Fingerprinting Deep Packet Inspection Devices by Their Ambiguities

Diwen Xue

CSE COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY OF MICHIGAN

香港中文大學
The Chinese University of Hong Kong

IRTFOPEN 2026

Academics: Russia deployed new technology to throttle Twitter's traffic

Russia to spend over half a billion dollars to bolster internet censorship system

By **Gleb Stolyarov** and **Lucy Papachristou**

September 10, 2024 1:05 PM EDT · Updated 5 months ago

Aa

Home / Tech / Security

Kazakhstan's HTTPS interception efforts Facebook, Google, Twitter, others

Researchers: HTTPS interceptions happening on one ISP for the moment, target communications services, follow erratic pattern.

Rain throttles Internet speeds for customers on VPNs

Jamie McKane 1 February 2021

PETER GUEST BUSINESS 26.10.2023 01:42 PM

The UK's Controversial Online Safety Act Is Now Law

The UK government says its Online Safety Act will protect people, particularly children, on the internet. Critics say it's ineffective against dangerous misinformation and may be a threat to privacy.

EFF DONATE

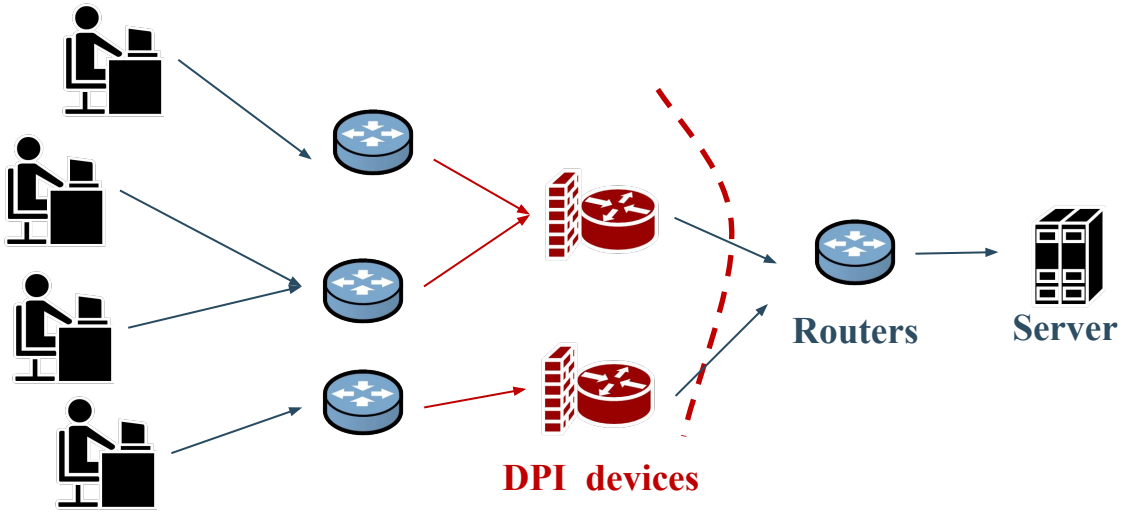
The U.S. Government Wants To Control Online Speech to "Protect Kids"

DEEPLINKS BLOG

Governor Abbott Announces Statewide Plan Banning Use Of TikTok

Austrian ISPs 'Had No Choice' But to Block Pirate Sites AND Cloudflare

Dual-use Deep Packet Inspection (DPI)



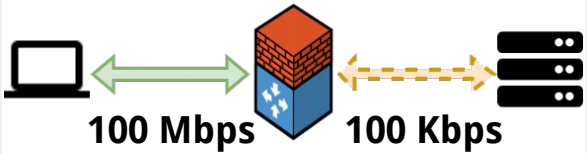
- Intrusion Detection
- Malware Blocking
- Network Management

But also…

- Censorship
- Traffic Interception
- Targeted Throttling
- Injection

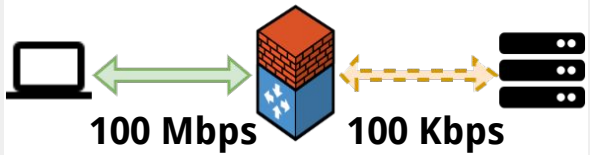
Commoditization of DPIs →
Increased need for transparency and accountability

Example: Targeted Throttling by DPIs



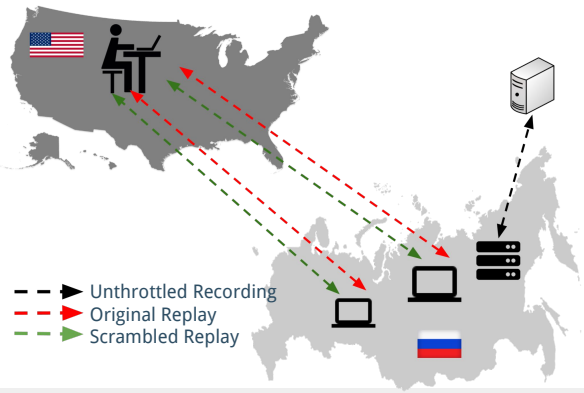
March, 2021:
First-ever large-scale, targeted throttling of user traffic by **DPIs**

Example: Targeted Throttling by DPIs

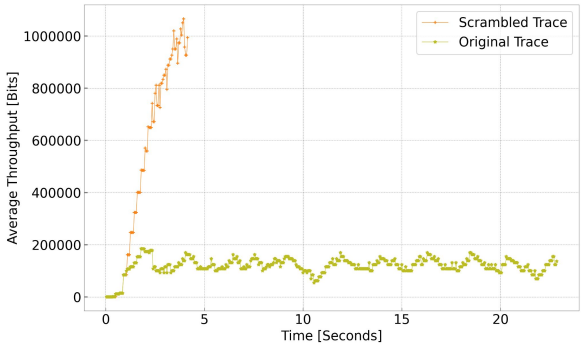


March, 2021:
First-ever large-scale, targeted throttling of user traffic by **DPIs**

Record a transcript, and
1) **replay** from affected networks (throttled)
2) **replay** with permutations (baseline)



3) Reliably observe the **effect** of throttling



Xue, Diwen, et al. "Throttling Twitter: an emerging censorship technique in Russia." *Proceedings of the 21st ACM internet measurement conference* (IMC'21).

The Problem: Measuring DPIs

Existing measurement studies focus on the *effect* of interference, not *DPI devices* responsible

➔ What gets censored?

➔ What gets throttled?

➔ **Who built them?**

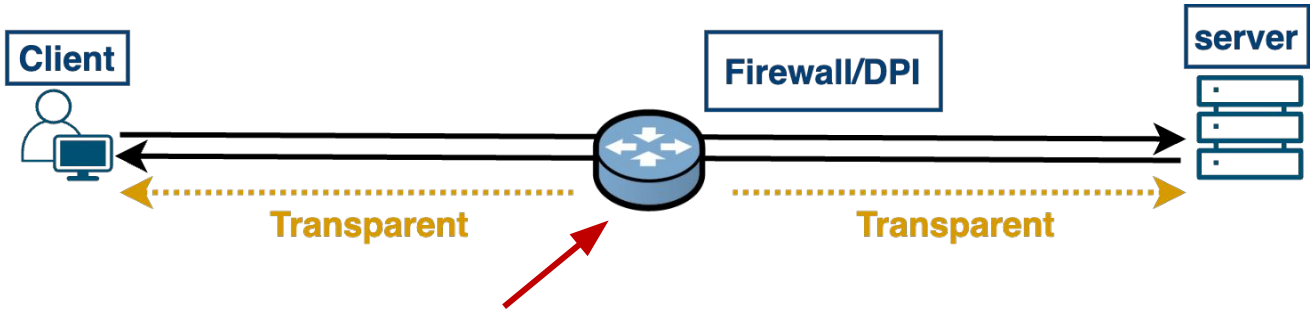
➔ **How they're being disseminated?**

The Problem: Measuring DPIs

Existing measurement studies focus on the *effect* of interference, not *DPI devices* responsible

Measuring DPIs is technically challenging

1. **Network intermediaries "invisible" to typical scanning (Nmap, ZMap)**
2. Vendors intentionally obscure their devices



➔ **How to characterize these intermediaries?**

➔ **How do we make the invisible visible?**

The Problem: Measuring DPIs

Existing measurement studies focus on the *effect* of interference, not *DPI devices* responsible

Measuring DPIs is technically challenging

1. Network intermediaries "invisible" to typical scanning (Nmap, ZMap)

2. **Vendors intentionally obscure their devices**

➔ No banner, no vendor-identifying blockpages. Aim to be **anonymous intermediaries**.

The Problem: Measuring DPIs

Existing measurement studies focus on the *effect* of interference, not *DPI devices* responsible

Measuring DPIs is technically challenging

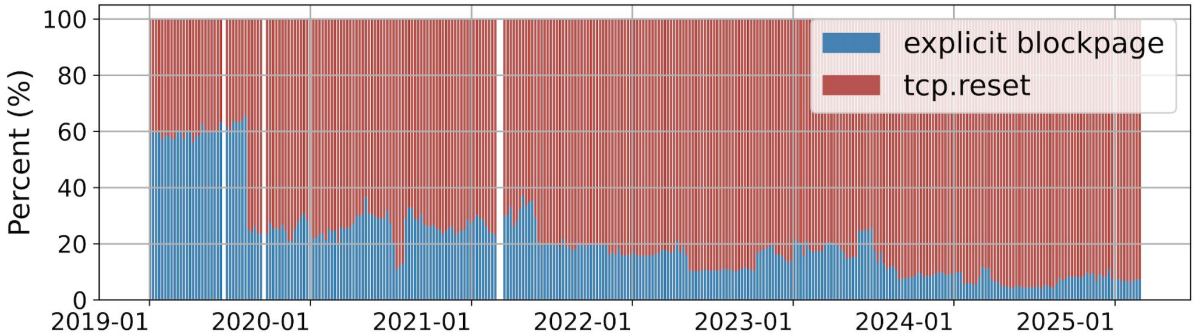
1. Network intermediaries "invisible" to typical scanning (Nmap, ZMap)

2. **Vendors intentionally obscure their devices**

Previously: blockpages are vendor-identifying

- "Measuring the Deployment of Network Censorship Filters at Global Scale" (NDSS'20)

Problem: DPI actions are becoming generic



Example: Throttling DPIs in Russia

The throttling is real — but whose devices?

Context: Russia followed a **decentralized censorship model** prior to 2021

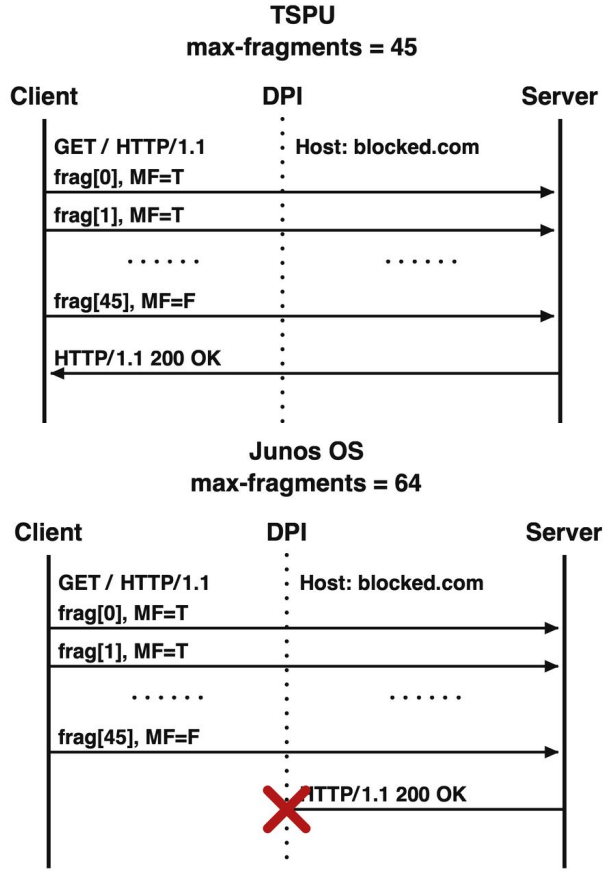
- One unified censorship blocklist
- Enforcement left to the discretion of individual ISPs (each with own commercial DPI devices)
- “Decentralized Control: A Case Study of Russia” (NDSS’20)
- Rumor about **a new, centralized DPI system**

Throttling: ISP commercial DPIs or national firewall?

Example: Throttling DPIs in Russia

The throttling is real — but whose devices?

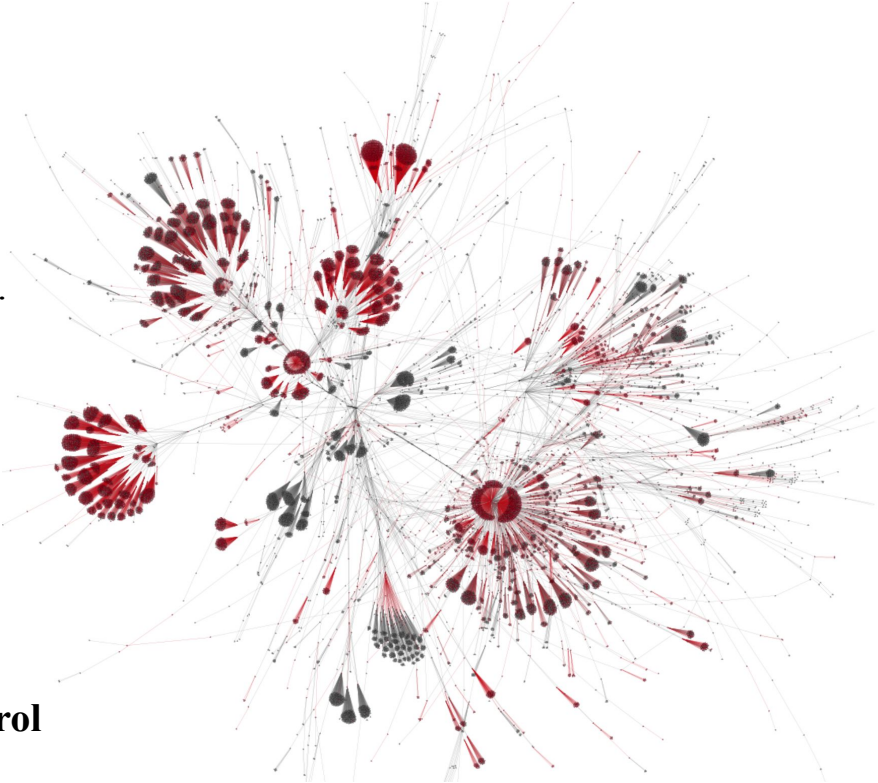
- Throttling itself carries little identifying information.
- But the throttling DPIs have a strange IP fragmentation reassembly behaviors
  - Discards overlapping fragments
  - Changes the TTL of fragments
  - Accepts up to **45 fragments**
- High degree of uniformity suggested central control



Example: Throttling DPIs in Russia

The throttling is real — but whose devices?

- Throttling itself carries little identifying information.
- But the throttling DPIs have a strange IP fragmentation reassembly behaviors
  - Discards overlapping fragments
  - Changes the TTL of fragments
  - Accepts up to 45 fragments
- **High degree of uniformity suggested central control**



"TSPU: Russia's Decentralized Censorship System" (IMC'22)

dMAP : Framework for Fingerprinting DPIs

Fingerprinting Deep Packet Inspection Devices by their Ambiguities

Diwen Xue, A. Huremagic, W. Wang, R. Ram Sundara Raman, R. Ensafi

dMAP :
A framework for characterizing and
fingerprinting network intermediaries (e.g., firewall, DPI)

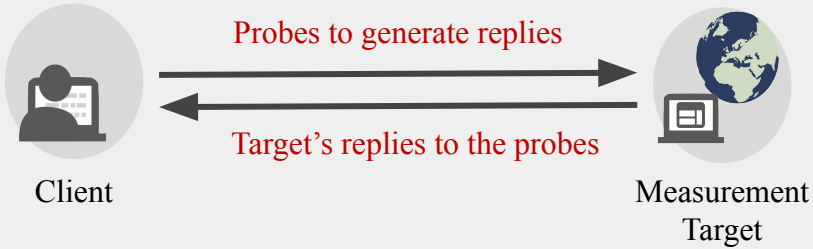
Following the same naming convention, but fundamentally different
from traditional scanners

dMAP : Framework for Fingerprinting DPIs

Fingerprinting Deep Packet Inspection Devices by their Ambiguities

Diwen Xue, A. Huremagic, W. Wang, R. Ram Sundara Raman, R. Ensafi

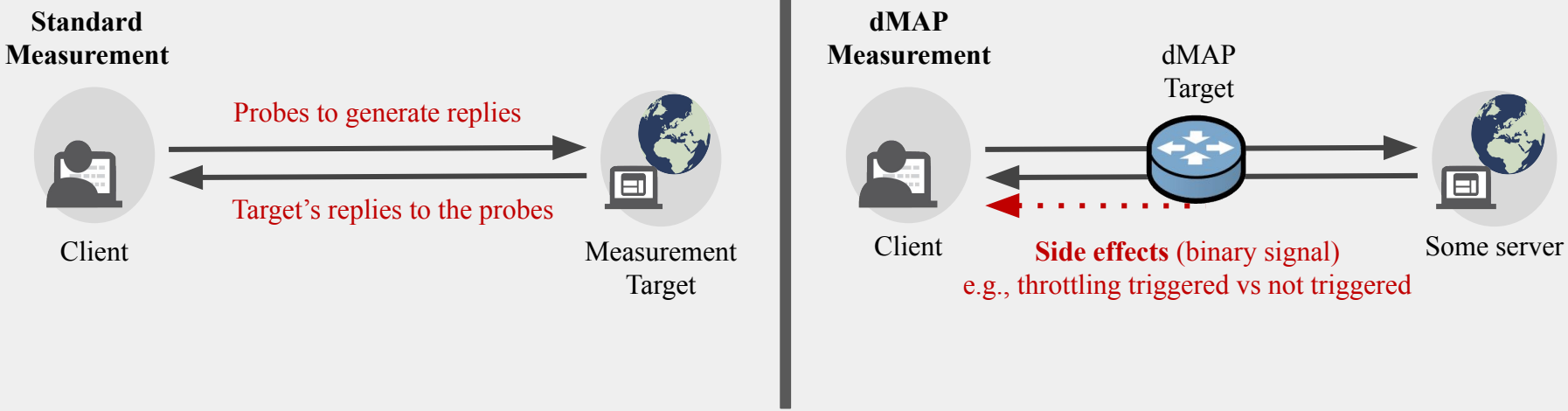
Standard Measurement



dMAP : Framework for Fingerprinting DPIs

Fingerprinting Deep Packet Inspection Devices by their Ambiguities

Diwen Xue, A. Huremagic, W. Wang, R. Ram Sundara Raman, R. Ensafi

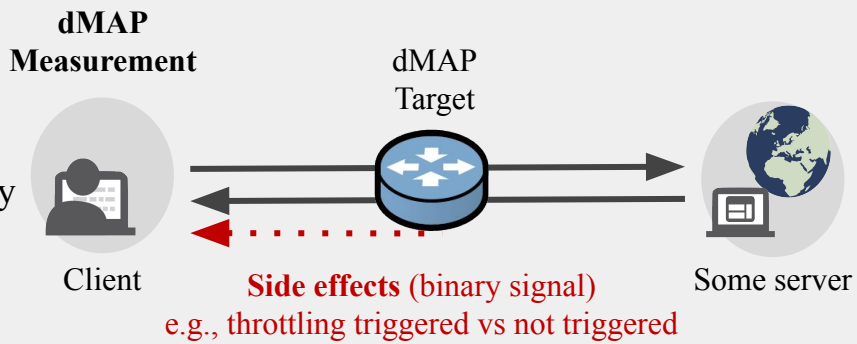


dMAP : Framework for Fingerprinting DPIs

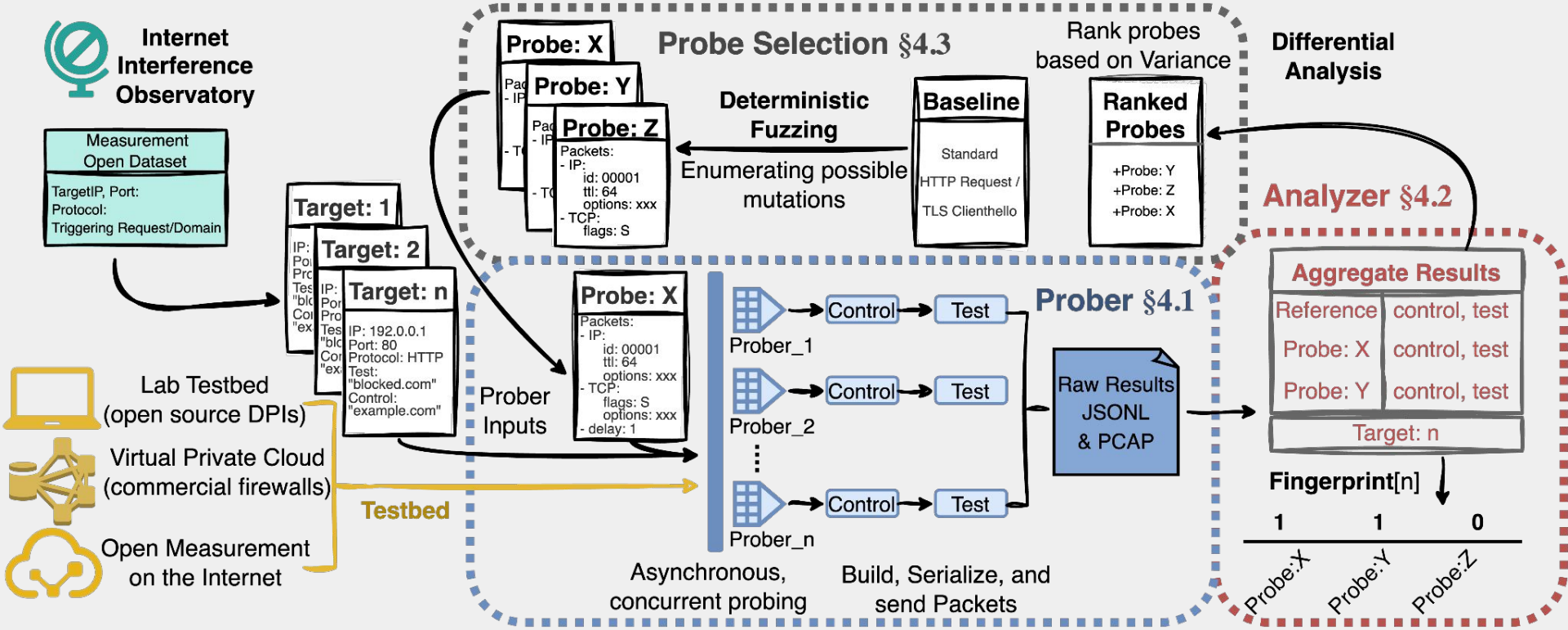
Fingerprinting Deep Packet Inspection Devices by their Ambiguities

Diwen Xue, A. Huremagic, W. Wang, R. Ram Sundara Raman, R. Ensafi

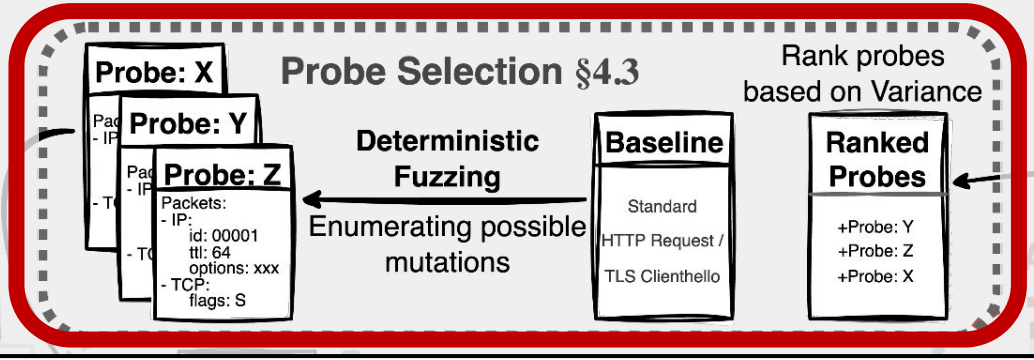
dMAP :
using this binary signal as a **side-channel**
to **infer 1-bit information** each time about the intermediary
to **gradually build a fingerprint**.



dMAP : Framework for Fingerprinting DPIs



dMAP : Framework for Fingerprinting DPIs



- ➔ Probes that contain "**ambiguities**", whose handling is not well-defined
- ➔ Ambiguities lead to **differing handling across DPIs**, allowing for fingerprinting

dMAP : Framework for Fingerprinting DPIs

Ambiguities in Interpreting Network Traffic

- **Under-specified Behaviors**
  - Buffer size, timeouts, state management
- **Ambiguities in Specifications**
  - Languages leave room for varied interpretation
  - more than one logical form, missing context
- **Handling of Edge-cases**
  - DPIs may not implement full protocol behaviors (e.g., rare options, short-cuts)

dMAP : Framework for Fingerprinting DPIs

Ambiguities in Interpreting Network Traffic

- ➔ **Under-specified Behaviors**
  - ◆ Buffer size, timeouts, state management
- ➔ **Ambiguities in Specifications**
  - ◆ Languages leave room for varied interpretation
  - ◆ more than one logical form, missing context
- ➔ **Handling of Edge-cases**
  - ◆ DPIs may not implement full protocol behaviors (e.g., rare options, short-cuts)

Example: **TCP Timestamp**:

Open-source DPIs **disagree** on*:

1. **Process or not?**
2. **How to parse?**
3. **What's a valid time?**
4. **What to do if the timestamp is invalid?**

* Zeek, Suricata, Snort, nDPI

dMAP : Framework for Fingerprinting DPIs

Common Probes

Layer	Type	Name	Description
IP	Fragment	Fragment[l:IP;t:maxDist;maxdist:16]	Split the triggering request at the IP layer into two fragments; send the first fragment; then send 16 dummy fragments with random data with the same IP addresses but different IPID; finally send the second fragment of the triggering request.
TCP	Insert	Insert[p:I3;f:P;d:altProto;option:]	Insert a TCP packet containing a non-triggering request of the other protocol (i.e., send a HTTP GET if the current measurement is HTTPS, or a TLS clienthello if HTTP), before sending the triggering request.
TCP	Mutate		oad with padding so
IP	Fragment		nd them backwards.
IP	Fragment		with two fragments rger offset and ends
IP	Mutate	Mutate[l:IP;f:option;option:noop]	Add a NO-Operation (NOP) option to the IP header of the triggering request.
TCP	Mutate	Mutate[l:TCP;f:checksum;checksum:corrupt]	Corrupt the TCP checksum of the triggering request.
TCP	Mutate	Mutate[l:TCP;f:urgentPointer;option:noack]	Add the Urgent TCP flag to the TCP header of the triggering request while removing the ACK flag. Use a random ($\leq len(payload)$) value as the value of the urgent pointer.
TCP	Insert	Insert[p:I3;f:P;d:controlRequest;option:]	Insert a TCP packet containing a non-triggering request, with only PSH flag (no ACK) before sending the triggering request.
TCP	Mutate	Mutate[l:TCP;f:option;option:timestamp]	Add a TCP Timestamp option to the TCP header of all outgoing packets. Then mutate the timestamp of the triggering request to a timestamp earlier than that of the preceding outgoing packet.

➔ **Ground-truth testbed with commercial & open-source DPIs**

➔ **Differential fuzzing generated > 2000 candidate probes**

Example Probe: Negative Sequence Number handling

Mutate{layer:TCP;field:seq;option:negativeSeqWithPadding}

$packet_0$: [SYN (seq=ISN_{client}, ack=0)]
\\ waiting for incoming ([SYNACK (seq=ISN_{server}, ack=ISN_{client}+1)])
$packet_1$: [ACK (seq=ISN_{client}+1, ack=ISN_{server}+1)]
$packet_2$: [PSH/ACK (seq=ISN_{client}-100, ack=ISN_{server}+1,
payload=[0]*101 + request[TestDomain])]
$packet_{3-4}$: [FIN/ACK...], [ACK ...]

Zeek & nDPI: check $seq \leq client_ISN$

Snort: check $seq \leq rcv_next + window_size$ && $seq + len(packet) \geq rcv_next$

Example Probe: PAWS Check

Mutate{layer:TCP;field:option;option:timestamp}

$packet_0$: [SYN (timestamp: 1000)]
\\ waiting for incoming ([SYNACK (timestamp: 2000)])
$packet_1$: [ACK (timestamp: 1001)]
$packet_2$: [PSH/ACK (timestamp: 999, payload=request[TestDomain])]
$packet_{3-4}$: [FIN/ACK...], [ACK ...]

Protection Against Wrapped Sequence number (PAWS):
- discards packet with old timestamp as an old duplicate
- Zeek, nDPI: not implemented at all
- Snort: implemented the **opposite behavior**

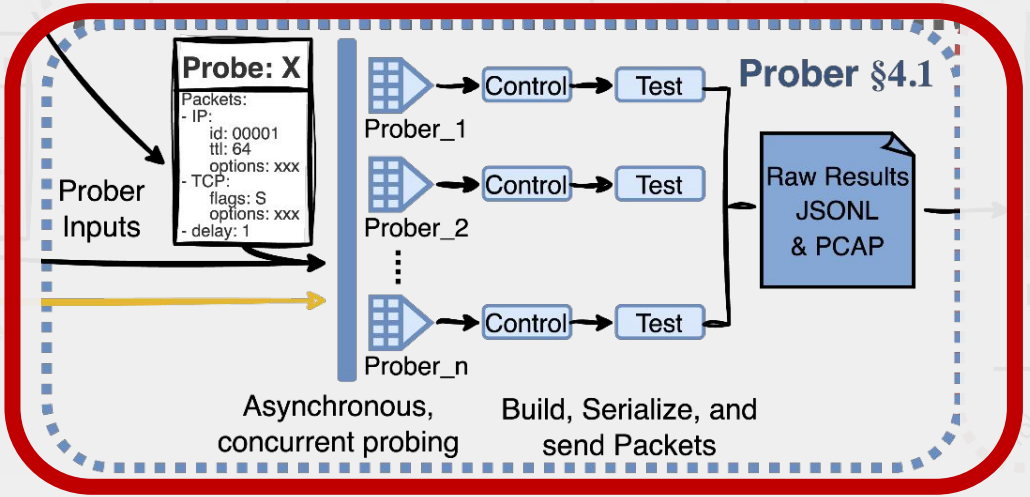
All wrong, but wrong in fingerprintable ways.

RFC 7323:
"...require that PAWS … to invalidate the TS.Recent value when a connection is idle…"

RFC: **invalidate the check** (accept the packet)
Snort: **invalidate the packet** (drop the packet)

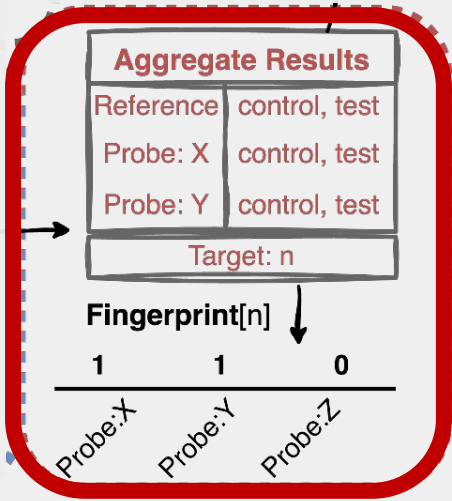
dMAP : Framework for Fingerprinting DPIs

- ➔ Send **ambiguous probes along with "triggers"** (e.g., twitter.com)
- ➔ Look for **side-effects** (e.g., throttling triggered vs. not triggered)
- ➔ For each ambiguity / probe, measure a **binary outcome** $\in \{1, 0\}$



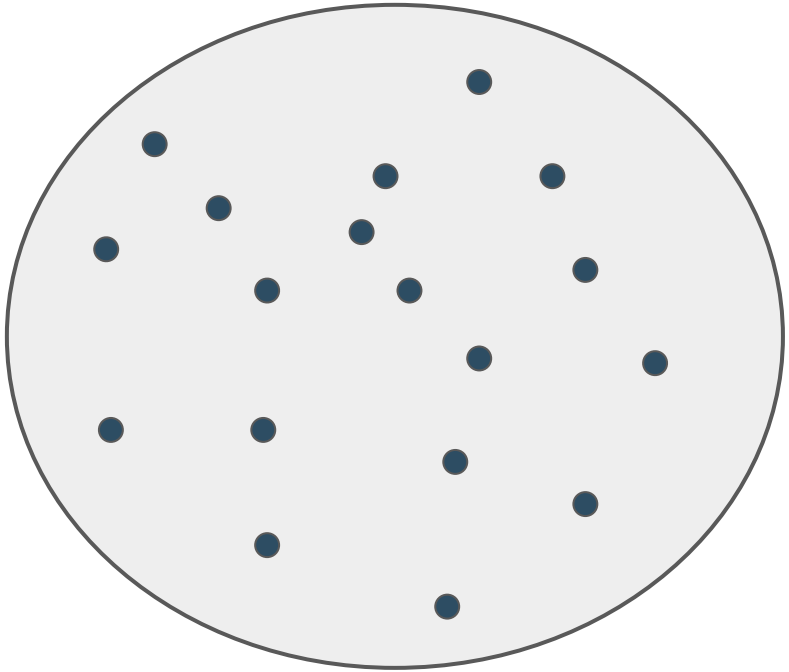
dMAP : Framework for Fingerprinting DPIs

➔ Concatenate binary outcomes together into a **fingerprint**



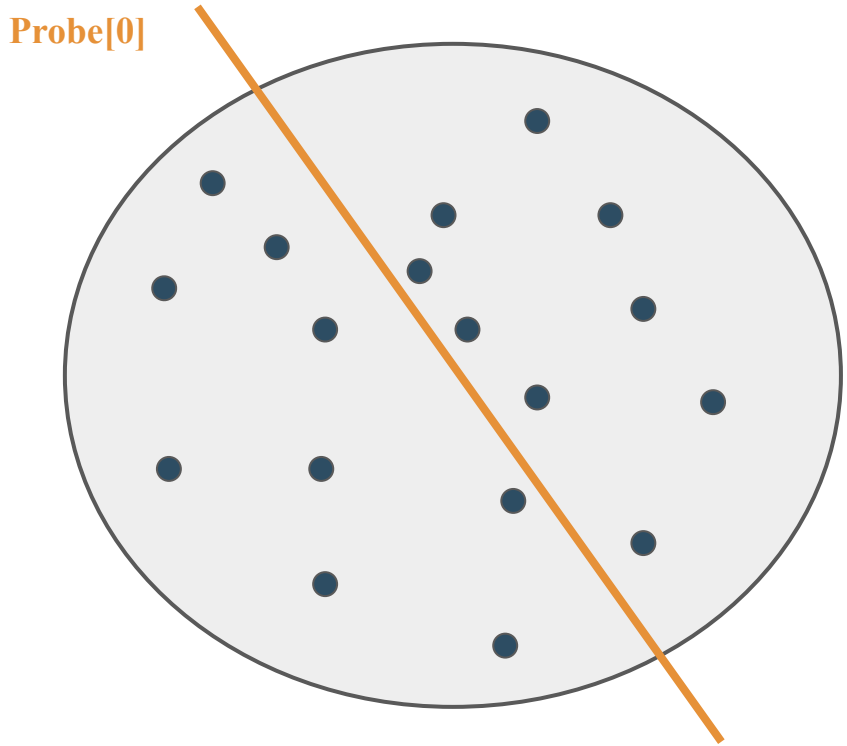
dMAP : Framework for Fingerprinting DPIs

- **Probes are different in their discriminative power**
- **Each probe divides all DPIs into two partitions**
- **More probes => smaller, more homogeneous partitions**



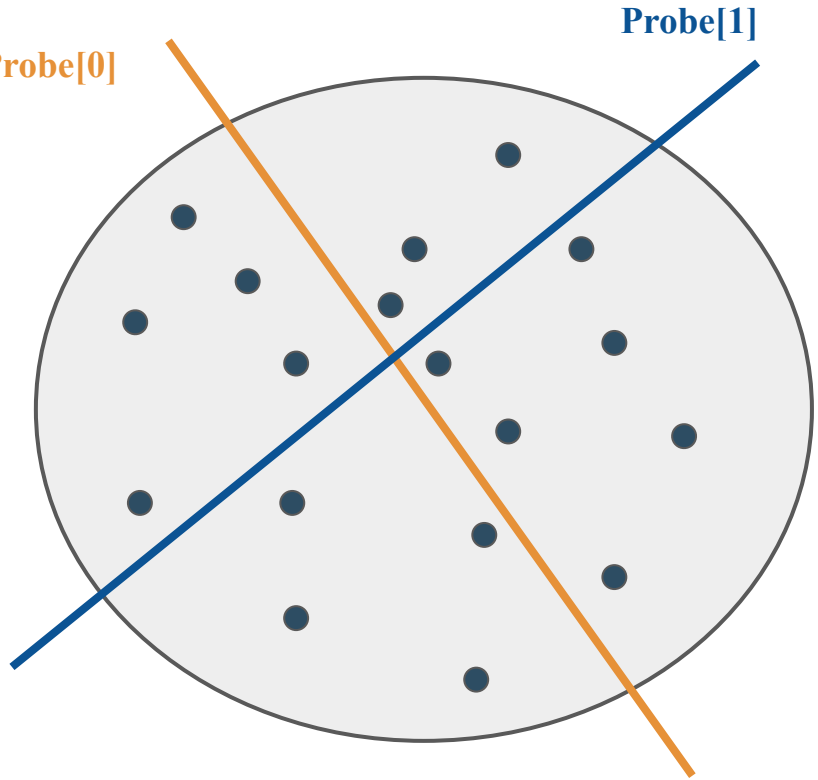
dMAP : Framework for Fingerprinting DPIs

- **Probes are different in their discriminative power**
- **Each probe divides all DPIs into two partitions**
- **More probes => smaller, more homogeneous partitions**



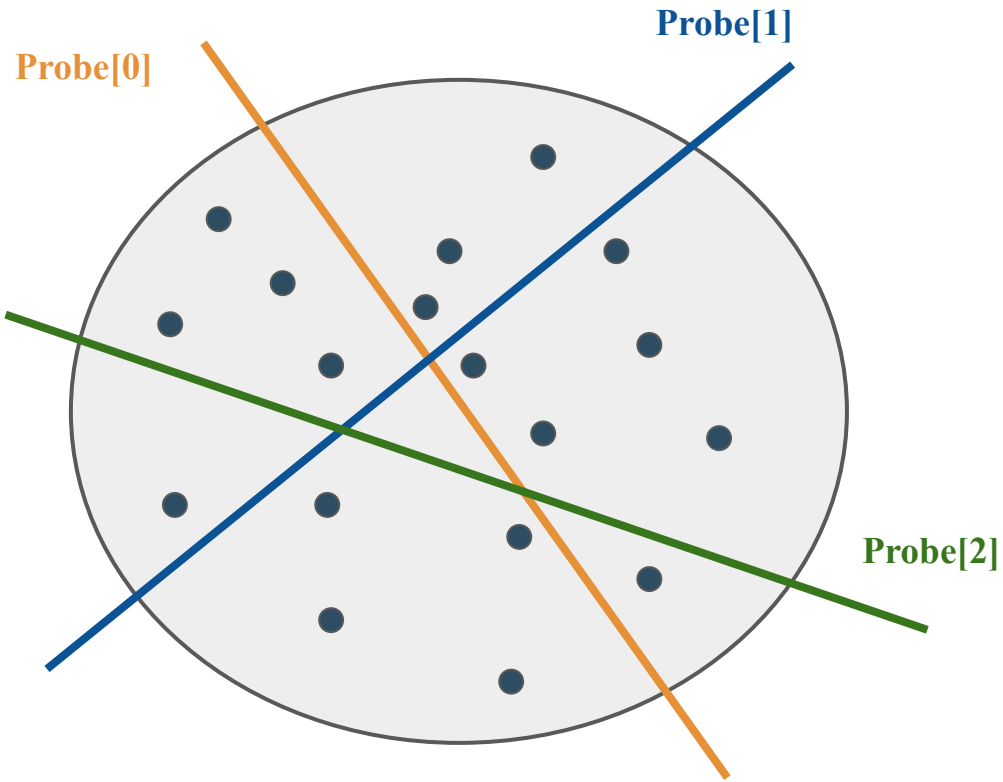
dMAP : Framework for Fingerprinting DPIs

- **Probes are different in their discriminative power**
- **Each probe divides all DPIs into two partitions**
- **More probes => smaller, more homogeneous partitions**



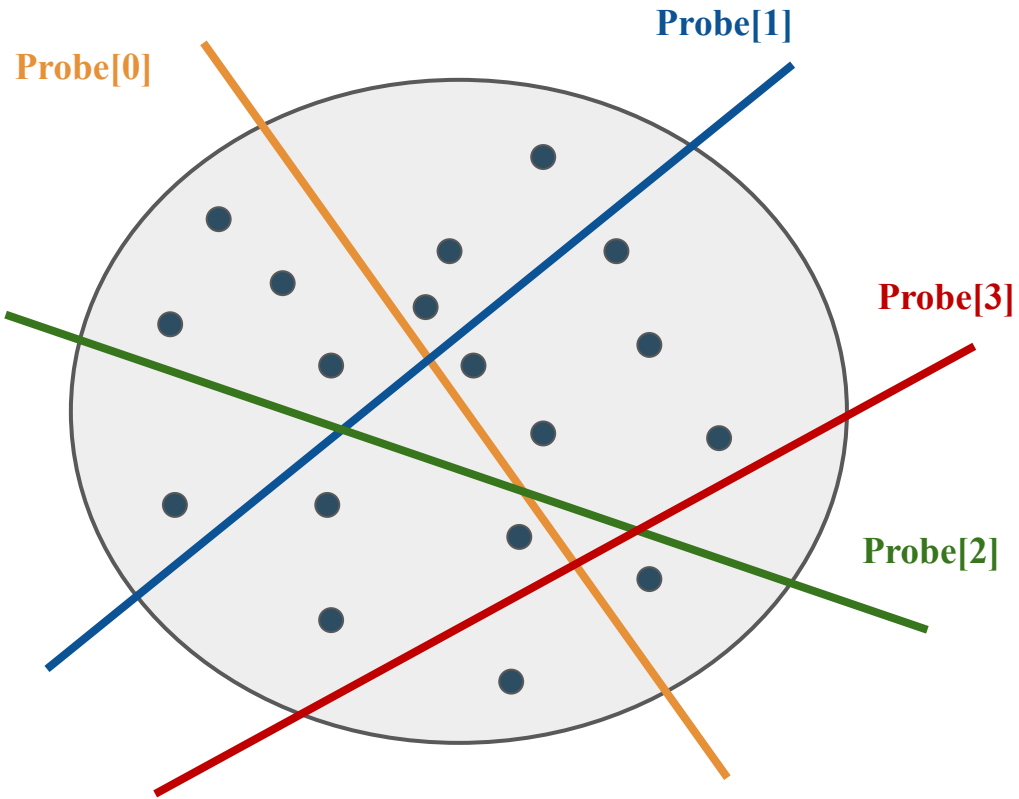
dMAP : Framework for Fingerprinting DPIs

- **Probes are different in their discriminative power**
- **Each probe divides all DPIs into two partitions**
- **More probes => smaller, more homogeneous partitions**



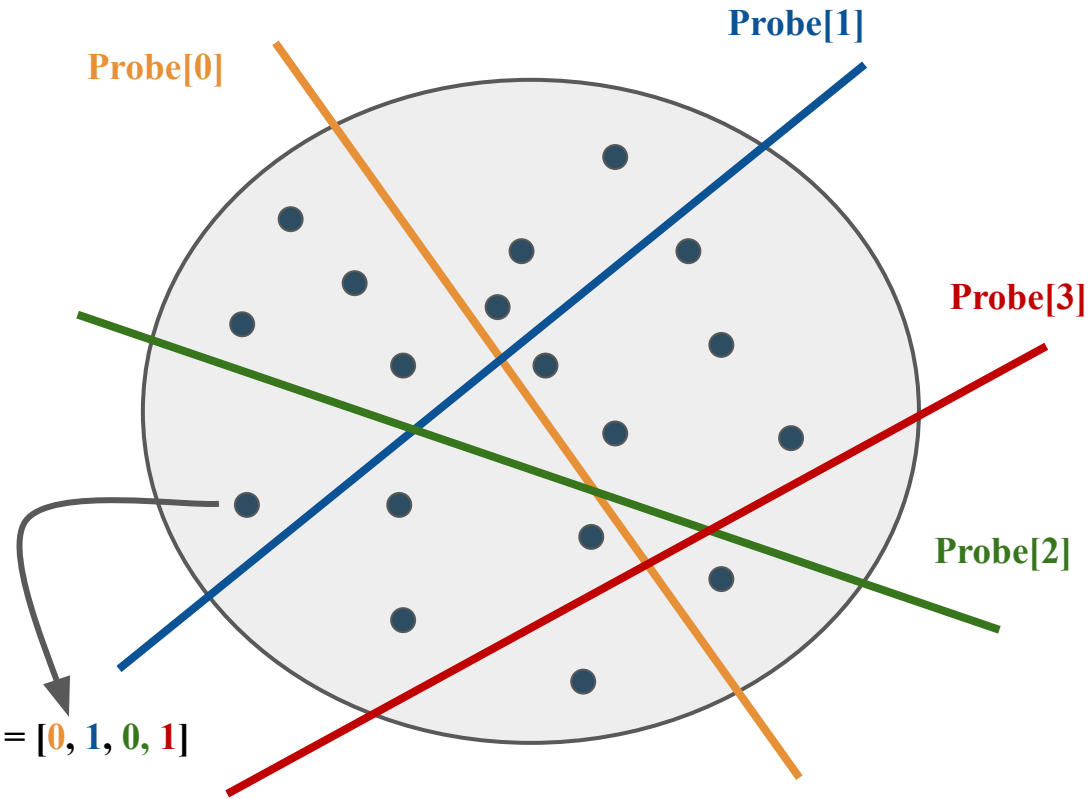
dMAP : Framework for Fingerprinting DPIs

- **Probes are different in their discriminative power**
- **Each probe divides all DPIs into two partitions**
- **More probes => smaller, more homogeneous partitions**



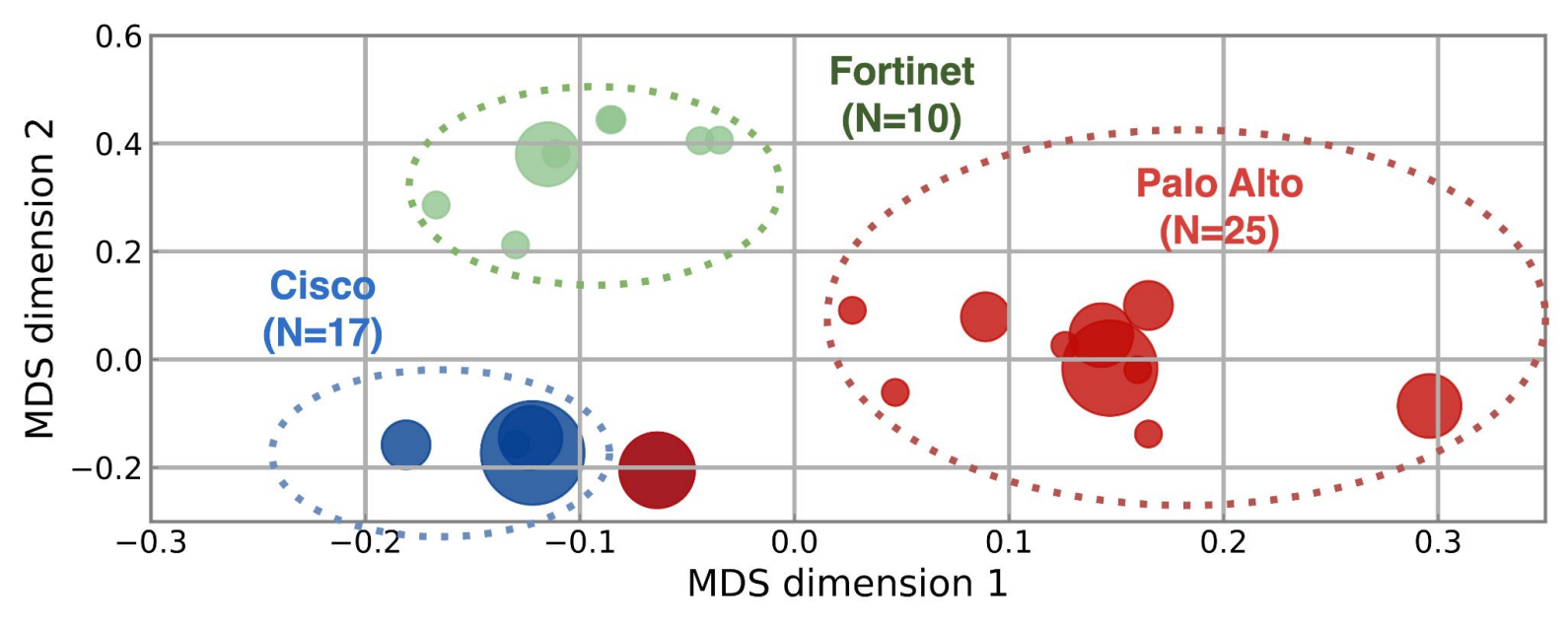
dMAP : Framework for Fingerprinting DPIs

- Probes are different in their discriminative power
- Each probe divides all DPIs into two partitions
- More probes => smaller, more homogeneous partitions

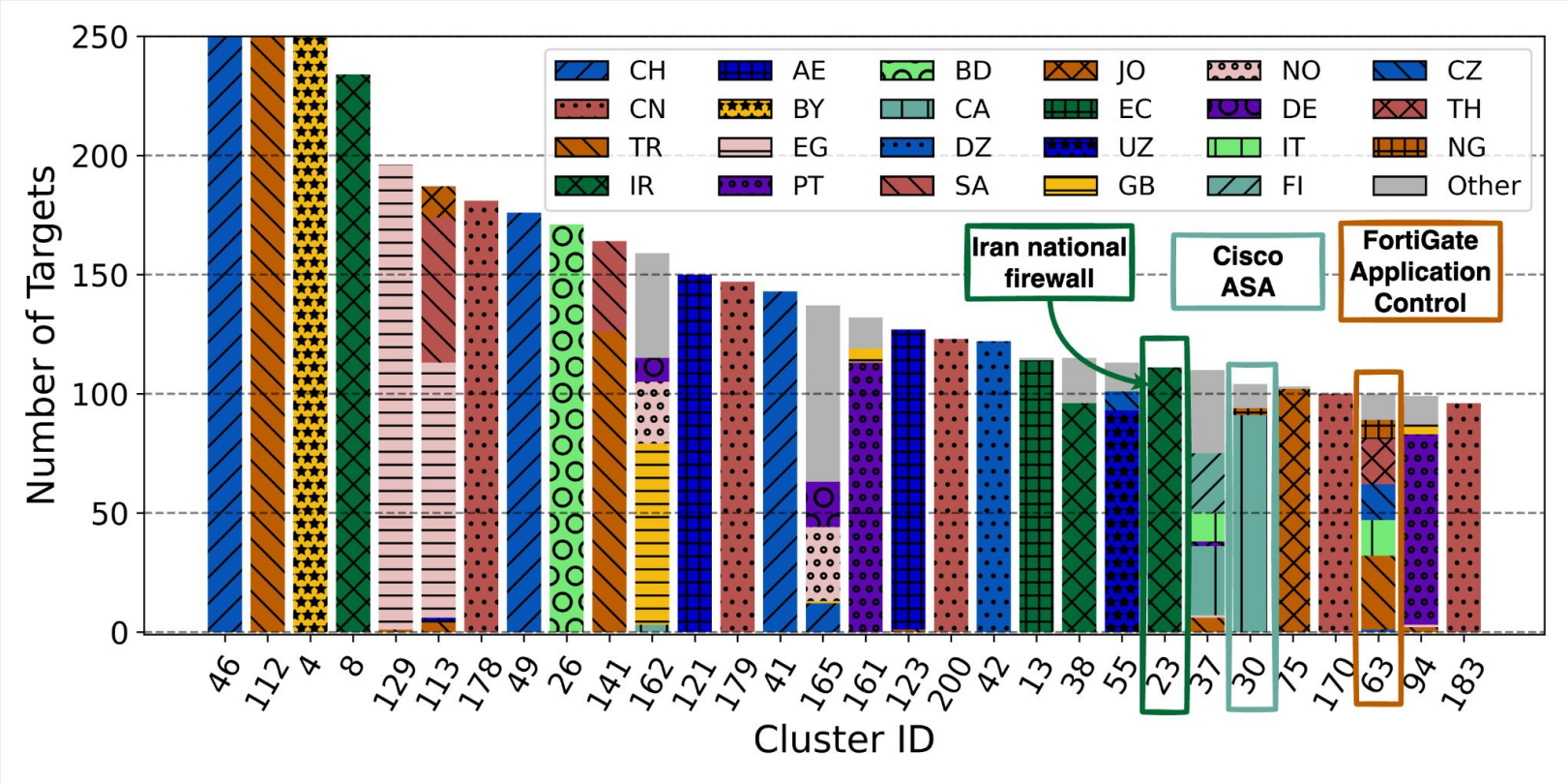


Applying *dMAP* In Practice

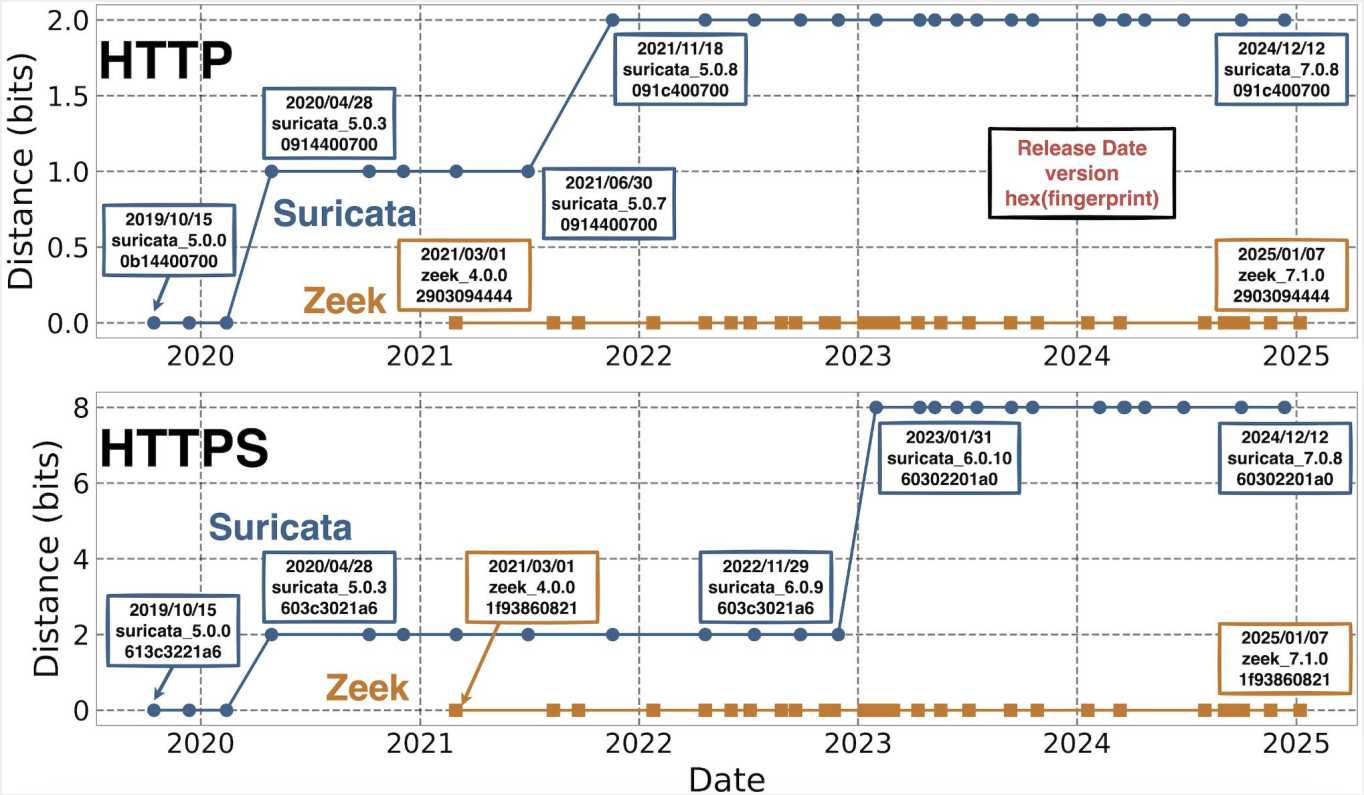
40-probe (40-bit) *dMAP* fingerprint, projected down into two dimensions with MDS.
Firewalls from the same vendor cluster closer together.



Applying *dMAP* In Practice



Churn in DPI Fingerprints

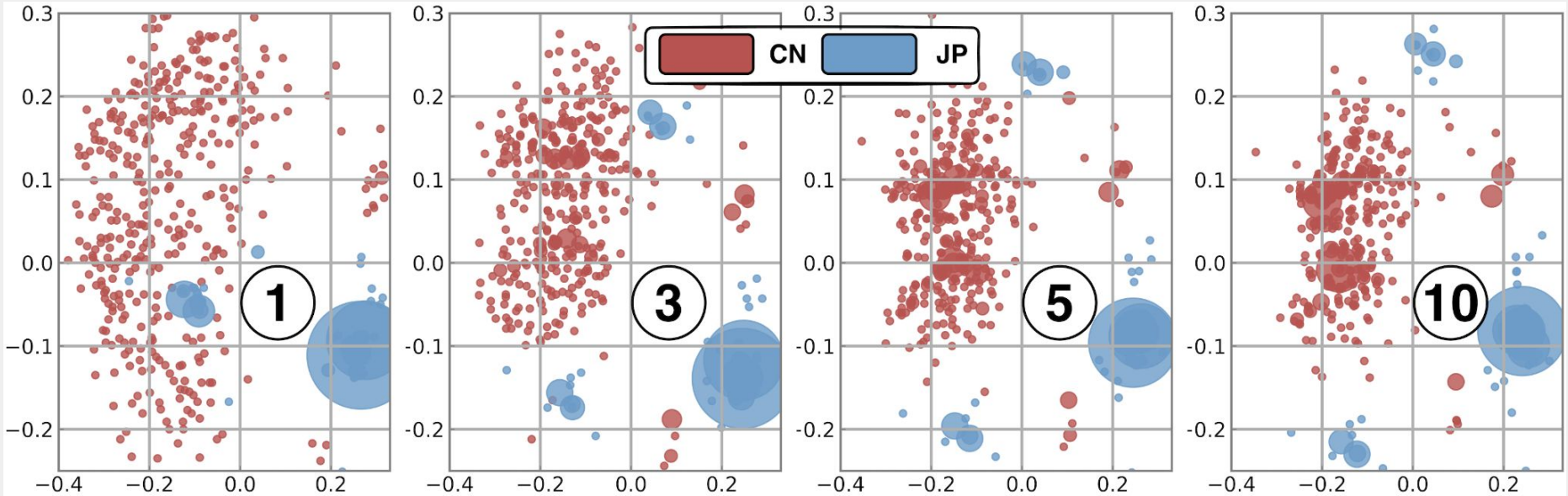


Fingerprinting in Noisy Environments

Limitation: Sporadic interference (censorship) causes "**bit-flipping**".

Solution: Repeat measurements, then **aggregate outcomes with logical OR** (prioritize "1"s over "0"s)

- Under the assumption that most DPIs fail-open rather than fail-close



Constraints of Remote Measurements

Limitation: We're measuring the *Path*.

Observed apparent blocking, could be due to
-> Ambiguity had no effect
-> Packet dropped or normalized upstream
-> End-host no longer live

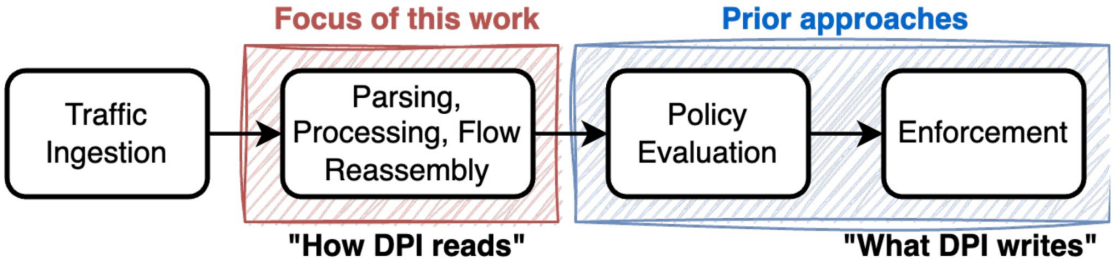
(Partial) Solution: Controls & Tests
-> 4 measurements per bits
-> (Benign vs. Triggering) x (w/ vs. w/o ambiguity)
-> Keep only valid groupings

Results	Interpretation	Verdict
{R1,R3,R4}{R2}	Mutation has no effect on endhost but bypasses DPI	**Bypass**
{R1}{R2,R3,R4}	Reference blocking behavior indistinguishable from how endhost handles mutation, unclear if DPI is triggered in R4	Inconclusive
{R1,R3}{R2,R4}	Mutation has no impact on either the DPI or the endhost	**NoEffect**
{R1,R3}{R2}{R4}	Mutation does not affect endhost; however, differing blocking behaviors imply possibly two distinct DPIs in the path	Inconclusive
{R2,R3}{R1}{R4}	Possibly two DPIs in path: first drops all packets when triggered (R2), superseding second DPI's explicit blockpage. Mutation bypasses the first DPI but not the second.	Inconclusive
{R2,R4}{R1}{R3}	Mutation affects the endhost but the DPI remains triggered.	**NoEffect**
{R3,R4}{R1}{R2}	Could happen when Mutated Control (R3) and Mutated Test (R4) both yield no response, different from reference behaviors.	Inconclusive
{R1}{R2}{R3}{R4}	Mutation affects both the endhost and potentially multiple DPI implementations in the path that respond differently.	Inconclusive

➔ **Only 3 out of 15 possible groupings are valid**

Can't fully rule out: 2 DPIs, same hop, same policy -> one compound fingerprint

Conclusion

- We need better understanding of how DPIs are *deployed* and *disseminated*.
  - Prior approach relied on self-identifying blockpages, not applicable anymore.

- *dMAP*: A first step to measure DPIs
  - To remotely differentiate and cluster based on their implementations.
  - Key idea: A shift of focus from what DPIs write to how DPIs read.
  - Non-trivial limitations remain open problems.