

A pair of silver-colored metal calipers is shown measuring a small white cutout of a human figure. The calipers are positioned vertically, with the jaws of the tool gripping the top and bottom of the figure. The main beam of the calipers has a scale in millimeters, with markings from 0 to 10. The sliding vernier scale has markings from 0 to 10, with a label '1/20mm' indicating the resolution. The background is a dark, textured surface.

Measurement Studies with Ark
and Scamper-DSL

The image features three red, 3D location pins of different sizes. One is in the foreground, one is in the middle ground, and one is in the background. They are positioned on a dark, textured surface that has a light-colored, wavy path or road leading towards them. The background is a soft, light blue gradient, suggesting a sky or a digital interface. The text "RTT-based Geolocation" is centered over the image in a white, serif font.

RTT-based Geolocation



Motivation

Darwich et al.'s replication paper from IMC 2023.

Greedily selected VPs to maximize geographical coverage, probed targets and computed CBG.

Using those constraints, they probed the same targets again, selecting one VP per AS/city pair to pinpoint the target's location.

Limited to 2 rounds as RIPE Atlas API call takes minutes to return the results.



Why a good fit?

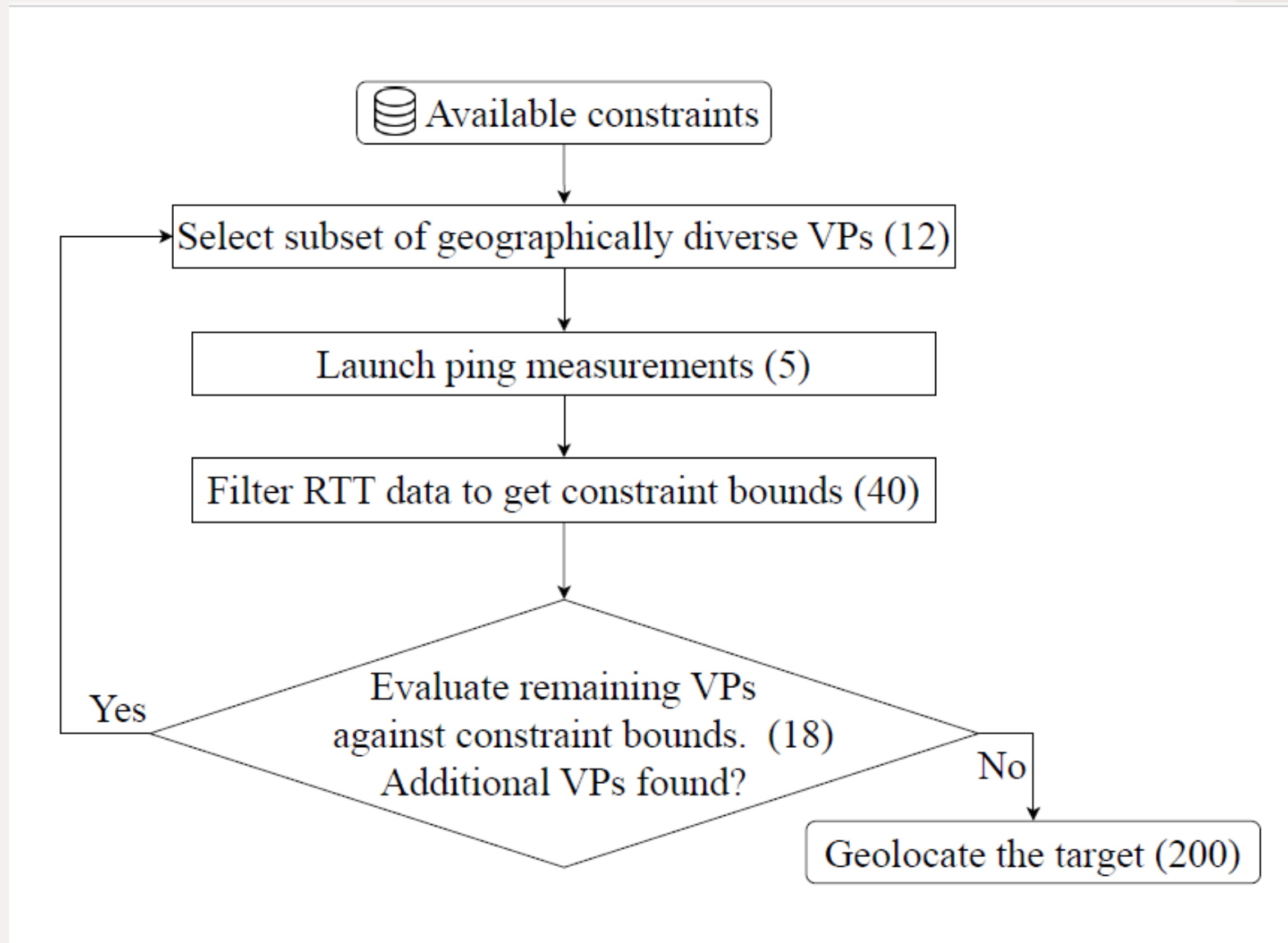
Event-Based
Asynchronous Execution

Integration with Existing
Research and Tools

Blocking, handles non-
responsive targets

Adaptation of existing
active measurement
code using scamper
primitives

Geolocation flow diagram, with lines of code per block.

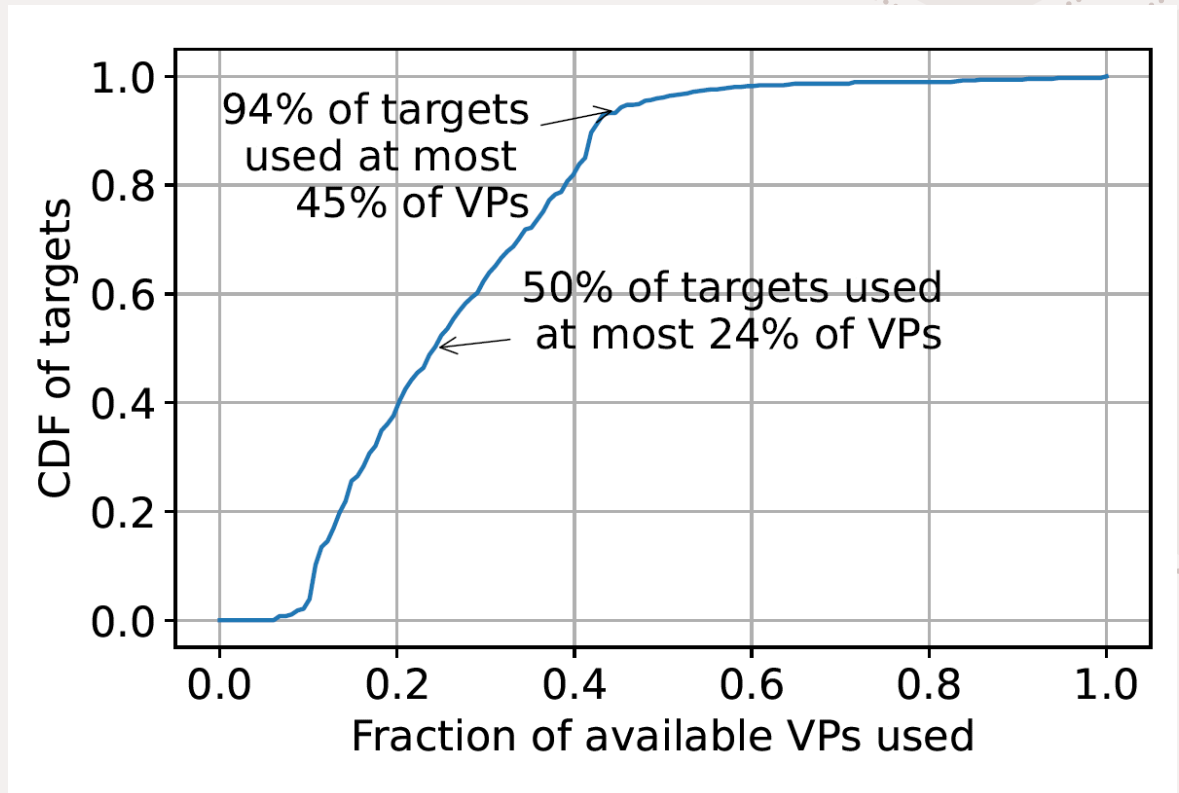




Comparison of % of Ping Measurements

VPs in the first step	Measurements
10	5.95M
100	4.59M
300	3.30M
500	2.88M
1000	2.92M
All	21.7M

Replicability Paper Results



Scamper-DSL with ARK Results

NETFLIX

Measuring CDN catchment and
routing



Motivation

Characterizing CDNs through Geographically Distributed Probing

Analysis of Netflix's CDN through fast.com



Why a good fit?

Comprehensive Support for Required Operations

DNS lookups, HTTP support to fetch JSON, Traceroute, Ping

Leveraging Python's Capabilities

Python's built-in JSON Parsing

Asynchronous and Event-Based Design

Blocking, handles non-responsive targets

Our Solution with Ark

DNS Lookup to fetch IP
address of fast.com

HTTP call to fetch JSON URLs
from the fast.com RESTful API

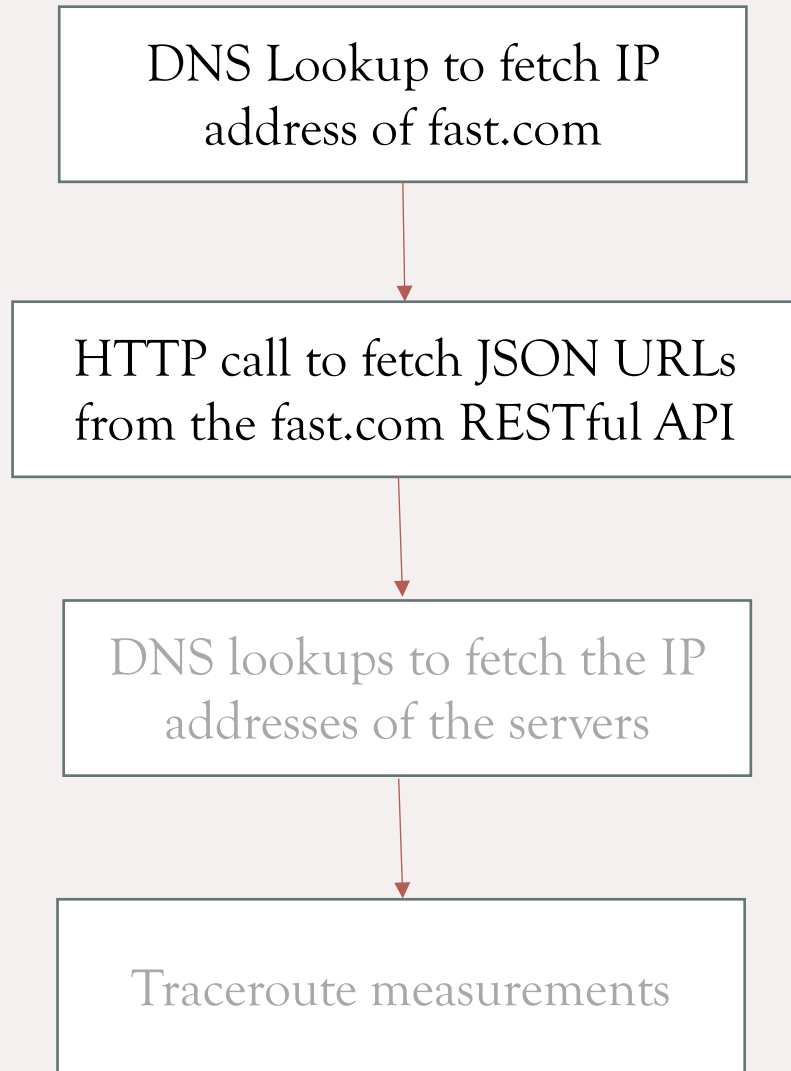
DNS lookups to fetch the IP
addresses of the servers

Traceroute measurements

```
url = "https://api.fast.com/netflix/speedtest/v2?"
date = datetime.now().strftime("%Y-%m-%d_%H:%M:%S")
filename = f"fast.{date}.warts.gz"
out = ScamperFile(filename, 'w')
ctrl = ScamperCtrl(remote_dir=sys.argv[1],
                  outfile=out)

# query for api.fast.com IP address from each VP
http_addrs = {}
for inst in ctrl.instances():
    ctrl.do_dns(urlparse(url).hostname, inst=inst)
for obj in ctrl.responses(timeout=5):
    addrs = obj.ans_addrs()
    if len(addrs) > 0:
        http_addrs[obj.inst] = addrs[0]
```

Our Solution with Ark



```
# HTTP query for server names from each VP
dns_hosts = {}
for inst, ip in http_addrs.items():
    ctrl.do_http(ip, url, inst=inst)
for obj in ctrl.responses(timeout=25):
    if not isinstance(obj, ScamperHttp): continue
    json_data = json.loads(obj.response.decode())
    tgts = json_data.get("targets", [])
    tgturls = [tgt["url"] for tgt in tgts]
    dns_hosts[obj.inst] = [urlparse(tgt).hostname
                           for tgt in tgturls]
```

Our Solution with Ark

DNS Lookup to fetch IP
address of fast.com

HTTP call to fetch JSON URLs
from the fast.com RESTful API

DNS lookups to fetch the IP
addresses of the servers

Traceroute measurements

```
# query for speedtest server IPs from each VP
server_ips = {}
for inst, hosts in dns_hosts.items():
    for host in hosts:
        ctrl.do_dns(host, inst=inst)
for obj in ctrl.responses(timeout=5):
    if isinstance(obj, ScamperHost):
        server_ips[obj.inst] = obj.ans_addrs()
```

Our Solution with Ark

DNS Lookup to fetch IP
address of fast.com

HTTP call to fetch JSON URLs
from the fast.com RESTful API

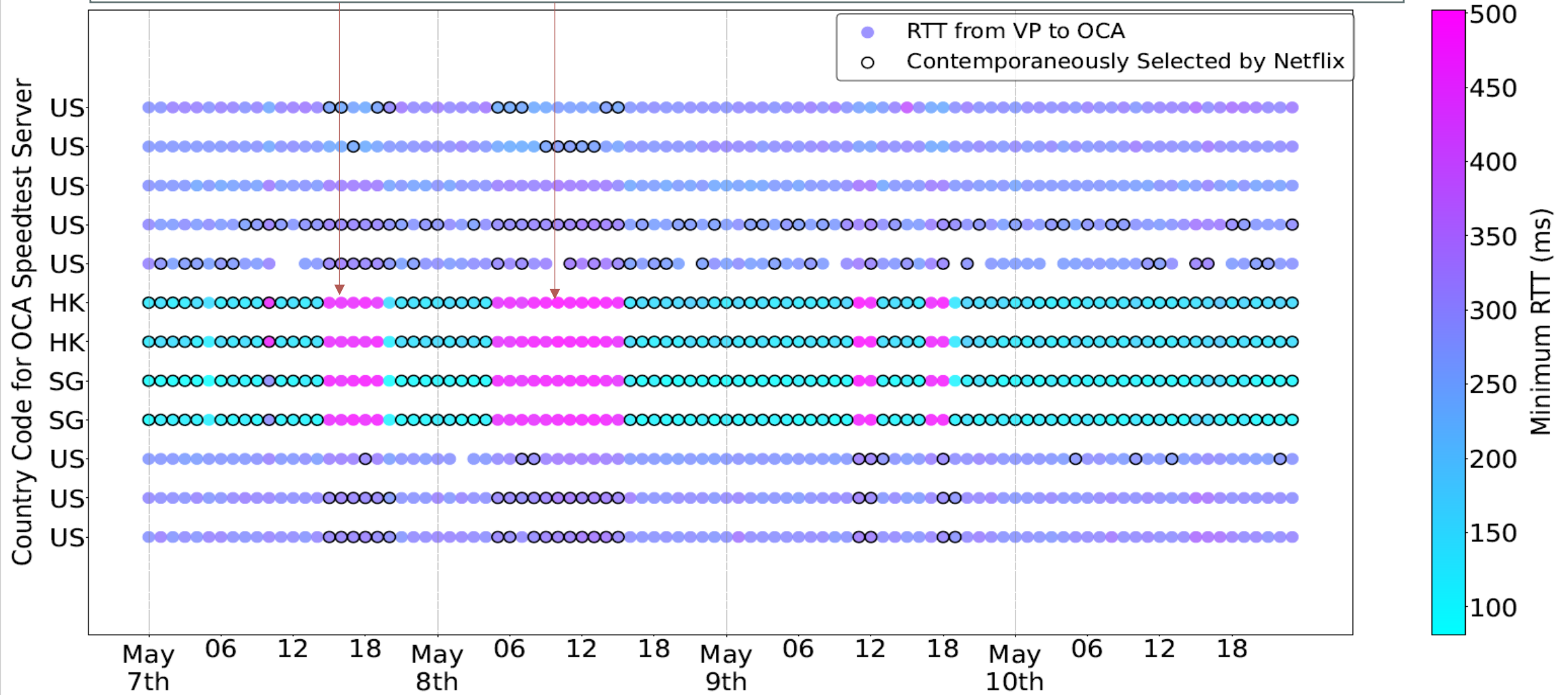
DNS lookups to fetch the IP
addresses of the servers

Traceroute measurements

```
# collect topology to each IP with TCP traceroute
for inst, ips in server_ips.items():
    for ip in ips:
        ctrl.do_trace(ip, wait_timeout=1, dport=443,
                      method="tcp", inst=inst)
for obj in ctrl.responses(timeout=25): pass
```

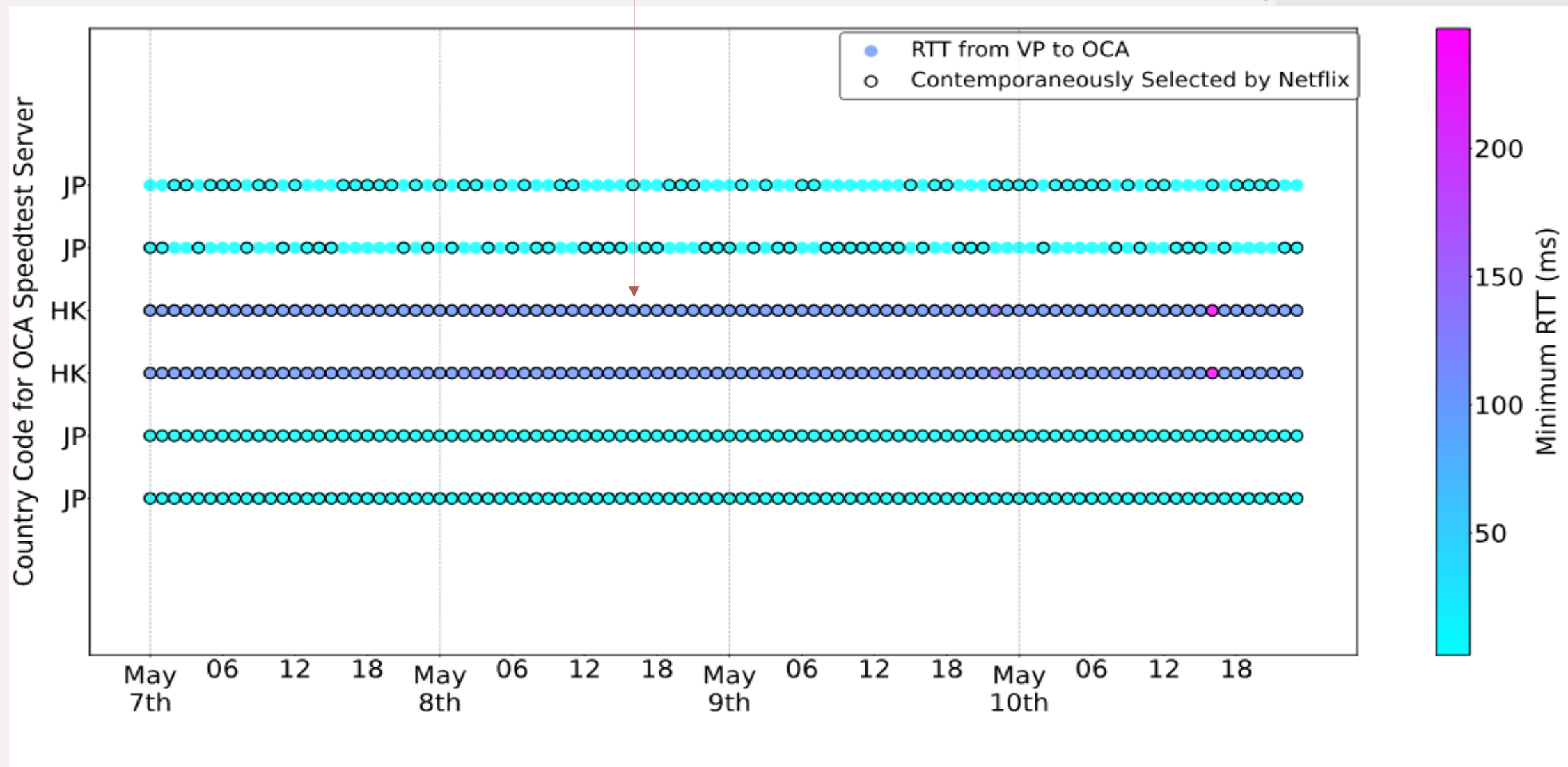
Netfli's strategy for a VP in Bhutan

Shift in Netflix's server selection strategy when high latency is reported from certain OCAs



Netflix's strategy for a VP in Japan

The list of servers returned by Netflix are not always the closest



Netflix's strategy for a VP in India

Server selection strategy balanced load between servers from Bharti Airtel Ltd. (AS 9498) and Bharat Sanchar Nigam Ltd. (AS 9829)

