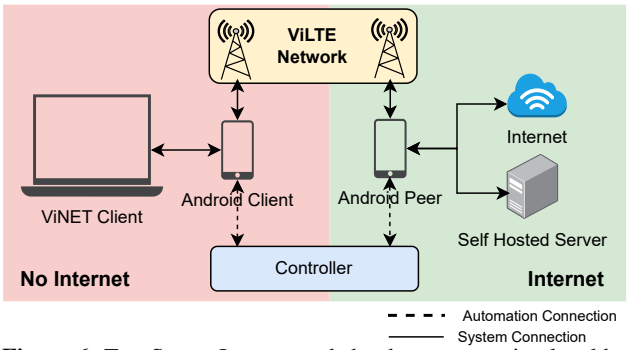

Figure 6: Test Setup: Internet and shutdown zones simulated by switching off Internet connectivity at the client devices.

To assess ViNET's performance, we used two Motorola G40 Fusion phones (Android v12), one phone simulating Alice in a shutdown region and the other as Bob outside, tunnelling data via ViLTE's RTP channel (*i.e.*, ViNET). The cellular data, as well as WiFi calling, were disabled on both phones, and a video call, using operator-provided services, was established between them.

The experimental setup utilised two laptops to emulate the ViNET client and controller for automated testing. One of the laptops functioned concurrently as the ViNET client—tethered to Alice's smartphone via a Wi-Fi hotspot—and as the automation controller. The first device was equipped with an AMD Ryzen 7 7730U processor, 16 GB of RAM, and operated on Debian 12 with Linux kernel version 6.1.0-33-amd64. The second device featured an Intel Core i5-6200U processor, 8 GB of RAM, and ran Ubuntu 20.04 LTS with Linux kernel version 6.11.0-19-generic.

We tested ViNET across three major MNOs, covering >90% subscribers in our region, with 10 experiment iterations. The setup enabled systematic evaluation of ViNET's data tunnelling functionality under simulated shutdowns.

All the code for ViNET is written in C++ and is compiled using `aarch64-linux-android-g++` compiler, which is a part of Android Native Development Kit (NDK) used to compile code for x64 ARM architecture [102].

5.2. Controlled Experiments

We first perform a few controlled experiments to assess the raw performance and efficiency of ViNET under

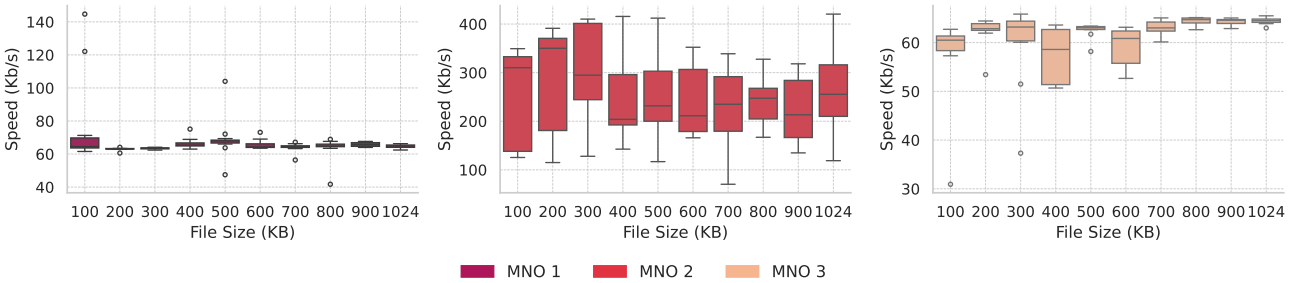

Figure 7: Download speeds (over HTTP) for different file sizes using ViNET.

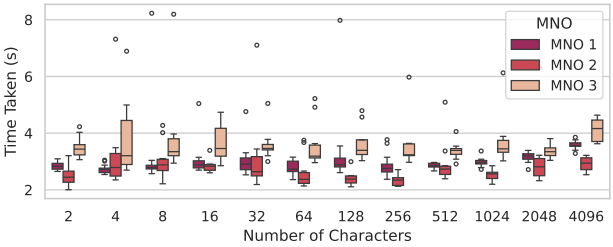

Figure 8: Time taken to send Telegram text messages of different sizes iterations using ViNET.

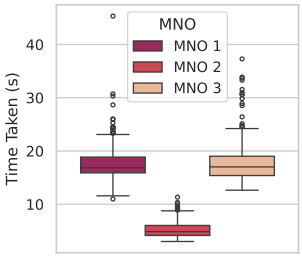

Figure 9: Webpage load times from websites in the custom dataset using ViNET.

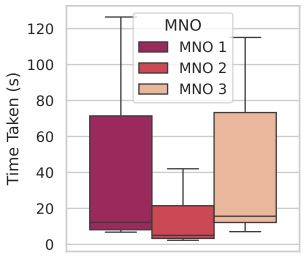

Figure 10: Webpage load times from websites in the Tranco dataset using ViNET.

isolated and deterministic conditions. This setup ensures that the core capabilities of ViNET are evaluated without interference from external Internet-related factors. For this, we first check the suitability of the ViLTE channel for Internet data transfer by observing the throughput it offers. Next, we employ ViNET and conduct iperf and file download-based throughput experiments.

TABLE 2: ViLTE downstream throughput in Kilo bits per second for different MNOs.

Metric	MNO 1	MNO 2	MNO 3
Mean	67.30	428.27	59.62
Std	29.67	182.00	16.52
Min	24.02	95.13	31.97
Max	272.56	762.10	143.73

First, we measure the throughput of the ViLTE call to gauge the theoretical maximum throughput of our proposed ViNET system, setting it as the ideal baseline for our system. For the same, we collect a packet trace while a ViLTE call runs. Table 2 shows the throughput values for three MNOs. We observe that MNO 2 provides the highest throughput amongst these three MNOs.

5.2.1. `iPerf3`. Next, we performed `iPerf3` throughput tests to benchmark the performance of ViNET. For this, the client in the shutdown region connects to an `iPerf3` server running on a self-hosted server in the Internet region. We kept `iPerf3` at its default settings to measure the throughput of ViNET. Table 3 shows the throughput measured through `iPerf3`. Note that it has a similar pattern as measured over a normal ViLTE call in Table 2

5.2.2. File Download over HTTP. To further characterise network performance, we conducted a comprehensive file download experiment across the three MNOs. The experimental setup involved a client positioned in the shutdown region downloading files of varying sizes (100 KB to 1 MB) from an *Apache* HTTP server deployed in the Internet Region. File transfers were performed using `curl`,

TABLE 3: Available capacity measured using `iPerf3`, through ViNET, for different MNO (Kbps).

Metric	MNO 1	MNO 2	MNO 3
Mean	79.06	388.43	61.24
Std	29.55	81.38	4.81
Min	35.92	260.56	53.54
Max	126.37	519.57	67.19

with each test configuration repeated 10 times to ensure statistical significance.

Figure 7 presents the experimental results, which demonstrate significant performance variations across MNOs. Our analysis reveals that MNO 2 achieves superior throughput performance relative to MNOs 1 and 3, albeit with higher variance in download times. MNO 1 exhibits more consistent performance characteristics with reduced variance compared to MNO 2. MNO 3 demonstrates throughput performance comparable to MNO 1. These empirical findings corroborate our baseline throughput measurements documented in Table 2 and Table 3, thereby validating the reproducibility and consistency of our performance evaluation methodology.

5.3. Internet Applications

Now, we evaluated ViNET's performance for Internet applications. We consider social media and web browsing applications.

5.3.1. Social media applications (Telegram):. We evaluated ViNET's performance for sending Telegram text messages ($\leq$ 4096 characters) using Telegram's API and Python automation scripts. Over 10 iterations, ViNET delivered messages in under 4 seconds across three MNOs, demonstrating practicality (Figure 8).

5.3.2. Webpage load-time:. ViNET's primary objective is to enable users to maintain access to essential services

during Internet shutdowns. To evaluate this capability, we assessed ViNET's performance in accessing critical Indian websites. India was selected as the focus of this evaluation due to its documented history of frequent Internet shutdowns, representing one of the highest rates of such disruptions globally.

We chose the Integrated Government Online Directory (IGOD) [103], an initiative by the Indian government that offers centralised access to official websites across various administrative levels, including district administrations. These websites often contain vital information such as helpline numbers, hospital details, and other emergency services, which are particularly crucial during a crisis. We consider the district websites from a region in India (Jammu and Kashmir) that often experience shutdowns [60]. We show the full list of websites in Section A in Table 8.

We used `curl` to fetch these sites and measured the total page load times in 10 iterations for each site. Figure 9 shows the page load time for all three MNOs. The median page load time for most websites was less than 5 sec for MNO 2, making it suitable to access essential services during Internet shutdown. The page load times for MNO 1 and MNO 3 are a bit on the higher side, i.e., median more than 15 sec for most. However, still under half a minute.

5.3.3. Tranco-webpage loadtime:. To ensure a comprehensive evaluation, we also included websites from the Tranco dataset [104]. We filtered the dataset to select the top 10 news and information-related websites that were accessible to us. The filtering process was assisted by a large language model, as detailed in the Section B.

The selected websites were then fetched using the `curl` command to measure total page load times across 10 iterations. Figure 10 presents the page load times recorded for all three MNOs. The median page load time MNO 2 was about 5 sec and about 10 sec for MNO 1 and MNO 3. These observations are similar to the observations for essential websites.

TABLE 4: Time taken (in seconds) for Google search across 10 iterations using ViNET for all MNOs.

Statistic	MNO 1	MNO 2	MNO 3
Mean	6.99	2.92	5.25
Std	2.69	1.24	1.53
Min	4.81	2.15	2.53
Max	12.82	6.36	7.44

5.3.4. Google search:. Next, we evaluate ViNET in terms of the latency of searching a query over Google. We crafted a Google search query (explained in Section C) and performed 10 iterations of the same over both MNOs. The query search times are reported in Table 4. The median search time for MNO 2 is 2.92 sec, for MNO 1 and MNO 3 it is 6.99 sec and 5.25 sec respectively.

5.4. Takeaways

Our extensive evaluation of ViNET demonstrated that it is suitable for running real-time applications in an Internet shutdown zone with acceptable latencies, considering that no such systems are available yet. Access to frequently visited websites was possible with a median load time of about 15 sec (for MNO 2 it was about only 5 sec). Such latencies are a testimony of ViNET's resilience under shutdown.

ViNET provides adequate bandwidth for browsing using a text-based web browser(`Lynx` browser [105]) in real-time. It is also possible to chat comfortably (with 4K characters delivered in under 5s) using applications like Telegram.

ViNET provides maximum data rates from 60 Kbps to 400 Kbps, comparable to 2G data speeds [30], [31], making it a viable option for accessing Internet applications in real-time while preserving end-to-end confidentiality and integrity, absent in previous systems.

6. Discussion

In this section, we present the possible attacks to detect ViNET, its current limitations, and the key experimental challenges encountered during our evaluation.

6.1. Traffic Analysis Attack

We analysed whether an adversary could use traffic analysis to distinguish ViLTE from ViNET calls. ViNET encapsulates non-media data like HTTP traffic within ViLTE's RTP streams to bypass Internet shutdowns. A potential adversary may attempt to distinguish between standard ViLTE and ViNET sessions by analysing statistical patterns in traffic features such as packet sizes, directions, and inter-arrival times. Our goal is to assess whether such an adversary can reliably detect ViNET traffic without incurring high rates of collateral damage.

To establish a detection baseline, we adopted the MPTAnalysis [106] framework, which represents the state-of-the-art in identifying tunnelled data in RTP-based media flows, particularly WebRTC. MPTAnalysis distinguishes between legitimate media traffic and tunnelled application data using two feature sets: (1) Summary Statistics (Sum Stats), capturing temporal and structural properties of RTP streams (e.g., packet size variance, burstiness), and (2) Quantized Packet Lengths (Quant PLs), which model the frequency distribution of packet sizes.

We collected 60-second traffic traces for both ViLTE and ViNET sessions across three major MNOs, recording over 500 sessions per MNO, over three days. ViNET sessions emulated intermittent web browsing by transferring 100KB HTTP files every 5 seconds. This dataset spans diverse network conditions and carrier-specific implementations, ensuring realism in modelling traffic behaviour.

Following MPTAnalysis, we extracted both feature sets and trained XGBoost classifiers with session-level labels. Performance was evaluated across varying time windows (10–60s), using False Positive Rate (FPR), Accuracy, and F1-score as metrics, and ROC curves for robustness.

Table 5 summarizes classifier performance. Across all MNOs, high FPRs were observed, ranging from 10% to 40%, with only MNO 2 and the Summary Statistics feature set achieving lower FPRs ($\approx$2%).

TABLE 5: Performance Metrics of SOTA for distinguishing ViLTE and ViNET traffic.

Carrier	Metric	Quantized Packet Lengths						Summary Statistics					
		10s	20s	30s	40s	50s	60s	10s	20s	30s	40s	50s	60s
MNO 1	FPR	12.91	13.17	12.19	11.81	13.75*	10.22†	14.43	13.60†	13.80	15.67	17.64*	14.30
	Accuracy	80.63†	79.55	78.97	78.59	77.96*	79.78	83.66†	82.36	81.22	80.15	79.11*	79.49
	F1-score	78.12†	76.74	75.63	75.06	74.57*	76.10	82.74†	80.84	79.50	78.44	77.57*	78.27
MNO 2	FPR	30.29*	29.97	25.82	27.21	26.67	20.04†	2.62*	1.93	2.09	1.47†	1.88	1.67
	Accuracy	61.61*	62.89	65.01†	63.87	63.42	62.67	94.10*	94.80	95.48	95.44	94.62	95.92†
	F1-score	51.45	52.43	53.63†	52.53	51.65	50.78*	92.67*	93.75	94.91	95.02†	93.70	94.99
MNO 3	FPR	20.06†	21.5	23.48	25.82	24.28	26.69*	19.07†	22.26	23.2	22.62	23.66	25.76*
	Accuracy	72.62†	70.75	68.83	67.19	67.23	65.84*	78.43†	78.29	76.52	75.74	75.76	75.11*
	F1-score	70.43†	68.2	66.03	64.28	63.89†	62.56*	77.55	78.49†	76.52	75.07*	75.43	75.10
MNO 2 ‡	FPR	40.11	34.32†	36.42	37.08	40.71*	35.37	8.17†	9.22	10.32	11.38	11.50*	10.17
	Accuracy	58.10*	60.45 †	60.26	60.02	59.28	60.45	93.26†	91.70	90.62	90.54	89.80*	91.37
	F1-score	56.96*	57.71	58.55	58.42	58.88†	58.02	93.39†	91.69	90.61	90.66	89.83*	91.28

‡ represents the results when the amount of data tunnelled per session is reduced.
† represents the most favourable scenario for the adversary.
∗ represents the least favourable scenario for the adversary.

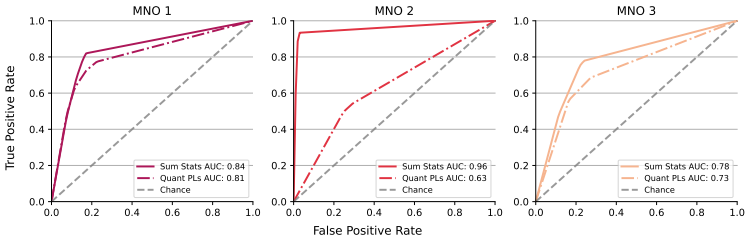

Figure 11: ROC curves of SOTA for distinguishing ViLTE and ViNET traffic.

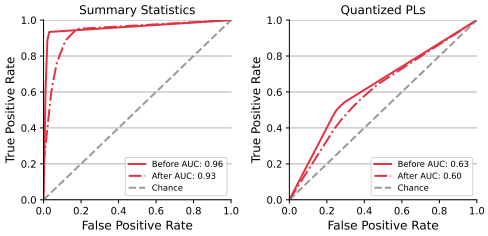

Figure 12: ROC curves of SOTA on MNO 2 when the amount of covert data transferred in a session is reduced to only 1 download per minute.

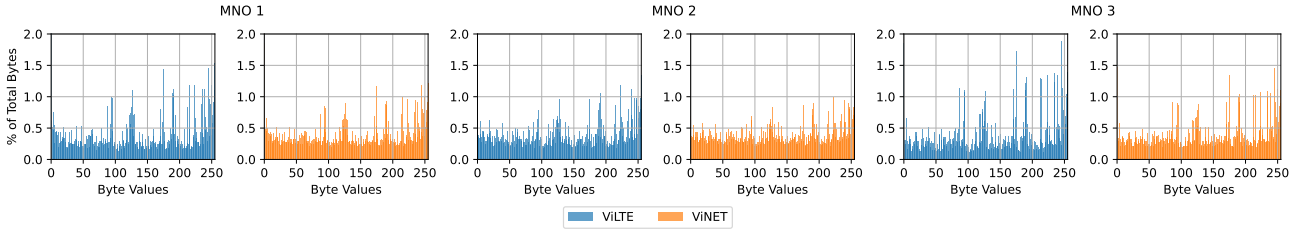

Figure 13: Distribution of byte values in ViLTE and ViNET payload for different MNOs.

Figure 11 further illustrates the detectability challenge: AUC values typically hover around 0.8, indicating that high true positive rates come at the cost of high false positives. The best-case scenario (MNO 2, Summary Stats) yields an AUC of 0.96, while Quantised PLs consistently underperform (*e.g.*, 0.63 for MNO 2).

To enhance ViNET's resistance in case of MNO 2, we reduced covert traffic to one HTTP download per minute. This reduction degraded classifier performance, with AUC dropping from 0.63 to 0.60 for Quantised PLs and from 0.96 to 0.93 for Summary Stats, dramatically increasing the FPR from $\approx 2\%$ to as high as 11.5%. Such slower downloads are akin to bursty network patterns seen during web browsing, which depend largely on how users navigate and visit webpages.

Prior work [107] indicates that even 1% FPRs incur significant economic costs due to low circumvention traffic volumes. False positives, as noted, cause unsustainable collateral damage, with economic losses from blocking legitimate ViLTE traffic leading to customer dissatisfaction and revenue loss [108]. Thus, even moderate FPRs render large-scale ViNET detection classifiers economically unviable.

Our findings show that while statistical features help distinguish ViNET from ViLTE, by design ViNET offers meaningful resistance to traffic analysis, making it difficult for adversaries to reliably detect and block without incurring significant collateral damage.

It's worth noting that our evaluation assumed a closed world model, with highly favourable conditions for an adversary: all devices are geographically co-located and operating under strong, stable network connectivity. These controlled conditions provide a best-case scenario for adversarial detection, thereby offering a conservative lower bound on the stealth and robustness of ViNET. These results may likely degrade (rather than improve) under more realistic, large-scale deployment scenarios. In practice, end-user devices will experience diverse network environments, like variable signal strength, differing MNO performance, and network congestion. Moreover, the heterogeneity in smartphone hardware may further impact system behaviour and potentially reduce detection accu-

racy.

6.2. Deep Packet Inspection

TABLE 6: Mean and standard deviation of Shannon's Entropy for NAL Units in ViLTE and ViNET Streams across all 3 MNOs.

Provider	Entropy	ViLTE	ViNET
MNO 1	Mean	7.00	7.02
	Std. Deviation	0.74	0.82
MNO 2	Mean	7.42	7.42
	Std. Deviation	0.67	0.68
MNO 3	Mean	6.91	6.93
	Std. Deviation	0.62	0.71

Since IMS traffic is not necessarily end-to-end encrypted, adversaries could lawfully intercept and analyze packet contents, for specific byte sequences (often at specific offsets), using Deep packet inspection (DPI). They could use DPI-capable middleboxes and firewalls to potentially distinguish ViLTE from ViNET traffic. However, in ViNET, the encrypted payload is embedded in RTP NAL units (Sections 4.2.1 and 4.2.2), leaving no identifiable patterns or fixed sequences that the middleboxes easily spot.

But an attacker may inspect ViLTE's payloads, expecting media-specific RTP NAL unit data, absent in ViNET's encrypted payloads. Predictable byte patterns in ViLTE's NAL units, missing in ViNET, could signal tunnelling. For scalable detection, adversaries may compute the entropies of the ViLTE packets, as repeating/identifiable sequences (*likely* in regular ViLTE payloads) could reduce it, compared to ViNET AES-encrypted payloads.

To test these claims, we computed the entropies of ViLTE and ViNET payloads for the data captured and used in Section 6.1. Evidently (Table 6), both ViLTE and ViNET packets have equally high entropies, for all three service providers, indicating the absence of detectable patterns or byte sequences, making DPI non-trivial. This is likely due to the variations in frame and pixel data encoding in the RTP payloads.

Further, Figure 13 shows the byte value distributions of the RTP NAL units of ViLTE and the encrypted payloads of ViNET. The distributions exhibit similar patterns without discernible skew. We additionally computed the KL divergence to quantify the distance between the two distributions. As reported in Table 7, the divergence values between the byte-value distributions of ViLTE and ViNET are minimal i.e. close to zero, indicating that the two distributions are highly similar.

TABLE 7: KL Divergence Scores between ViLTE and ViNET byte value distributions

Provider	KL-Divergence Score
MNO 1	0.103
MNO 2	0.005
MNO 3	0.168

Hence, distinguishing is non-trivial using DPI-based traffic classification. This observation aligns with prior work [109], indicating that distinguishing compressed data (using H.264 and H.265 codecs, used in ViLTE calls) from encrypted data is particularly challenging. Specifically, for payloads of 2048 bytes (significantly larger than ViLTE's, Table 1), binary classification, as reported in [109] accuracy did not exceed $approx$ 65%, which they claim would decrease if packet sizes dropped. However, ViLTE's/ViNET's payload never exceeds 1200 bytes Table 1. Taken together, these results demonstrate that ViNET traffic is resistant to identification, even by sophisticated deep packet inspection techniques employed by adversaries.

6.3. Rendering ViLTE Stream

ViNET's substitution of ViLTE's H.264 NAL units with encrypted data may produce visual anomalies like corrupted frames, unlike ViLTE's coherent video. To an adversary, these artifacts may signal the use of ViNET. However, network issues like packet loss or jitter could cause similar distortions in ViLTE, complicating adversarial detection. Quantifying distortions demands analyzing numerous streams, incurring significant computational and storage costs, rendering large-scale monitoring impractical and limiting attack feasibility. Though this attack lacks scalability, adversaries could target specific/suspected ViNET users. However, they would probably use such tactics cautiously, as network-induced visual distortions might lead to false positives.

6.4. Curious Peers Observing Sites Visited

As mentioned earlier, we assume peers are trusted, yet curious. Such peers could observe the sites visited by the ViNET users (in the shutdown region), by observing the TLS handshake messages, decapsulated and transmitted to their intended destinations. To prevent this, ViNET clients could rely on establishing VPN connections over ViNET first and then accessing sites. This way, the peer only observes packets going to the VPN gateways, without knowing their intended destinations.

6.5. Packet-drop based distinguishing attacks

Prior work [110] highlights significant vulnerabilities in censorship circumvention systems like SkypeMorph [53] that utilizes voice/video protocols as cover channels. The paper details successful attacks against SkypeMorph based on identifying, dropping, and replaying its internal Acknowledgment (ACK) packets. This is possible because, in SkypeMorph, the ACK packets could be identified and tracked by the censor.

Such attacks are not directly applicable to ViNET because its payload, which contains the underlying transport layer control packets (like the ACKs), is fully encrypted by a Pre-Shared Key (PSK) before encapsulation inside the ViLTE packet. Therefore, an adversary cannot inspect the ViLTE packet header or payload to selectively target and drop the ACKs, which are essential for the throttling or stalling attacks described in the said paper. Thus in case of ViNET, an adversary is limited to only non-targeted random packet dropping. While the underlying TCP connection could recover from fewer such drops, it may fail to do so when such drops increase, causing its termination. Moreover, that could also hamper the service quality of ViLTE calls of oblivious users.

6.6. Limitations

The ViNET system enables covert data transmission over ViLTE channels to provide connectivity during Internet shutdowns but faces technical and operational constraints limiting its reliability, flexibility, and scalability. These stem from ViLTE's protocol stack, MNO-specific architectures, and strict device requirements. While ViNET outperforms systems like Dolphin, addressing these limitations is crucial for practical deployment in censorship circumvention. We analyze each constraint's technical basis, performance impact, and mitigation strategies, based on our design (Section 4) and evaluations (Section 5).

Unreliable UDP-based channel:. ViLTE uses RTP over UDP for media transmission, lacking reliability mechanisms like error correction, retransmission, or congestion control. This makes ViNET's data channel prone to packet loss, jitter, and out-of-order delivery, particularly in congested or unstable networks, as shown in Table 4 and Figure 9).

Unlike Dolphin, that uses TCP-style reliability layer for the voice channels, ViNET relies on underlying protocols (*e.g.*, TCP for HTTP traffic), causing variability and delays due to TCP retransmissions over unreliable UDP channels. This limits ViNET's suitability for latency-sensitive or data-intensive applications like real-time messaging or large file transfers. Future work could develop a custom reliability protocol within ViLTE payloads, using lightweight forward error correction and selective retransmission tailored to ViLTE's QoS constraints. However, implementing this without modifying ViLTE's protocol stack presents significant engineering challenges, requiring extensive carrier network testing.

MNO-induced ViLTE channel shutdown:. ViNET's reliance on ViLTE channels, operating within the IP Multimedia Subsystem (IMS) controlled by MNOs, introduces a critical vulnerability during extreme Internet shutdowns, wherein MNOs may disable ViLTE entirely. In regions like India, where Internet shutdowns are frequent for political control or security, MNOs can selectively suspend IMS services to disrupt communication, as seen during protests and elections.

Unlike circuit-switched voice calls, which often remain operational to support emergency services, ViLTE's packet-switched nature makes it a prime target for shutdowns, halting ViNET's ability to transmit data. In our experience, we did not encounter such scenarios, but ViLTE's availability may be MNO-dependent, with only some carriers prioritising voice over video services during restrictions. This could limit ViNET's resilience in highly censored environments.

Baseband-level ViLTE call processing:. ViNET's packet interception mechanism, implemented using the Netfilter framework on the `rmnet_data` interface (Section 4), relies on ViLTE packets being accessible at the Android OS level. However, in certain device-MNO configurations, ViLTE calls could be processed entirely at the baseband level within the modem, bypassing the OS and preventing the Tunnel application from capturing packets (*i.e.*, much like VoLTE).

Unlike VoLTE, ViLTE's accessibility on `rmnet_data` requires only rooting, but baseband processing could undermine this advantage in affected setups. When packets are directly processed by the modem, ViNET cannot extract or modify payloads, rendering the system ineffective, a critical limitation for implementing ViNET. This issue might be exacerbated by the lack of standardised modem APIs. Potential mitigations include developing low-level interception techniques, such as custom firmware patches, though these are impractical due to restricted access, security risks, and potential IPR violations.

MNO-specific performance variability:. ViNET's performance varies significantly due to MNO-specific ViLTE architectures, impacting throughput, latency, and encapsulation efficiency, as shown through our experiments with three MNOs. Differences in MTU, MSS, traffic shaping, and congestion policies lead to distinct outcomes (Table 4 and figs. 7 and 8). These variations, driven by carrier-specific RTP implementations and network conditions, necessitate per-MNO tuning, as seen in our empirical packet size adjustments (Section 4.2). This requirement reduces ViNET's portability, complicating deployment in diverse regions where MNO configurations differ, such as rural versus urban networks. The lack of standardised ViLTE handling across MNOs may exacerbate such issues. Automated configuration tools could dynamically adjust ViNET's parameters (*e.g.*, packet sizes, retry intervals) based on real-time MNO profiling, improving adaptability. However, developing such tools requires extensive testing across multiple carriers, a resource-intensive process. Thus, enhancing portability remains a critical challenge for ViNET's scalability.

Generalization of functionality/results:. ViNET was tested in three of the most pervasive MNOs of India. As per Telecom Regulator Autority of India (TRAI) August 2025 report [111], there are almost a billion mobile network subscribers in India, that accounts for about 12.5% of the global population. Therefore, we can confidently claim that ViNET was tested on networks serving a significant population. While we don't have empirical evidence to show that ViNET shall work in other countries, we speculate it would, since it follows the specification of ViLTE.

6.7. Experimental Challenges

We now present the challenges that we faced during the experimental evaluation of ViNET.

Ensuring sufficient throughput:. As discussed in Section 4.2.5, the Maximum Transmission Unit (MTU) on the client device must be configured appropriately based on the MNO. Incorrect MTU settings can render ViNET inoperable. Specifically, if the client or server generates packets larger than those typically observed in ViLTE streams, data embedding becomes infeasible (Section 4.1). This mismatch prevents the seamless flow of ViNET data between the client and peer devices, effectively disabling the channel.

DNS resolution challenges:. DNS queries generated by the ViNET client (Alice's machine) are directed to the Android WiFi hotspot, and tunnelled to the peer's device, using ViNET. However, since the peer is unaware of the client's local resolver's IP, the encapsulated packets are dropped, thereby failing resolution. As a workaround, we chose public DNS resolver IPs (*e.g.* Google DNS resolver), which, when decapsulated and decrypted at the peer, can be handled like regular DNS queries destined to such open resolvers. Some web clients (e.g., curl) may fail to resolve queries due to ViLTE packet drops or timeouts from long delays. As DNS is critical for web browsing, we use TCP-based DNS to ensure reliable query response delivery.

Impact of environmental factors:. We observed variations in system performance based on environmental factors like dimensions and position of the room where the phones were placed. These could potentially impact factors like signal strength, interference, network errors *etc.*, impacting ViLTE call throughput. To maintain experimental consistency, we conducted our evaluation in a location where network conditions across all three MNOs were relatively stable and favourable.

Static v/s dynamic video input:. Our experiments revealed that system performance degrades when the smartphone cameras capture static scenes. ViLTE appears to allocate lower bandwidth under such conditions, reducing the available capacity for embedding data. To maintain optimal throughput, we ensured that some degree of motion or activity was consistently present in the video feed during all experiments, *e.g.*, point the camera at ceiling fans, or a screen playing high-framerate videos.

6.8. Ethical Considerations

This research involved the design and implementation of ViNET, a novel system aimed at resisting Internet shutdowns by leveraging Video over LTE (ViLTE). Internet censorship circumvention tools like ViNET may face legal scrutiny (e.g., VPN restrictions in India, [112]). However, the UN recognizes internet access as a human right, and India's Constitution (Articles 19(1)(a), 19(1)(g)) supports this [113]. ViNET aids citizens facing shutdowns, aligning with these rights. ViNET does not over-utilise the ViLTE channel by using more resources than allocated, as shown in Table 3. Though it fully utilises the baseline capacities (Table 2), but never exceeds them. Thus, it does not cause any adverse effects on other users or the ViLTE network infrastructure. Furthermore, ViNET's throughput is not high enough to strain modern Internet infrastructure, but still, responsible experimentation was prioritised. To minimize any unintended impact on network infrastructure, all experiments were carefully scheduled and distributed over several days. This approach was specifically adopted to reduce potential network congestion and avoid any adverse effects on other users of the shared infrastructure. Finally, no human subjects were involved at any stage of this study, thereby exempting it from the need for human-subjects ethical review.

7. Open Science Expectation and Artifact Availability

In accordance with the Euro S&P Open Science expectations, we are making the source codes and relevant research artifacts available through a long-term publicly accessible repository at https://github.com/MasterChief02/ViNET.git.

8. Conclusion

Internet shutdowns, wielded by authoritarian regimes, sever a lifeline as critical as water or electricity, choking communication, cloaking truth and democratic will into silence. Established tools like VPNs and Tor crumble without live Internet connections, and past attempts like Dolphin, though clever, delivered scant bandwidth with no shield for privacy. ViNET, a deft circumvention tool, forged from Video over LTE (ViLTE) calls, which are often unhindered during shutdowns, involves embedded network traffic inside video packets (RTP), clocks upwards of 50 all the way to 400 Kbps. These speeds support basic web-browsing, using text-mode browsers (*e.g.* `lynx`), while supporting end-to-end confidentiality and message integrity. Furthermore, the crafty data embedding inside LTE packets makes spotting harder, even for adversaries employing ML-based traffic classifiers. Additional encryption makes spotting TLS handshake messages embedded in ViLTE payloads difficult at scale.

Yet, ViNET's tether to ViLTE's survival is its Achilles' heel–should regimes snuff out these calls, ViNET shall stumble. Still, ViNET strives forward: a fresh dodge around suppression, a leap past prior limits, and a covert resilience that defies detection.

References

[1] "United nations general assembly, human rights council thirty-second session, third item," https://www.article19.org/data/files/Internet_Statement_Adopted.pdf.

[2] Z. Faiaz, "Right to access the internet is a human right," https://www.thedailystar.net/tech-startup/news/right-access-the-internet-human-right-3659561.

[3] C. Howell and D. M. West, "The internet as a human right." https://www.brookings.edu/articles/the-internet-as-a-human-right/.

[4] S. Correspondent, "Access to internet is a basic right, says kerala high court," https://www.thehindu.com/sci-tech/technology/internet/access-to-internet-is-a-basic-right-says-kerala-high-court/article29462339.ece, 2019.

[5] C. Woodard, "Estonia, where being wired is a human right," https://www.csmonitor.com/2003/0701/p07s01-woeu.html, 2003.

[6] A. Now, "Shutdown tracker optimization project (stop)," https://www.accessnow.org/guide/shutdown-tracker-optimization-project/.

[7] AccessNow, "Accessnow report on internet shutdowns, accessnow, 2019," https://www.accessnow.org/wp-content/uploads/2020/02/KeepItOn-2019-report-1.pdf.

[8] S. F. L. C. India, "Protest shut," https://internetshutdowns.in/static-page/caa-protest/#:~:text=anti%2DCAA%20protests-,Citizenship%20Amendment%20Act%2C%202019%20in%20India%20led%20to%20wide%2Dscale,19th%20December%202019%20as%20well.

[9] A. Deep, "India records highest number of internet shutdowns globally in 2023." https://www.thehindu.com/sci-tech/technology/india-records-highest-number-of-internet-shutdowns-globally-in-2023/article68178061.ece.

[10] I. Today, "Poonch attack: Mobile internet suspended as operation on to hunt down terrorists," https://www.indiatoday.in/india/story/jammu-and-kashmir-poonch-attack-mobile-internet-shut-rajouri-operation-hunt-down-terrorists-2479505-2023-12-23.

[11] H. Correspondent, "Most internet shutdowns in last 3 years to curb protests: Report," https://www.hindustantimes.com/india-news/18-indian-states-shut-down-internet-at-least-once-in-3-years-iff-and-hrw-report-on-internet-shutdowns-in-india-101686769925670.html.

[12] AccessNow, "Tell mena authorities: #noexamshutdown," https://www.accessnow.org/campaign/no-exam-shutdown/.

[13] T. of India News Desk, "Jharkhand suspends mobile internet to check malpractices during exam: All you need to know." https://timesofindia.indiatimes.com/india/jharkhand-suspends-mobile-internet-to-check-malpractice-during-exam-all-you-need-to-know/articleshow/113539597.cms.

[14] NDTV, "Temporary internet shutdown in assam tomorrow to stop cheating in recruitment exam," https://www.ndtv.com/india-news/temporary-internet-shutdown-in-assam-tomorrow-to-stop-cheating-in-recruitment-exam-6566135.

[15] S. Migliano, "Government internet shutdowns cost $7.69 billion in 2024," https://www.top10vpn.com/research/cost-of-internet-shutdowns/?ref=finshots.in.

[16] A. Ganesan, "Post anuradha bhasin, how does restricted internet impact the right to education in jammu and kashmir?" https://www.medianama.com/2022/11/223-jammu-kashmir-internet-shutdown-petition-supreme-court/.

[17] H. R. Watch, "No internet means no work, no pay, no food," https://www.hrw.org/report/2023/06/14/no-internet-means-no-work-no-pay-no-food/internet-shutdowns-deny-access-basic.

[18] R. Dingledine, N. Mathewson, P. F. Syverson *et al.*, "Tor: The second-generation onion router." in *USENIX security symposium*, vol. 4, 2004, pp. 303–320.

[19] D. Fifield, C. Lan, R. Hynes, P. Wegmann, and V. Paxson, "Blocking-resistant communication through domain fronting," *Proceedings on Privacy Enhancing Technologies*, vol. 2015, no. 2, pp. 46–64, 2015.

[20] M. Wei, "Domain shadowing: Leveraging content delivery networks for robust Blocking-Resistant communications," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, 2021, pp. 3327–3343.

[21] J. Karlin, D. Ellard, A. W. Jackson, C. E. Jones, G. Lauer, D. P. Mankins, and W. T. Strayer, "Decoy routing: Toward unblockable internet communication," in *USENIX Workshop on Free and Open Communications on the Internet (FOCI 11)*, 2011.

[22] P. K. Sharma, D. Gosain, H. Sagar, C. Kumar, A. Dogra, V. Naik, H. Acharya, and S. Chakravarty, "Siegebreaker: An sdn based practical decoy routing system," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, no. 3, pp. 243–263, 2020.

[23] E. Wustrow, C. M. Swanson, and J. A. Halderman, "Tapdance: End-to-middle anticensorship without flow blocking," in *Proceedings of 23rd USENIX Security Symposium (USENIX Security 14)*. San Diego, CA: USENIX Association, August 2014.

[24] M. Nasr, H. Zolfaghari, and A. Houmansadr, "The waterfall of liberty: Decoy routing circumvention that resists routing attacks," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 2037–2052.

[25] Briar, "Briar:secure messaging anywhere," https://briarproject.org/.

[26] Starlink, "Starlink: High-speed interent around the world," https://www.starlink.com/.

[27] "Dolphin code," https://github.com/pi-yush/Dolphin-code.

[28] SLFC, "Internet shutdown tracker in India – latest news," https://internetshutdowns.in/news/.

[29] Wikipedia, "Telecommunications in India," https://en.wikipedia.org/wiki/Telecommunications_in_India.

[30] A. Ghayas, "Mobile data speed with 2G, 3G, 4G and 5G cellular networks; Commsbrief — commsbrief.com," https://commsbrief.com/mobile-data-speed--with-2g-3g-4g-and-5g-cellular-networks, 2020.

[31] M. Index, "The Speed of the different Mobile Networks (2G, 3G, 4G, 5G and 6G). What are the speeds? Why is this important or your Mobile Brand / MVNO? — mvno-index.com," https://mvno-index.com/the-speed-of-the-different-mobile-networks/.

[32] AccessNow, "Shutdown tracker optimization project (stop) dashboard v.1," https://www.accessnow.org/keepiton-data-dashboard/.

[33] ——, "Accessnow report on internet shutdowns," https://www.accessnow.org/wp-content/uploads/2025/02/KeepItOn-2024-Internet-Shutdowns-Annual-Report.pdf, 2025.

[34] ——, "#keepiton stop data 2016-2024," https://docs.google.com/spreadsheets/d/1DvPAuHNLp5BXGb0nnZDGNoiIwEeu2ogdXEIDvT4Hyfk/edit?gid=1225026792#gid=1225026792, 2025.

[35] SLFC, "Internet shutdown tracker in India," https://internetshutdowns.in.

[36] I. Society, "Policy brief: Internet shutdowns, Internet Society, December 2019," https://www.internetsociety.org/policybriefs/internet-shutdowns.

[37] I. Outage Detection and Analysis, "Internet Outage Detection and Analysis (IODA) Project," https://ioda.caida.org/.

[38] Cisco, "Internet Outages Map," https://www.thousandeyes.com/outages/.

[39] GSMA, "ViLTE (Video over LTE)," https://www.gsma.com/solutions-and-impact/technologies/networks/ip_services/vilte/.

[40] 3rd Generation Partnership Project, "3gpp ts 26.114," September 2023. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1404

[41] D. Rupprecht, K. Kohls, T. Holz, and C. Pöpper, "Call me maybe: Eavesdropping encrypted LTE calls with ReVoLTE," in *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, 2020, pp. 73–88.

[42] ——, "Breaking lte on layer two," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 1121–1136.

[43] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*. USA: USENIX Association, 2004, p. 21.

[44] S. Frolov and E. Wustrow, "*httpt*: A probe-resistant proxy," in *Proceedings of Workshop on Free and Open Communications on the Internet (FOCI)*, 2020.

[45] P. T. J. Kon, S. Kamali, J. Pei, D. Barradas, A. C. M. Sherr, and M. Yung, "Spotproxy: rediscovering the cloud for censorship circumvention," in *Proceedings of the 33rd USENIX Conference on Security Symposium*, ser. SEC '24. USA: USENIX Association, 2024.

[46] N. P. Hoang, P. Kintis, M. Antonakakis, and M. Polychronakis, "An empirical study of the i2p anonymity network and its censorship resistance," in *Proceedings of the Internet Measurement Conference 2018*. Association for Computing Machinery, 2018, p. 379–392.

[47] E. Wustrow, S. Wolchok, I. Goldberg, and J. A. Halderman, "Telex: Anticensorship in the network infrastructure," in *20th USENIX Security Symposium (USENIX Security 11)*. San Francisco, CA: USENIX Association, 2011.

[48] A. Houmansadr, G. T. Nguyen, M. Caesar, and N. Borisov, "Cirripede: circumvention infrastructure using router redirection with plausible deniability," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, ser. CCS '11, 2011.

[49] S. Frolov, F. Douglas, W. Scott, A. McDonald, B. VanderSloot, R. Hynes, A. Kruger, M. Kallitsis, D. G. Robinson, S. Schultze, N. Borisov, A. Halderman, and E. Wustrow, "An ISP-Scale deployment of TapDance," in *7th USENIX Workshop on Free and Open Communications on the Internet (FOCI 17)*, Vancouver, BC, 2017.

[50] S. Frolov, J. Wampler, S. C. Tan, J. A. Halderman, N. Borisov, and E. Wustrow, "Conjure: Summoning proxies from unused address space," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '19, 2019, p. 2215–2229.

[51] A. Devraj, L. Wang, and J. Rexford, "Redact: refraction networking from the data center," *SIGCOMM Comput. Commun. Rev.*, vol. 51, no. 4, p. 15–22, 2021.

[52] P. Kon, A. Gattani, D. Saharia, T. Cao, D. Barradas, A. Chen, M. Sherr, and B. E. Ujcich, "Netshuffle: Circumventing censorship with shuffle proxies at the edge," in *2024 IEEE Symposium on Security and Privacy (SP)*. Los Alamitos, CA, USA: IEEE Computer Society, 2024, pp. 3497–3514.

[53] H. Mohajeri Moghaddam, B. Li, M. Derakhshani, and I. Goldberg, "Skypemorph: protocol obfuscation for tor bridges," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, ser. CCS '12, 2012, p. 97–108.

[54] K. Kohls, T. Holz, D. Kolossa, and C. Pöpper, "Skypeline: Robust hidden data transmission for voip," in *Proceedings of the ACM Asia Conference on Computer and Communications Security (AsiaCCS)*, 2016, pp. 877–888.

[55] A. Houmansadr, T. J. Riedl, N. Borisov, and A. C. Singer, "I want my voice to be heard: IP over Voice-over-IP for unobservable censorship circumvention." in *Proceedings of the Symposium on Networks and Distributed Systems Security (NDSS)*, 2013.

[56] A. Zarras, "Leveraging Internet services to evade censorship," in *International Conference on Information Security (ICIS)*. Springer, 2016, pp. 253–270.

[57] C. Connolly, P. Lincoln, I. Mason, and V. Yegneswaran, "TRIST: Circumventing Censorship with Transcoding-Resistant Image Steganography," in *Proceedings of Workshop on Free and Open Communications on the Internet (FOCI)*, 2014.

[58] R. McPherson, A. Houmansadr, and V. Shmatikov, "Covertcast," *Proceedings on Privacy Enhancing Technologies' Symposium (PoPETS)*, vol. 3, pp. 1–14, 2016.

[59] D. Barradas, N. Santos, and L. E. Rodrigues, "Deltashaper: Enabling unobservable censorship-resistant tcp tunneling over videoconferencing streams." *Proceedings of Privacy Enhancing Technologies' Symposium (PoPETS)*, vol. 2017, no. 4, pp. 5–22, 2017.

[60] S. F. L. Centre, "Internet shutdown tracker," 2024. [Online]. Available: https://internetshutdowns.in/

[61] AccessNow, "2023 shutdown report," March 2024. [Online]. Available: https://www.accessnow.org/wp-content/uploads/2024/03/2023-Shutdown-Report.pdf

[62] S. C. of India, "Anuradha bhasin v. union of india," January 2020. [Online]. Available: https://indiankanoon.org/doc/101953971/

[63] T. I. Express, "Mobile internet ban in nine manipur districts extended till december 9," *The Indian Express*, December 2024. [Online]. Available: https://www.thehindu.com/news/national/manipur/mobile-internet-ban-in-nine-manipur-districts-extended-till-december-9/article68959697.ece

[64] C. K. LaDue, V. V. Sapozhnykov, and K. S. Fienberg, "A data modem for GSM voice channel," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 4, pp. 2205–2218, 2008.

[65] M. Perić, P. Milićević, Z. Banjac, and B. M. Todorović, "An experiment with real-time data transmission over global scale mobile voice channel," in *2015 12th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS)*. IEEE, 2015, pp. 239–242.

[66] B. T. Ali, G. Baudoin, and O. Venard, "Data transmission over mobile voice channel based on m-fsk modulation," in *IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2013, pp. 4416–4421.

[67] M. A. Özkan and S. B. Örs, "Data transmission via GSM voice channel for end to end security," in *2015 IEEE 5th International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*. IEEE, 2015, pp. 378–382.

[68] R. Kazemi, M. Boloursaz, S. M. Etemadi, and F. Behnia, "Capacity bounds and detection schemes for data over voice," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 11, pp. 8964–8977, 2016.

[69] A. Dhananjay, A. Sharma, M. Paik, J. Chen, T. K. Kuppusamy, J. Li, and L. Subramanian, "Hermes: Data transmission over unknown voice channels," in *Proceedings of the ACM International Conference on Mobile Computing and Networking*, 2010, pp. 113–124.

[70] T. Ahmad, E. Reed-Sanchez, F. Zarinni, A. Afutu, K. Adjaho, Y. Nyarko, and L. Subramanian, "Greenapps: A platform for cellular edge applications," in *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, 2018, pp. 1–5.

[71] F. R. Dogar, I. A. Qazi, A. R. Tariq, G. Murtaza, A. Ahmad, and N. Stocking, "Missit: Using missed calls for free, extremely low bit-rate communication in developing regions," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 2020, pp. 1–12.

[72] H. Kim, D. Kim, M. Kwon, H. Han, Y. Jang, D. Han, T. Kim, and Y. Kim, "Breaking and Fixing VoLTE: Exploiting hidden data channels and mis-implementations," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2015, pp. 328–339.

[73] C.-Y. Li, G.-H. Tu, C. Peng, Z. Yuan, Y. Li, S. Lu, and X. Wang, "Insecurity of voice solution VoLTE in LTE mobile networks," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2015, pp. 316–327.

[74] Y. Wang, W. Han, L. Wang, L. Liu, and X. Yu, "Encrypting volte communication via baseband firmware extension," in *Cyberspace Safety and Security*, J. Vaidya, X. Zhang, and J. Li, Eds. Cham: Springer International Publishing, 2019, pp. 45–56.

[75] T. E. Centre, "Study paper on voice over lte: New voice dynamics," https://tec.gov.in/public/pdf/Studypaper/Study%20Paper%20on%20VoLTE%20-%20Final%20(1).pdf, 2017.

[76] S. Kamali and D. Barradas, "Anix: Anonymous blackout-resistant microblogging with message endorsing," in *2025 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2025, pp. 1381–1399.

[77] A. Pradeep, H. Javaid, R. Williams, A. Rault, D. Choffnes, S. Le Blond, and B. Ford, "Moby: A Blackout-Resistant Anonymity Network for Mobile Devices," *Proceedings on Privacy Enhancing Technologies*, vol. 3, pp. 247–267, 2022.

[78] A. Lerner, G. Fanti, Y. Ben-David, J. Garcia, P. Schmitt, and B. Raghavan, "Rangzen: Anonymously getting the word out in a blackout," https://arxiv.org/abs/1612.03371, 2016.

[79] P. Shadbolt, "Firechat in hong kong: How an app tapped its way into the protests," *CNN, October*, vol. 16, 2014.

[80] Bridgefy, "Bridgefy messaging app," https://bridgefy.me/.

[81] A. Bienstock, P. Rösler, and Y. Tang, "Asmesh: Anonymous and secure messaging in mesh networks using stronger, anonymous double ratchet," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 1–15.

[82] Z. Doffman, "Russian Authorities "Secretly" Shutdown Moscow's Mobile Internet: Report, Forbes, August 2019," https://www.forbes.com/sites/zakdoffman/2019/08/08/russian-security-agencies-secretly-shut-moscows-mobile-internet-to-control-protestors-report/?sh=3137df6c75bf.

[83] NDTV, "Mobile internet services suspended in 13 districts in Haryana," https://www.ndtv.com/chandigarh-news/mobile-internet-services-suspended-in-13-districts-in-haryana-1779748.

[84] B. BusinessWorld, "Internet shutdown in response to mega public gathering in India, business world, october 2020," https://tinyurl.com/yxa9e32u.

[85] B. G. Dastidar, "Reimagining india's national telemedicine service to improve access to care," https://www.thelancet.com/journals/lansea/article/PIIS2772-3682(24)00130-6/fulltext.

[86] V. H. Nagaraja, B. Ghosh Dastidar, S. Suri, and A. R. Jani, "Perspectives and use of telemedicine by doctors in india: A cross-sectional study," *Health Policy and Technology*, vol. 13, no. 2, p. 100845, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221188372400008X

[87] Ericsson, "Voice and communication services in 4g and 5g networks," Ericsson, White paper, 2022. [Online]. Available: https://www.ericsson.com/4a19c3/assets/local/reports-papers/white-papers/voice-and-communication-services.pdf

[88] I. S. Pulse, "Internet shutdowns," https://pulse.internetsociety.org/en/shutdowns/.

[89] U. P. Shasan, "Uttar pradesh shasan, gopan anubhag-3, no. 90/2025-cx-3," https://internetshutdowns.in/media/Barelly_2809_8fe4f68d-c5ba-4d29-8cba-d08ce331d031..pdf.

[90] A. Buzz, "Internet suspended, curfew imposed: Maharashtra authorities respond to maratha quota agitation." https://aurangabadbuzz.com/2024/02/25/internet-suspended-curfew-imposed-maharashtra-authorities-respond-to-maratha-quota-agitation/.

[91] Research and Markets, "Myanmar telecom operators country intelligence report," https://www.researchandmarkets.com/reports/5134523/myanmar-telecom-operators-country-intelligence.

[92] S. M. Shah, "Pakistan telecom market share & revenue 2025," https://digitalpakistan.pk/pakistan-telecom-market-share-revenue-2025/.

[93] M. Landscapes, "Iran - mobile coverage," https://medialandscapes.org/static/country/iran/telecommunications/mobile-coverage.html.

[94] P. Bell, "Ukraine's telecom market, explained," https://resources.telegeography.com/ukraines-telecom-market-explained.

[95] AT&T, "At&t video call," https://www.att.com/features/video-calling/.

[96] R. Asia, "Ctm, csl launch volte roaming between hong kong and macau," https://www.retailnews.asia/ctm-csl-launch-volte-roaming-hk-macau/.

[97] AccessNow, "#keepiton stop data 2016-2024," https://docs.google.com/spreadsheets/d/1DvPAuHNLp5BXGb0nnZDGNoiIwEeu2ogdXEIDvT4Hyfk, 2025.

[98] Airtel, "Airtel network coverage," https://www.airtel.in/wirelesscoverage/?icid=footer.

[99] Vi, "Vi coverage map," https://www.myvi.in/vicoverage/.

[100] Jio, "5g and 4g coverage map," https://www.jio.com/selfcare/coverage-map/.

[101] Y. Wang, R. Even, T. Kristensen, and R. Jesup, "RTP Payload Format for H.264 Video," RFC 6184, May 2011. [Online]. Available: https://www.rfc-editor.org/info/rfc6184

[102] Google, "Use the NDK with other build systems — Android NDK — Android Developers — developer.android.com," https://developer.android.com/ndk/guides/other_build_systems?utm_source=chatgpt.com, 2024.

[103] "Integrated Government Online Directory — igod.gov.in," https://igod.gov.in/.

[104] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczynski, and W. Joosen, "Tranco: A research-oriented top sites ranking hardened against manipulation," in *Proceedings 2019 Network and Distributed System Security Symposium*. Reston, VA: Internet Society, 2019.

[105] "Lynx – the text web-browser," https://lynx.invisible-island.net/.

[106] D. Barradas, N. Santos, and L. Rodrigues, "Effective detection of multimedia protocol tunneling using machine learning," in *Proceedings of the 27th USENIX Conference on Security Symposium*, ser. SEC'18. USA: USENIX Association, 2018, p. 169–185.

[107] D. Xue, M. Kallitsis, A. Houmansadr, and R. Ensafi, "Fingerprinting obfuscated proxy traffic with encapsulated TLS handshakes," in *33rd USENIX Security Symposium (USENIX Security 24)*. Philadelphia, PA: USENIX Association, 2024, pp. 2689–2706.

[108] D. Fifield, "Threat modeling and circumvention of internet censorship," https://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-225.pdf, 2017.

[109] F. De Gaspari, D. Hitaj, G. Pagnotta, L. De Carli, and L. V. Mancini, "Reliable detection of compressed and encrypted data," *Neural computing & applications*, Jul. 2022.

[110] J. Geddes, M. Schuchard, and N. Hopper, "Cover your ACKs," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security - CCS '13*. New York, New York, USA: ACM Press, 2013.

[111] TRAI, "Trai press release no.118/2025 dated 27.10.2025," https://www.trai.gov.in/sites/default/files/2025-10/PR_No.120of2025_0.pdf.

[112] "Hands off our vpns! #savetheinternet," https://internetfreedom.in/hands-off-our-vpns-savetheinternet/#:~:text=Basically%2C%20the%20committee%20asked%20the,reiterated%20its%20stance%20on%20VPNs.

[113] "Indian journal of law and technology," https://repository.nls.ac.in/cgi/viewcontent.cgi?article=1293&context=ijlt.

[114] GoogleGuide, "Interpreting Your Google Query - Google Guide — googleguide.com," https://www.googleguide.com/interpreting_queries.html, 2022.

Appendix

1. Custom website dataset

Table 8 lists the custom domains used for the webpage load-time tests as described in Section 5.3.2, along with the page sizes and average load times for each MNO.

Domain	Size	Average load time (s)		
		MNO 1	MNO 2	MNO 3
anantnag.nic.in	117 kB	16.84	5.37	16.86
bandipore.nic.in	122 kB	16.58	4.96	18.83
baramulla.nic.in	120 kB	18.06	4.46	18.14
budgam.nic.in	120 kB	16.73	4.87	16.05
doda.nic.in	132 kB	16.40	5.97	18.89
ganderbal.nic.in	135 kB	18.73	5.54	18.51
jammu.nic.in	138 kB	18.60	6.75	20.66
kathua.nic.in	106 kB	14.01	4.77	16.55
kishtwar.nic.in	102 kB	23.39	4.30	14.84
kulgam.nic.in	132 kB	18.61	5.47	19.66
kupwara.nic.in	115 kB	17.06	6.06	17.02
poonch.nic.in	120 kB	17.54	5.01	18.86
pulwama.gov.in	114 kB	16.69	5.08	18.01
rajouri.nic.in	97 kB	13.59	3.65	15.01
ramban.gov.in	108 kB	15.85	5.41	17.26
reasi.nic.in	157 kB	23.91	5.63	23.41
samba.gov.in	180 kB	24.30	7.11	22.96
shopian.nic.in	138 kB	19.55	5.35	18.77
srinagar.nic.in	110 kB	17.38	5.02	16.60
udhampur.nic.in	116 kB	17.15	5.43	15.82

TABLE 8: Websites in custom dataset along with their average load times

2. Tranco filtered dataset

Since the Tranco [104] list does not provide inherent categorisation, we employed large language models (LLMs) to filter the dataset according to our criteria to create our custom dataset. The filtering process involved instructing the LLM to identify URLs belonging to information and news categories while excluding social media platforms, messaging services, search engines, and other irrelevant sites. We used `gemini-2.5-flash-preview-05-20` with Google Search and URL Context enabled for this classification task. Furthermore, we only kept domains that returned a 200 OK response code when accessed via `curl`. Following the filtering process, we selected the top 10 websites from the filtered list, as detailed in Table 9, along with

the page size and average load time for each MNO. The system prompt given to the LLM was

> *You are an data filtering system which will be given a bunch of URLs out of which you have to find out which URLs belong to information and news category. Exclude social media, messengers, search engines, and any other irrelevant URLs. Donot include the URLs which are not related to information and news category. Be as specific as possible.*

Domain	Size	Average load time (s)		
		MNO 1	MNO 2	MNO 3
wikipedia.org	88 kB	11.42	4.86	14.34
nist.gov	87 kB	13.61	5.65	14.41
nih.gov	61 kB	9.30	3.38	15.34
nytimes.com	770 kB	119.10	38.21	100.30
forbes.com	571 kB	72.61	20.99	71.70
w3.org	48 kB	8.09	2.88	12.47
bbc.com	509 kB	61.96	18.91	66.52
bbc.co.uk	655 kB	74.31	29.43	89.86
mit.edu	46 kB	7.17	3.66	8.43
wikimedia.org	45 kB	7.51	2.74	15.34

TABLE 9: Websites in the Tranco dataset along with their average load times

3. Google Search

We constructed a query with the maximum length permitted by Google Search [114]. This query was retrieved multiple times using Google Search, as detailed in our experiment. The formulated query was:

> *How to effectively use mindfulness meditation techniques to reduce stress and anxiety, improve focus, enhance productivity, and maintain mental well-being during challenging work or study periods while managing distractions?*