Operationalizing Yarrp: High-Speed Active Network Topology Mapping from AWS

https://yarrp.nps.tancad.net/

Justin P. Rohrer (jprohrer@nps.edu)

Department of Computer Science

US Naval Postgraduate School

AIMS-KISMET, February 28, 2020

alternate title: How we’ve collected hourly Internet topology snapshots for the last 6 months*

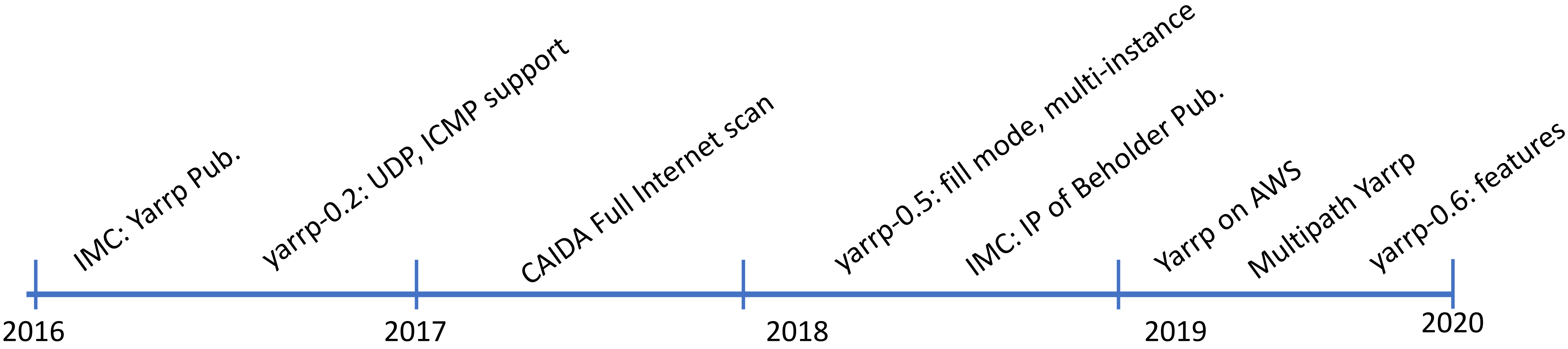
* Except for the month where AWS shut us down

Background

Background

- Yarrp is a thing: https://www.cmand.org/yarrp/
  - Probing rates ~1M PPS
- Active Network Topology Mapping:
  - Send probes into the network from vantage points
  - Induce routers to send responses
  - Build a map of how Internet is connected and data forwarded
- Goal: create/collect Internet topology “snapshots”
  - E.g. probe al IPv4 /24s within 5 minutes
  - Compare snapshots over time
- Vantage points supporting Yarrp CPU/BW are hard to find/maintain

Major Yarrp Milestones



IMC: Yarrp Pub.

yarrp-0.2: UDP, ICMP support

CAIDA Full Internet scan

yarrp-0.5: fill mode, multi-instance

IMC: IP of Beholder Pub.

Yarrp on AWS

Multipath Yarrp

yarrp-0.6: features

2016

2017

2018

2019

2020

Deploying Yarrp in the cloud

Distributed Yarrp (Freyr)

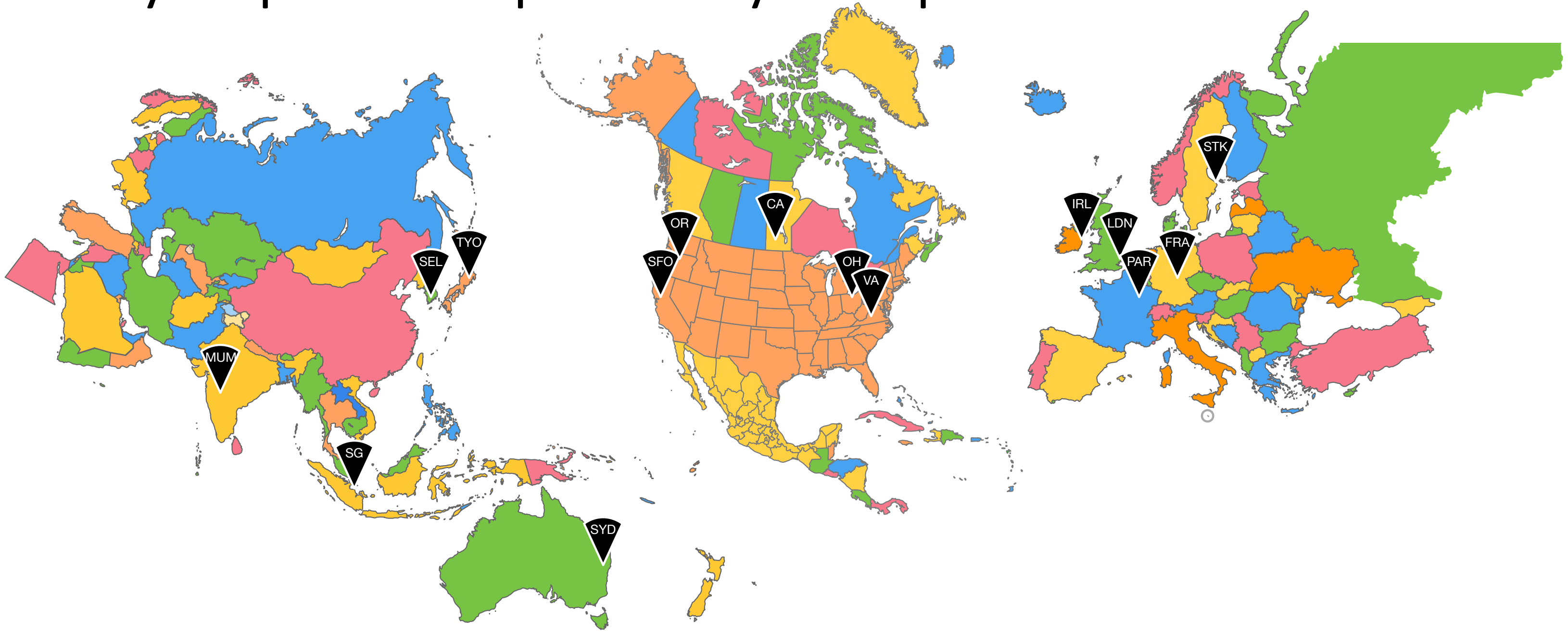
- Running Yarrp from multiple locations:
  - Provides greater discovery
  - Allows for higher aggregate rates
- Needs:
  - Deploy Yarrp at scale
  - Provide manageability and elasticity
  - Provide fault-tolerance and robustness
- Plan:
  - use AWS compute/bandwidth resources at geographically distributed vantage points

Challenges

- AWS designed to do the same job many times in one place (AZ)
    - Most services don’t support cross-region operation
- Undocumented behavior, easily overwhelmed middleboxes
    - E.g. security policy `allow ICMP from ANY` drops 90% of inbound ICMP
- All hosts NATed, even when assigned public IPs
- PTR record support extremely limited, only for SMTP servers
- IPv6 support not on par with IPv4
- No sysadmin to design/operate this
    - Needs to keep running with only sporadic attention from me
- High-bandwidth/CPU instances are expensive
- Getting data out of AWS is expensive

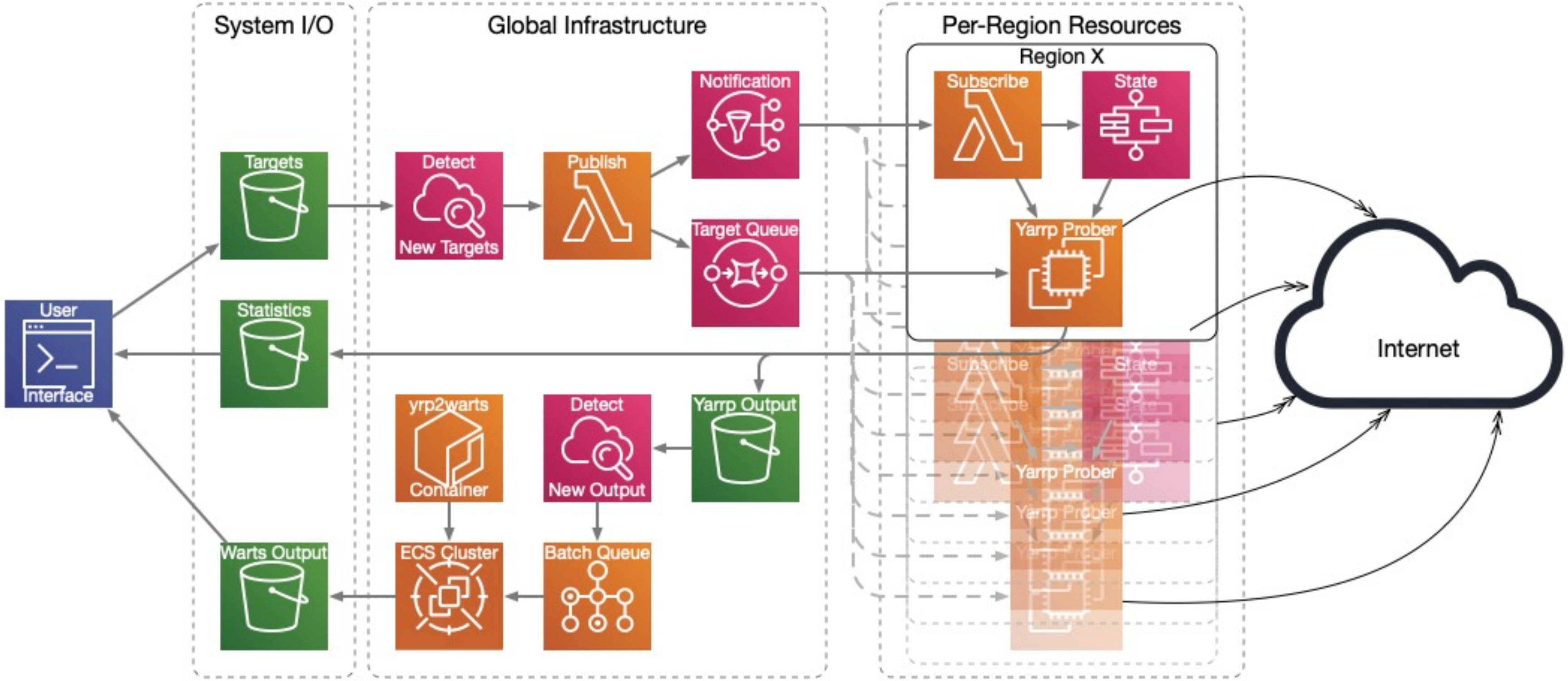
Yarrp AWS deployment scope

- Deployed to vantage points (VPs) in 15 datacenters worldwide
  - Particular measurements may use subset or all VPs
  - Targets may be distributed across VPs
    - Automatic resilience – unresponsive VP targets reassigned to responsive VPs
  - Targets may be probed in parallel by multiple VPs



Yarrp AWS deployment architecture

- Includes global (orchestration) infrastructure
  - Process & distribute targets to regions; Collect & process results
- Per-region probing resources are replicated to all data centers



Operational Status

- Probing Set 1:
  - A target address in each routed /24 of the IPv4 Internet
  - Once per hour
  - Distributed across 15 AWS regions
- Probing Set 2:
  - A target address in each routed /16 of the IPv4 Internet
  - Once per hour
  - Redundantly by all AWS regions
- Data available on request. Large downloads use “requester pays” model
- Currently running continuous production, work proceeds to improve user interface, add IPv6 support, etc.

Lessons Learned

AWS Policy Interactions

- Traceroute is not a violation of the AWS Acceptable Use Policy
    - But it could still get your account shut down
- Abuse reports only go to the root account
- The security and abuse team will never interact with users directly
    - A user must have an AWS account manager to advocate for them
- Each region has different limitations
    - E.g. don’t send packets with TTL=10 in region X

Topology Observations

- There are 10-12 (region dependent) hops between ec2 and Internet
    - Mostly in 100.64.0.0/10 shared address space (RFC6598)
- Comparing snapshots is hard due to prevalence of load-balancing
- Load-balancing analysis using MDA-Yarrp (shameless plug): https://rbeverly.net/research/papers/dminer-nsdi20.html
    - 65% of paths have load-balancing
    - Significant load-balancing between ASes
    - Observed diamonds with 100s of nodes and 1000s of edges
    - Flows rebalanced periodically (order of hrs)

Collaboration Goals

- Share the data
  - AWS S3 requester-pays model
- Make Yarrp data queryable
  - Via AWS Athena (BigTable equivalent)
  - Support multipath (primitive type can't be traceroute)
- Feedback on usefulness of hourly snapshots
  - Or, what is the "right" snapshot frequency
- Feedback on target set permutation and goals
  - Reuse for longitudinal analysis
  - Permutation for coverage

End of slides