Out in the Open:
On the Implementation of Mobile App Filtering in India

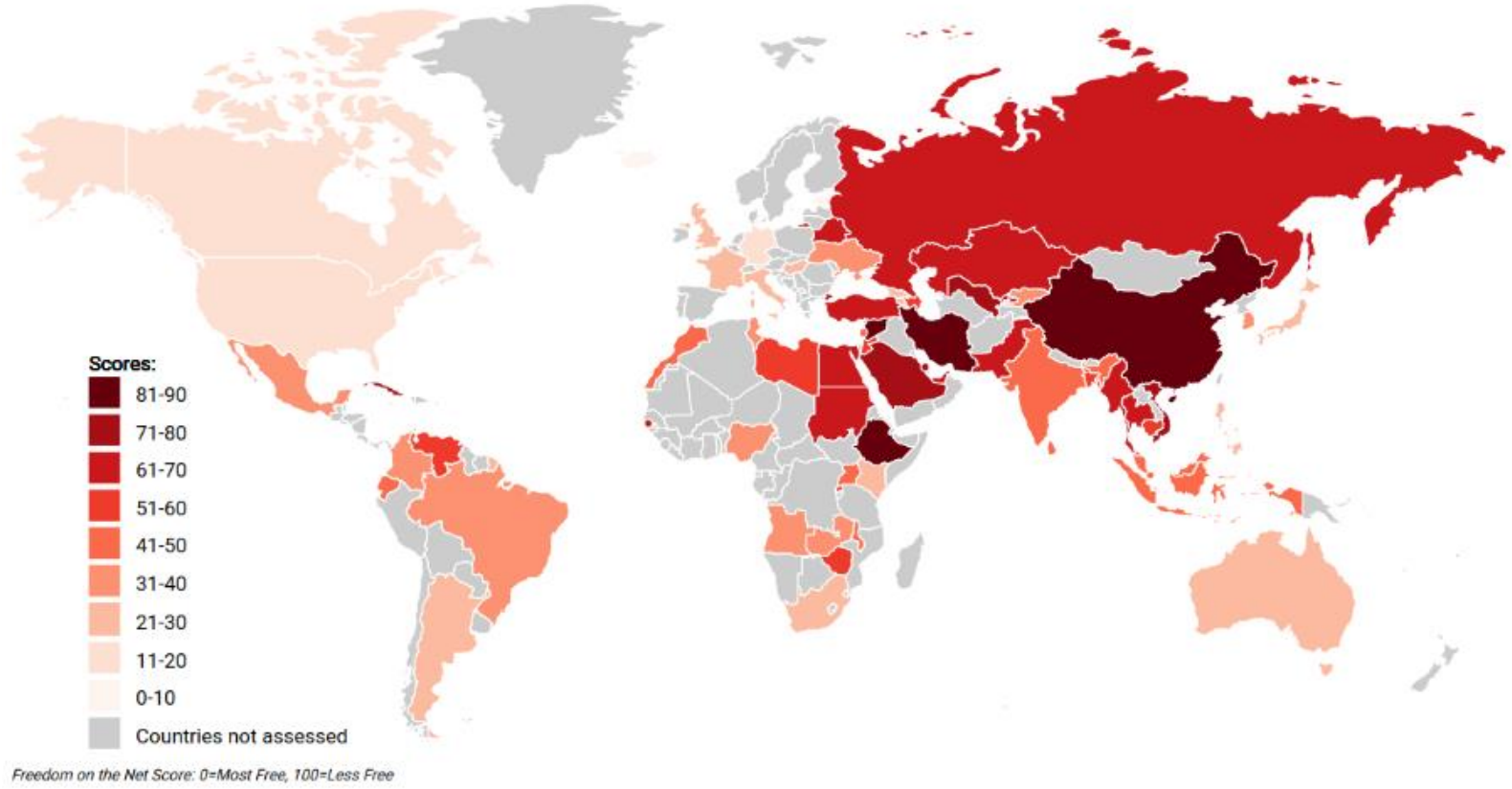
Devashish Gosain, Kartikey Singh, Rishi Sharma, Jithin S, Sambuddho

max planck institut informatik

BITS Pilani
K K Birla Goa Campus

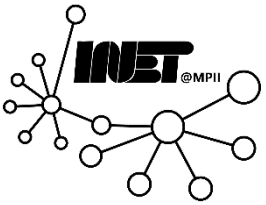
Internet Filtering

Control of what information can be accessed, published, or viewed on the Internet enacted by regulators.



Internet Filtering on Rise

- Internet shutdowns, banning of apps (e.g., TikTok) are on rise
- EU countries have started censorship (e.g., Greece [Ververis et al. 2015], Spain [Ververis et al., 2021])
  - Sweden and France soon to join the league
- Russia --- country legally passed the bill regarding Internet Censorship
- China [Xu et al., 2011], Iran [Aryan et al., 2013], Pakistan [Nabi et al., 2013] etc. --- *the usual suspects*

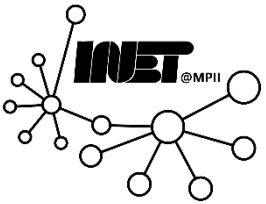


India and Traffic Filtering

- India is the second largest Internet userbase (after China), and much less studied

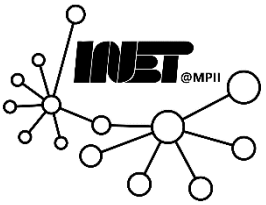
- Previous studies [Gosain et al. 2017, Yadav et al. 2018] highlight
  - "feudal model" of **web censorship** within India
  - Filtering infrastructure used by different ISPs

- But, in 2020 we observed an emergence of new form of blocking—Mobile App Filtering
  During June--September 2020, India officially banned 220 Chinese apps



Our Objectives

- To study this new form of Internet blocking
  - Mobile app filtering
  - Describe in detail the mechanics involved

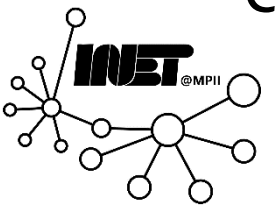


Initial observations

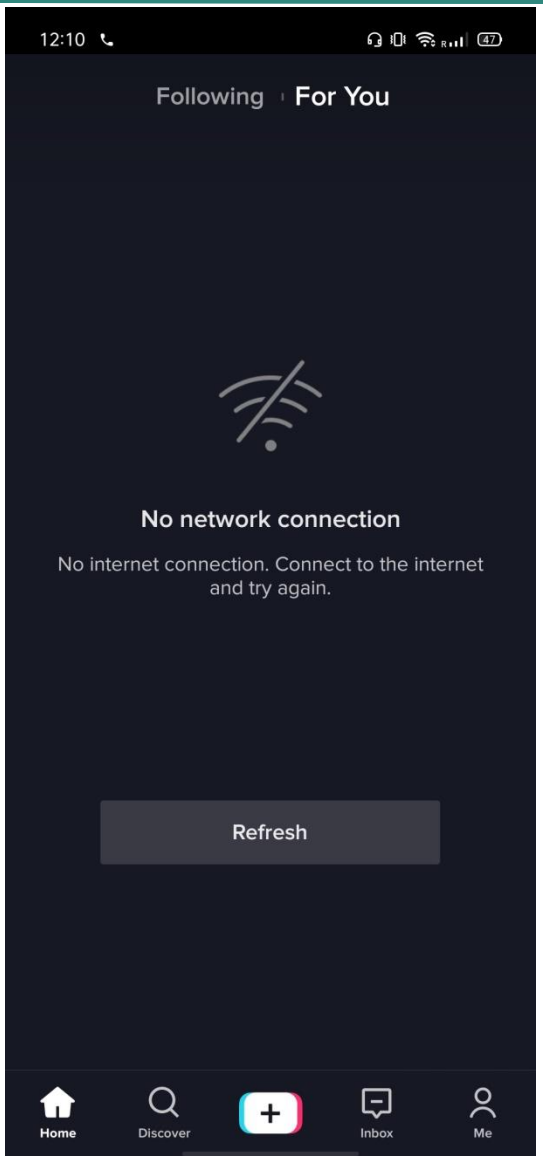
- After the ban, all the blocked apps were unavailable in the official Indian app stores (e.g., Google Playstore)

- Obtained the *.apk files* of 160 banned apps
  - from third-party sources (e.g., foreign app stores and websites like *apkmirror.com*)
  - 136 were directly accessible from IN

- But, accessing the remaining 24 apps resulted in network connection errors

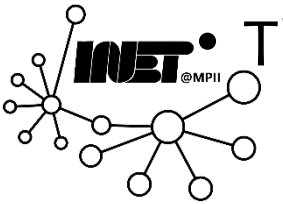


Initial observations



Investigating ISP Level Filtering

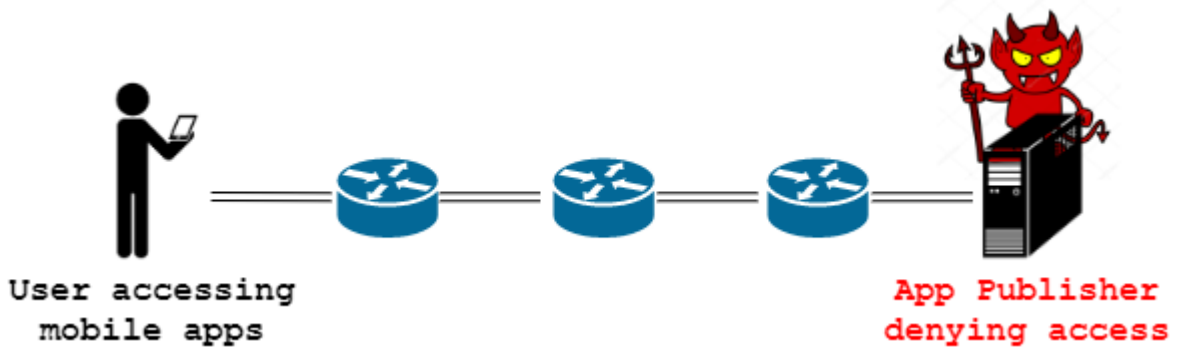
- Using *pcaps* we observed
  - Legitimate IP addresses (Akamai)
    → **No DNS Filtering**
  - Successful TCP (and TLS handshake)
    → **No TCP/IP Filtering**
  - Legitimate certificate (of TikTok.com)
    → **No Certificate Tempering/MITM**
  - Encrypted data exchange
- The blocking message was received from the app server itself!



Investigating ISP Level Filtering

This ruled out the involvement of Indian ISPs in App filtering

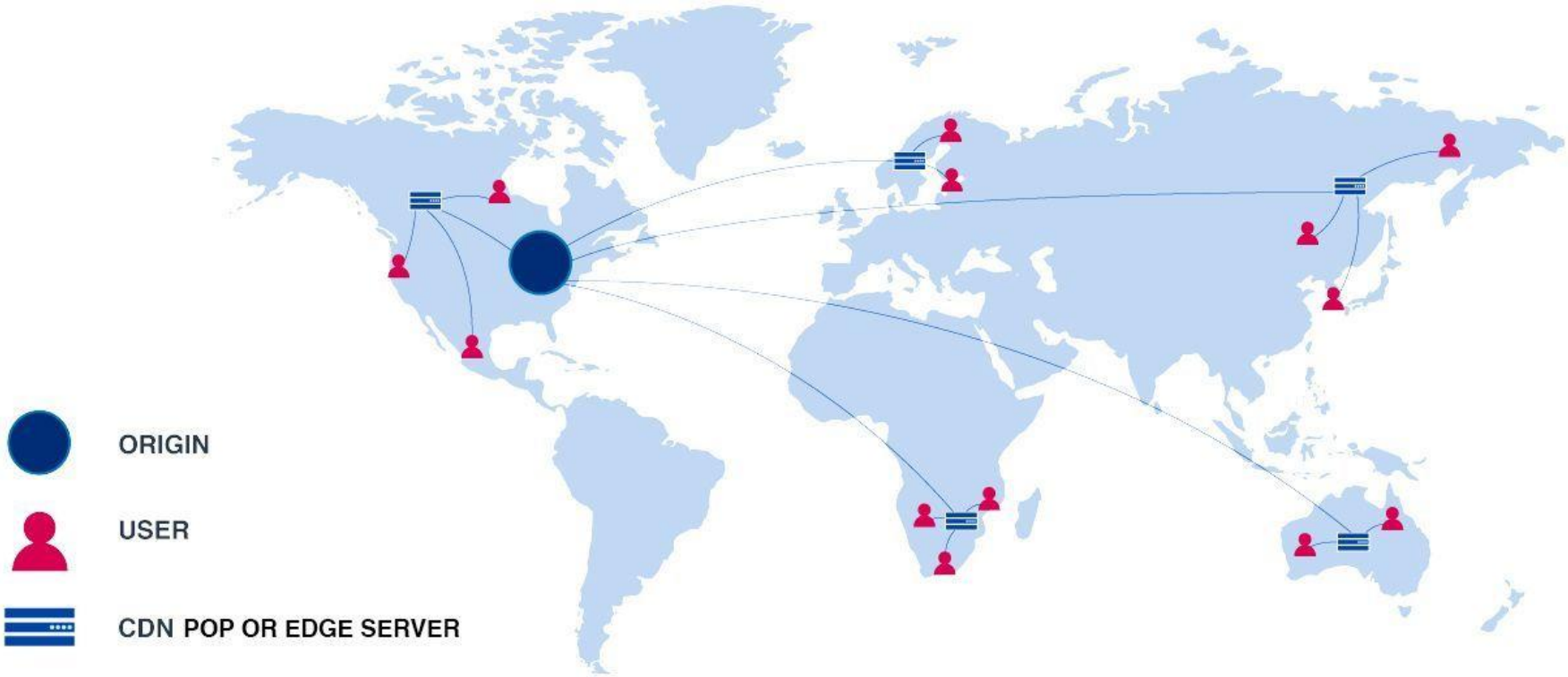
How App Publishers Restrict Access?



- **Hypothesis:** App publishers might **geo-block** clients
  - Based on source IP address
  - Restricting content on CDN edge-servers (serving Indian clients)

Content Distribution Network (CDNs)

- Popular web content rely on Content Distribution Networks (CDNs)
  - Same content is replicated a multiple geographic locations (**Front-ends**)



A simple solution

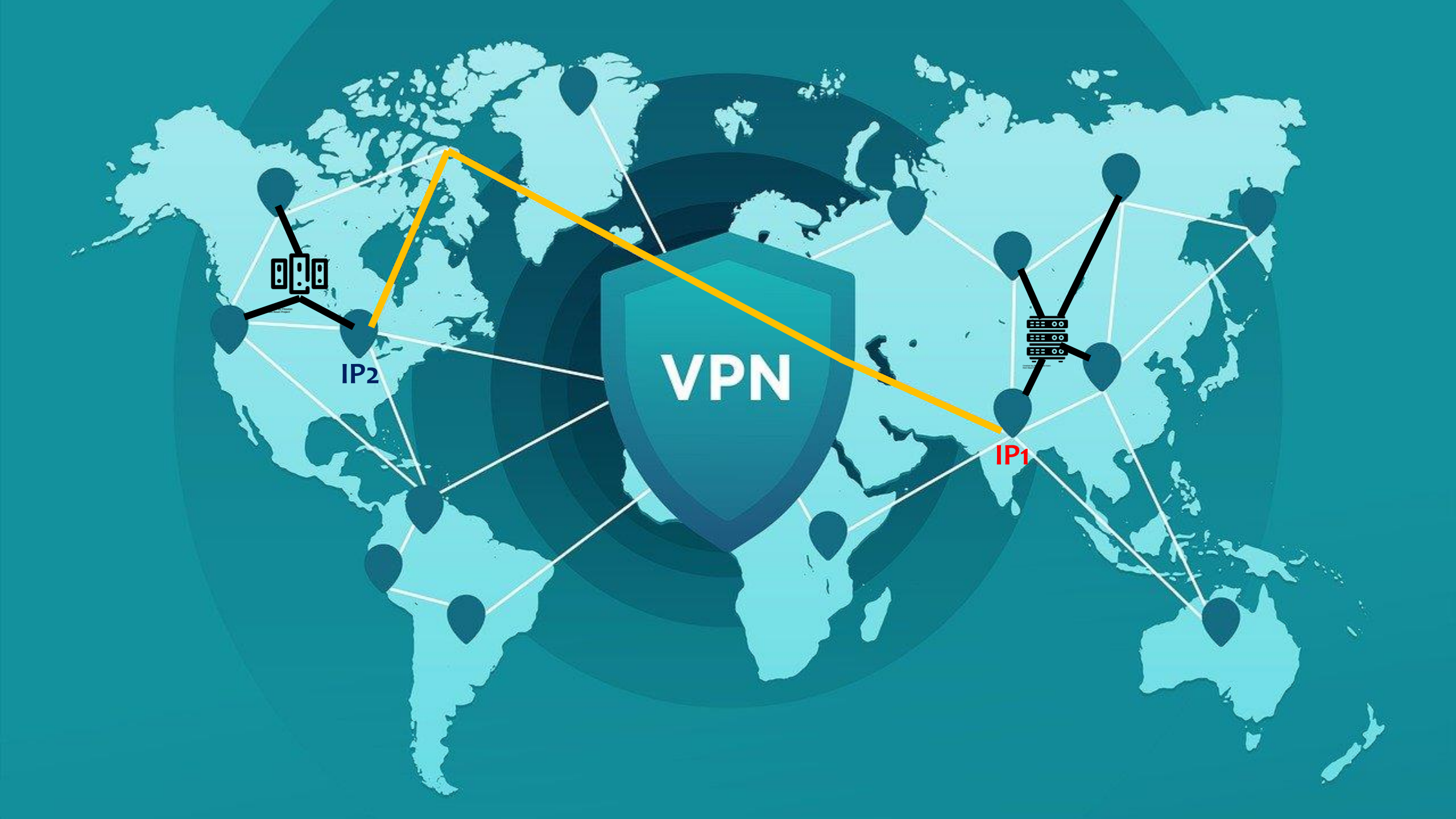
- Use VPNs!

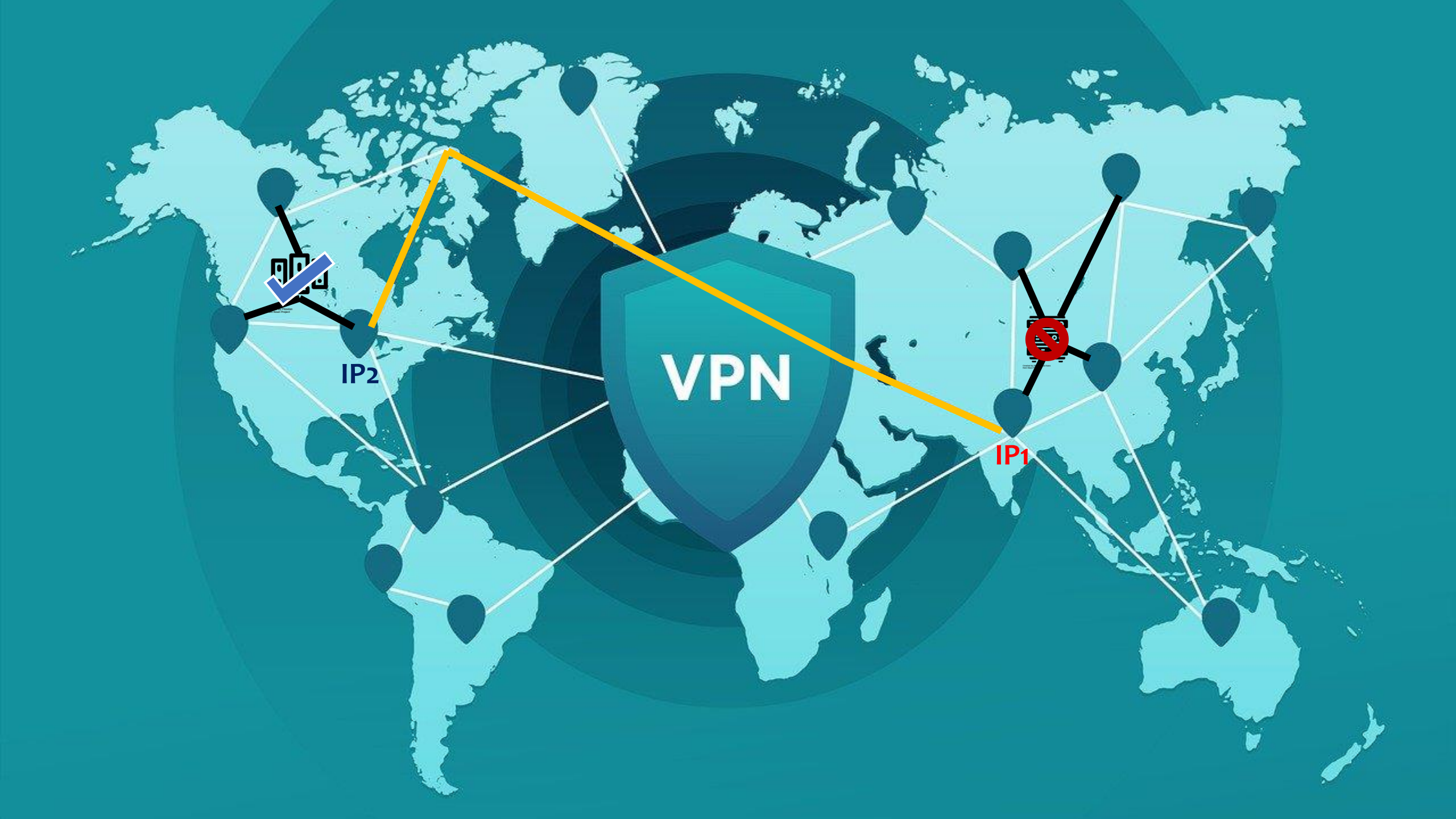
- But using VPNs would change both
  - Src. IP
  - CDN edge servers


VPN


VPN


VPN
IP2
IP1


VPN
IP2
IP1


VPN
IP2
IP1

Two variables four cases

1. Indian Source IP and Indian edge-server (default case)
2. Indian Source IP and Foreign edge-server
3. Foreign Source IP and Indian edge-server
4. Foreign Source IP and Foreign edge-server

Two variables four cases

1. Indian Source IP and Indian edge-server (default case)
2. Indian Source IP and Foreign edge-server 
3. Foreign Source IP and Indian edge-server 
4. Foreign Source IP and Foreign edge-server

Use of CDNs

- The blocked apps were relying on DNS based CDNs

- DNS based CDNs: Different front-ends (edge-servers) have *different* IP addresses
  - The client obtains the IP address of front-end closer to its **DNS resolver**

Two variables four cases

1. Indian Source IP and Indian edge-server (default case)
   - 24 were inaccessible

Two variables four cases

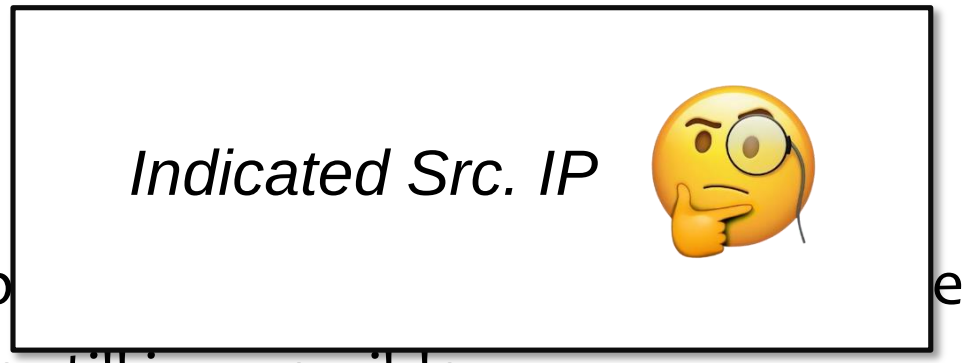
1. Indian Source IP and Indian edge-server (default case)
   - 24 were inaccessible

2. Indian Source IP and Foreign edge-server
   - Configured phone in IN to use open DNS resolver in uncensored countries
   - 24 apps were still inaccessible

Two variables four cases

1. Indian Source IP and Indian edge-server (default case)
   - 24 were inaccessible

2. Indian Source
   - Configured p er in uncensored countries
   - 24 apps were still inaccessible

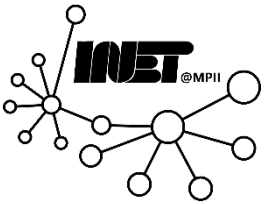


Two variables four cases

1. Indian Source IP and Indian edge-server (default case)
   - 24 were inaccessible

2. Indian Source IP and Foreign edge-server
   - Configured phone in IN to use open DNS resolver in uncensored countries
   - 24 apps were still inaccessible

3. Foreign Source IP and Indian edge-server
   - Configured phone in DE to use DNS resolvers in IN
   - 15/24 apps were accessible
   - 9 were still inaccessible

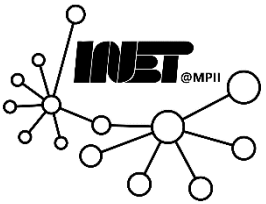


Two variables four cases

1. Indian Source IP and Indian edge-server (default case)
   - 24 were inaccessible

2. Indian Source IP and Foreign edge-server
   - Configured phone in IN to use open DNS resolver in uncensored countries
   - 24 apps were still inaccessible

3. Foreign Source IP and Indian edge-server
   - Configured phone in DE to use DNS resolvers in IN
   - 15/24 apps were accessible
   - 9 were still inaccessible

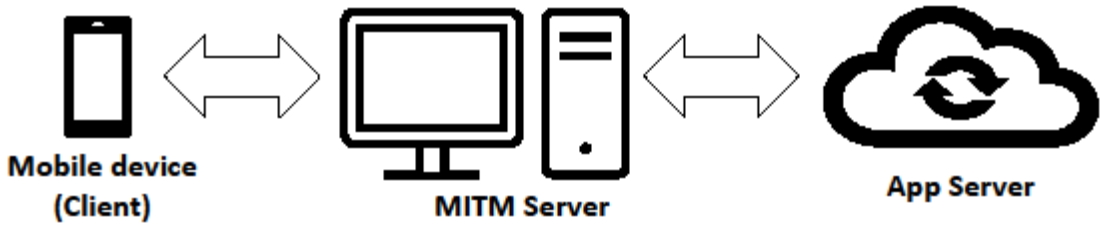


The last case

- Foreign Source IP and Foreign edge-server
  - Ideally all the apps should be accessible??
  - But, to our surprise out of previous 9 apps only 1 was accessible!
- The remaining 8 apps required further investigation
- As a case study we examined TikTok

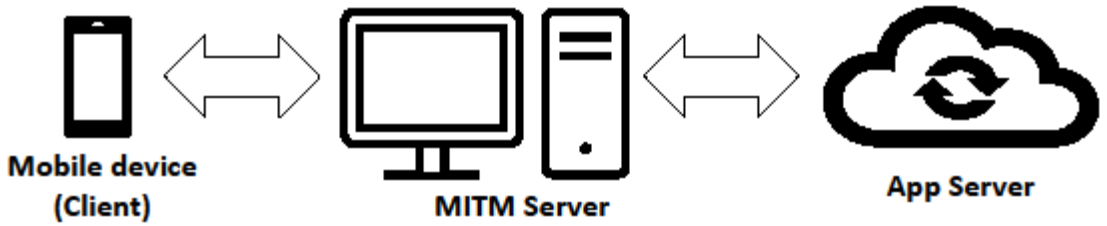
Analyzing the Apps Traffic in Plain-text

- MITM (used *mitmproxy*)



Analyzing the Apps Traffic in Plain-text

- MITM (used *mitmproxy)*



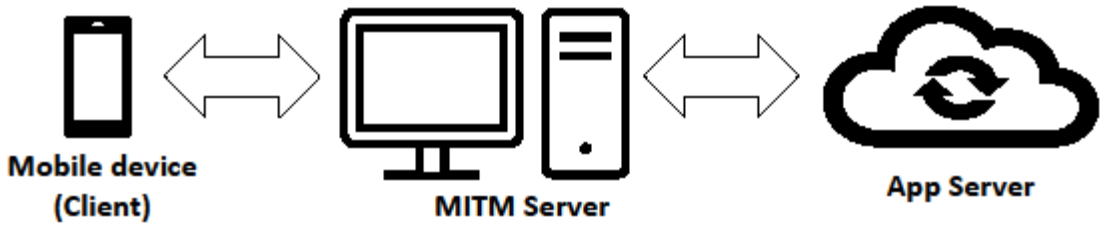

```
carrier_region=IN
```

- Reverse-Engineered the app (using jadx decompiler)

```
Object systemService = getContext().getSystemService("phone");
if (systemService != null) {
    String simCountryIso = ((TelephonyManager) systemService).getSimCountryIso();
    String str = simCountryIso;
    boolean z3 = false;
```

Analyzing the Apps Traffic in Plain-text

- MITM (used *mitmproxy*)




```
carrier_region=IN
```

- Reverse-Engineered the app (using jadx decompiler)

```
Object systemService = getContext().getSystemService("phone");
if (systemService != null) {
    String simCountryIso = ((TelephonyManager) systemService).getSimCountryIso();
    String str = simCountryIso;
    boolean z3 = false;
```

Analyzing the Apps Traffic in Plain-text

- MITM (used *mitmproxy*)

- Only when the Indian SIM Card is installed in the phone
  - TikTok sends `carrier_region=IN` parameter in HTTP requests
  - Else the parameter was absent

```
String simCountryIso = ((TelephonyManager) systemService).getSimCountryIso();
String str = simCountryIso;
boolean z3 = false;
```

Circumvention for the remaining 8 apps

- Change carrier_region=IN to any other country
  e.g., carrier_region = US

- Simply remove the SIM card

App Censorship at a Glance

Sl No.	App Name	App Type	Censorship Technique Used		
			Client Source IP	CDN Edge server	Client SIM Card
1	PUBG	Gaming	✓	✗	✗
2	ShareIt	Tools	✓	✗	✗
3	Shein	Shopping	✓	✗	✗
4	Baidu	Tools	✓	✗	✗
5	Tantan	Social	✓	✗	✗
6	VooV	Productivity	✓	✗	✗
7	RomWe	Shopping	✓	✗	✗
8	Ludo	Gaming	✓	✗	✗
9	Rangers of Oblivion	Gaming	✓	✗	✗
10	Ali Suppliers	Business	✓	✗	✗
11	Baidu Express	Tools	✓	✗	✗
12	DingTalk	Productivity	✓	✗	✗
13	MangoTV	Video Players	✓	✗	✗
14	Heroes Evolved	Gaming	✓	✗	✗
15	Singol	Dating	✓	✗	✗
16	ChessRush	Gaming	✓	✓	✗

Sl No.	App Name	App Type	Censorship Technique Used		
			Client Source IP	CDN Edge server	Client SIM Card
17	TikTok	Social	✗	✗	✓
18	Likee	Video Players	✗	✗	✓
19	Kwai	Social	✗	✗	✓
20	UC Browser	Browser	✗	✗	✓
21	FaceU	Photography	✗	✗	✓
22	Hago	Social	✗	✗	✓
23	V-Fly	Tools	✗	✗	✓
24	MICO Chat	Social	✗	✗	✓

Its Not The End !

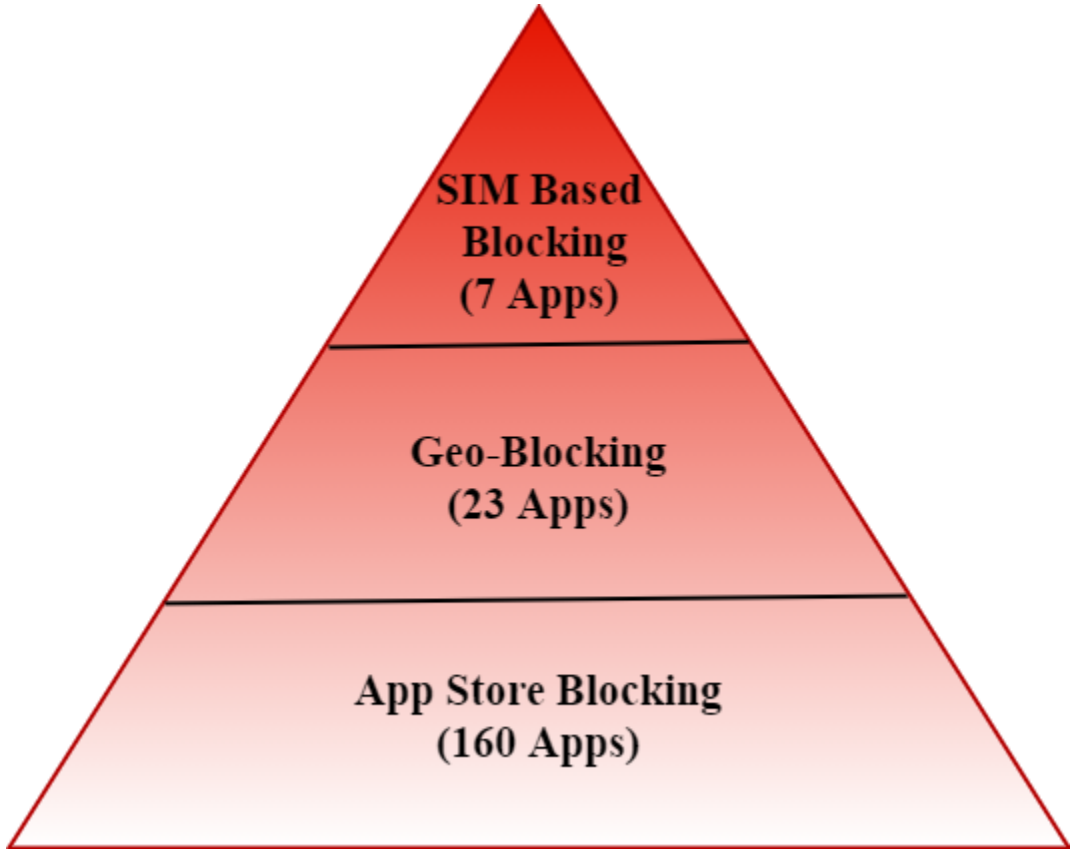
- In Jan 2021, Indian government imposed a permanent ban on the said apps

- Soon after the ban, for the 7 (out of 8) apps, censorship mechanics changed

Sl No.	App Name	App Type	Censorship Technique Used		
			Client Source IP	CDN Edge server	Client SIM Card
1	TikTok	Social	✓	✗	✓
2	Likee	Video Players	✓	✗	✓
3	Kwai	Social	✓	✗	✓
4	UC Browser	Browser	✓	✗	✓
5	FaceU	Photography	✓	✗	✓
6	Hago	Social	✓	✗	✓
7	V-Fly	Tools	✓	✗	✓

In 2021

App censorship (summary)



Conclusion

- We conducted a first systematic study of app censorship (with IN as a case study)
- Indian ISPs are not involved; rather app publishers are themselves filtering the content
- We report a novel three-tiered censorship
- We circumvented all the blocked apps