RPKI Propagation Emulation Measurement: an Early Report

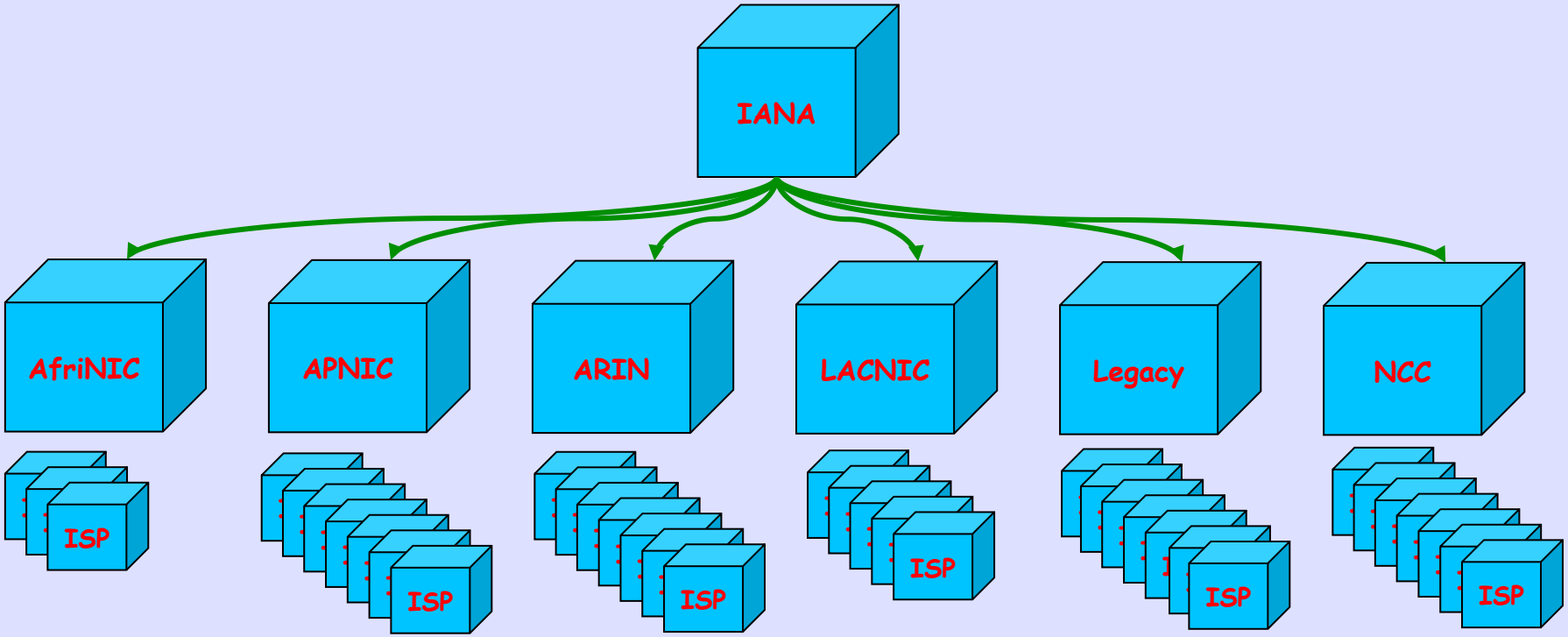
IEPG / 2012.07.29

Iain Phillips <i.w.phillips@lboro.ac.uk>
Olaf Maennel <o.m.maennel@lboro.ac.uk,>
Debbie Perouli <depe@cs.purdue.edu>
Rob Austein <sra@hactrn.net>
Cristel Pelsser <cristel@iij.ad.jp>
Keiichi Shima <keiichi@iijlab.net>
Randy Bush <randy@psg.com>

Questions

- What are the propagation characteristics of Relying Part (RP) infrastructure?
- How sensitive is propagation to inter-cache RTT?
- How sensitive is propagation to RP and cache fetch timers?
- How much is propagation and how much is validation?

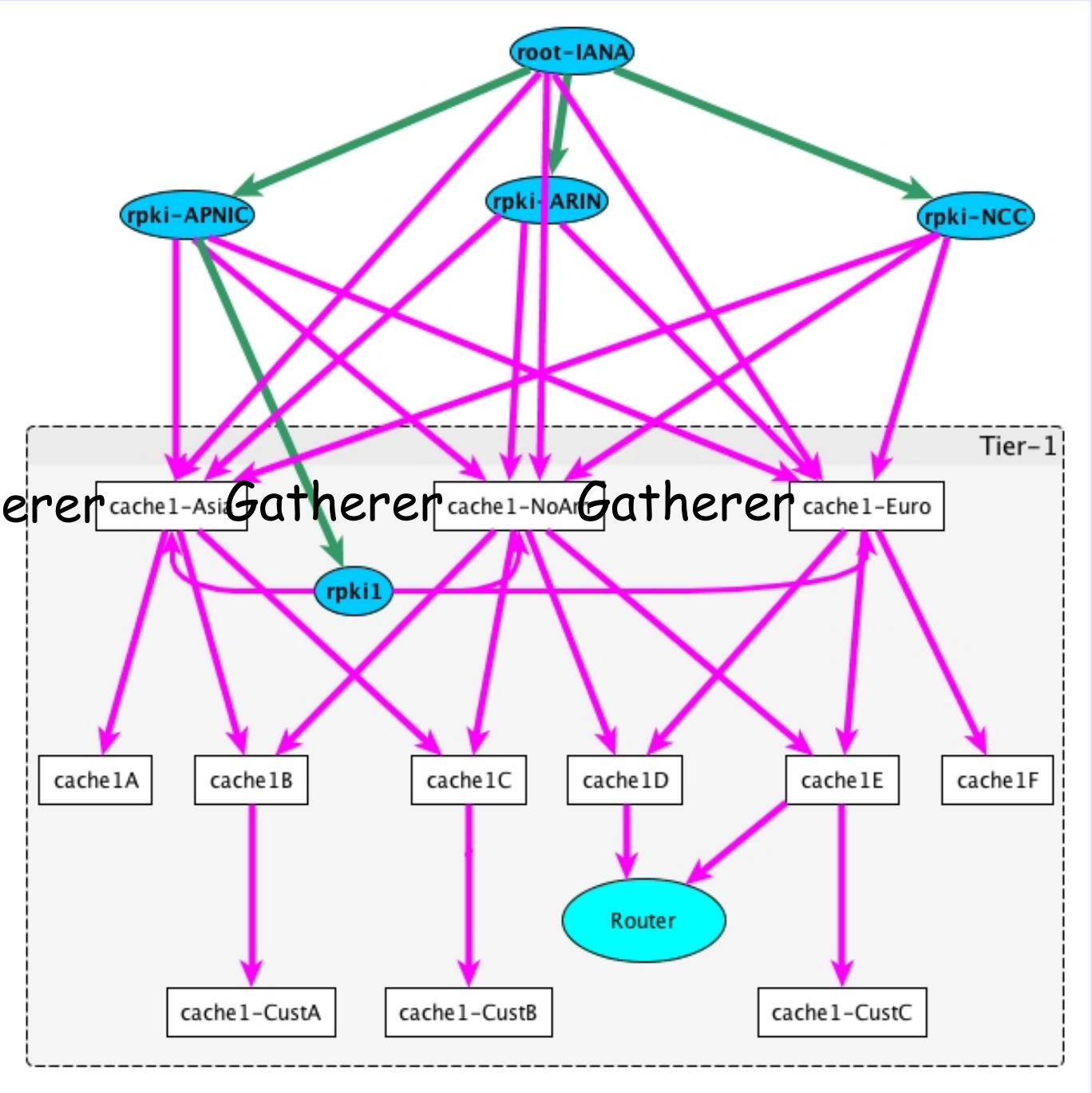
Publication Hierarchy



Not Critical as This Interest is Inter-Cache

Publication

Inter-Cache



What is Propagation?

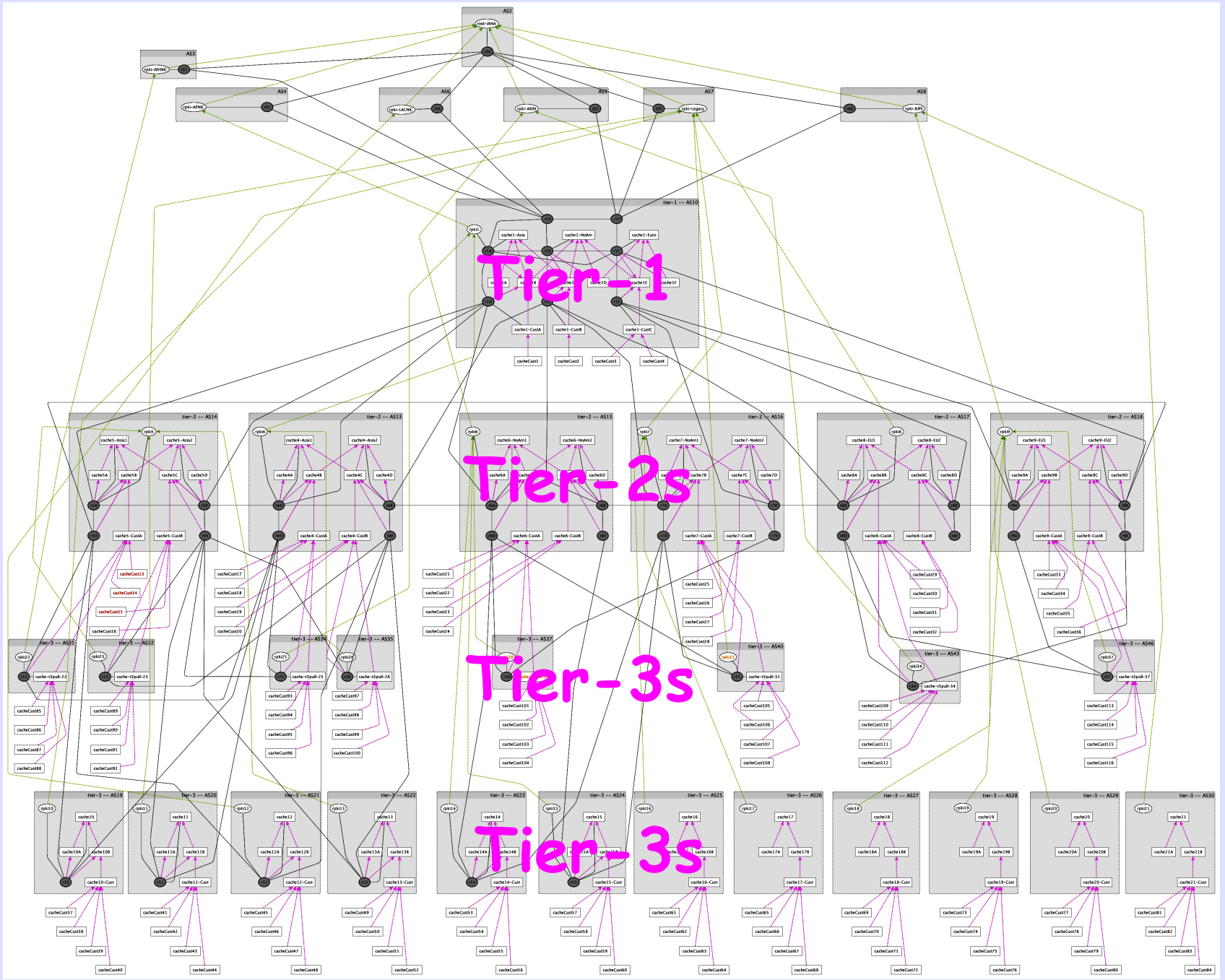
- The time from when a CA publishes an object (Cert or ROA) to when a Relying Party receives it.
- A Relying Party is a validated cache or a router via the rpki-rtr protocol.
- Measured by caches and routers logging every received object.

Architecture

- Do not care about routers, BGP, … as they do not contribute to measurement
- Use *pseudo-router*, an rpki-rtr client which logs each incoming VRP (ROA PDU)
- Caches also log receipt of objects
- Use routers to induce delay, as packets go from Japan to Texas and back

Caches

- Each cache rsyncs entire data from *parent* cache(s) or gatherers
- Each cache has a root TAL
- Every cache validates the data it has fetched


Tier-1
Tier-2s
Tier-3s
Tier-3s

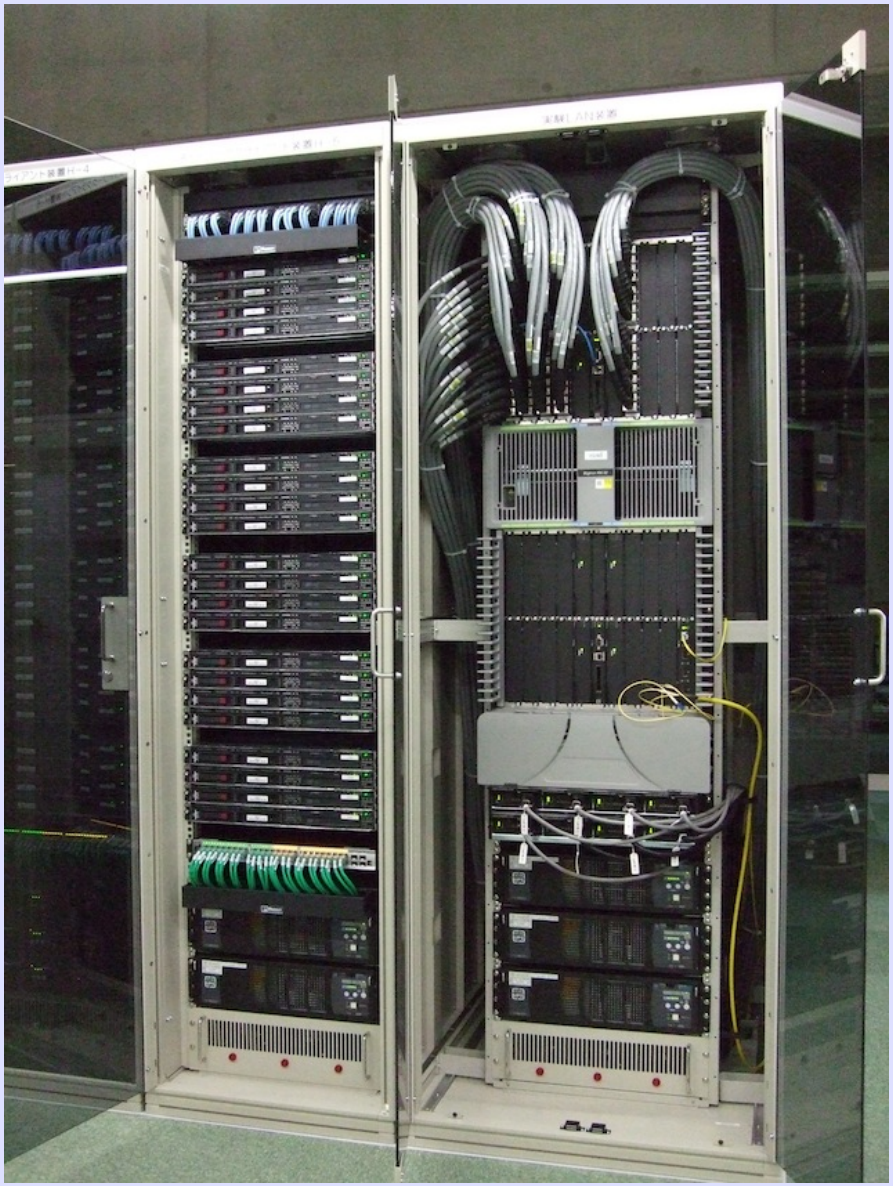
Full Testbed

- 2 Tier-1, each with 3 Gatherers
- 6 Tier-2 per Tier-1, each with 2 Gatherers
- 20 Tier-3s per Tier-1 - 12 have gatherer, 8 use upstreams' caches

	Count	Gatherers	Caches	CAs
Tier-1	2	2x3=6	2x16=32	2x1=2
Tier-2	2x6=12	2x18=36	18x12=216	2x6=12
Tier-3	2x20=40	2x12=24	2x8x5=80 2x12x8=196	2x20=60
Totals	**54**	**66**	**524**	**74+7=81**

How Do You Deploy a Testbed of About 250 Machines?

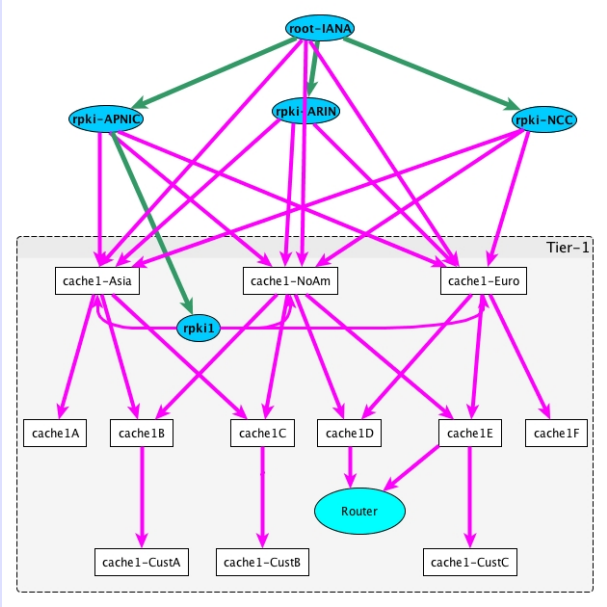
StarBED ~ 1000 KVMs



You Don't Configure 250 Servers by Hand

L'Borough AutoNetKit

- You draw this on your Mac using yEd



Yes, I am Serious

- AutoNetKit reads the graphml, Builds Server Configurations and Deploys them on StarBED, Junosphere, etc.

AutoNetKit

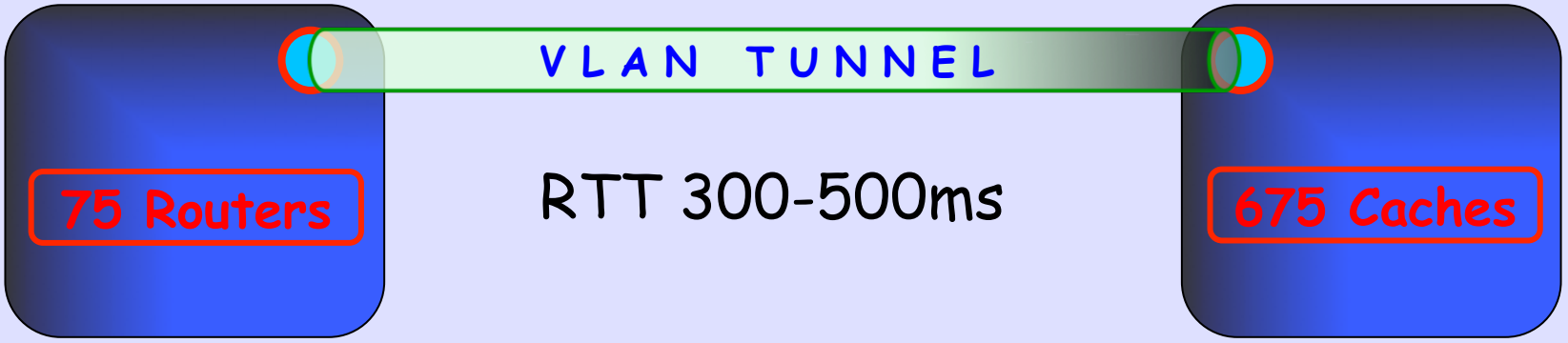
- NetKit originally Roma Tre University by Andrea Cecchetti, Lorenzo Colitti, Federico Mariani, Stefano Pettini, Flavia Picard, and Fabio Ricci
- AutoNetKit by Matt Roughan and Simon Knight at University of Adelaide
- Further Developed at University of Loughborough by Iain, Debbie, and Olaf

Enhancing AutoNetKit

- Was only routers and routing
- Address assignment was poor
- Needed to add concept of servers and services
- Needed to understand RPKI components: rpkid, pubd, caches, rtr-client, …
- Needed to handle RPKI object creation

Inducing Delay

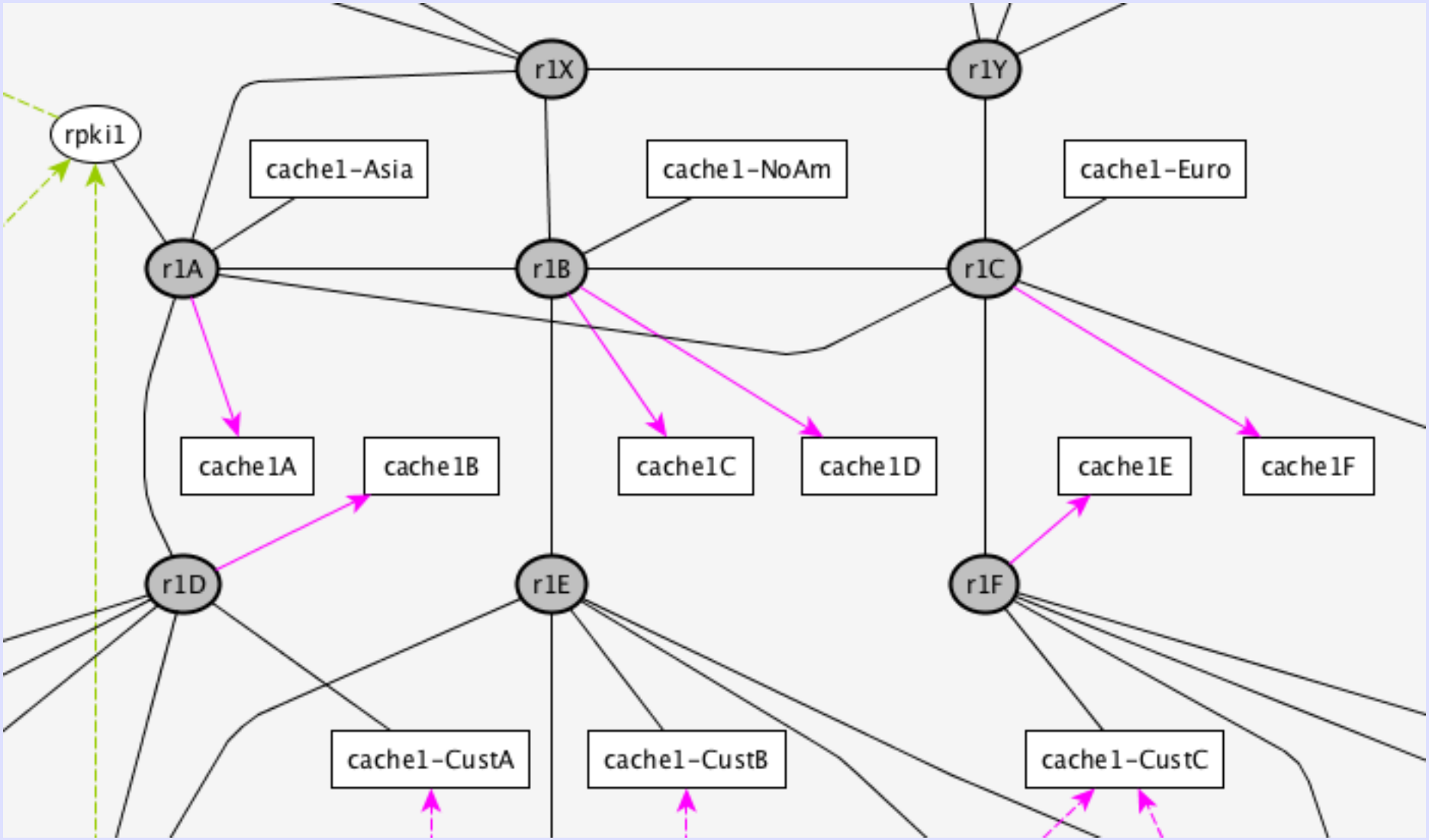
RTT to Remote Routers Induces Delay

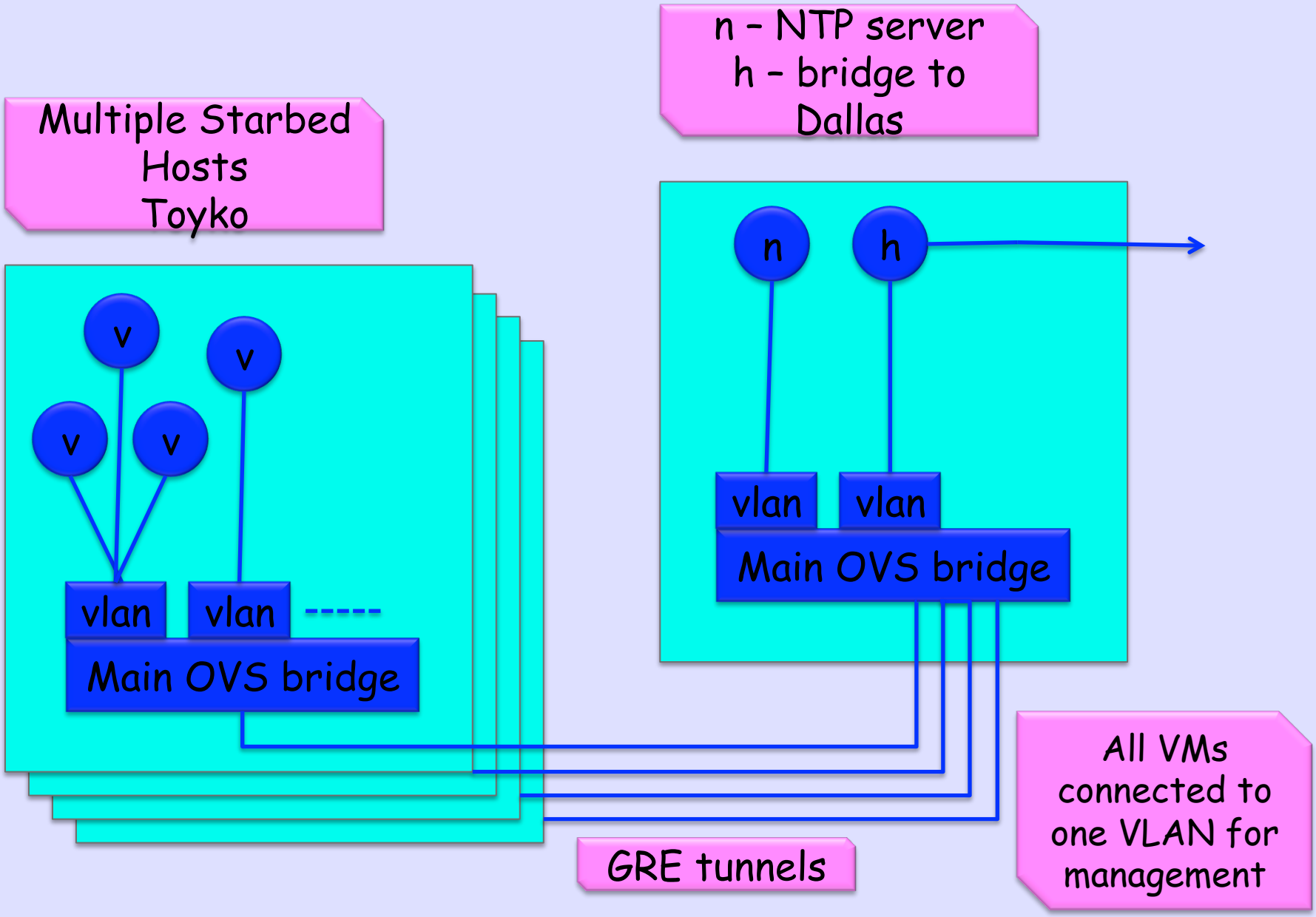


vmm.dfw
Four Server VMM
Junosphere
Dallas Texas US

StarBED
1,000 Servers
We're Using 50
Nomi-city, Ishikawa Pref

Inducing Delay




n - NTP server
h - bridge to Dallas
Multiple Starbed Hosts Toyko
n
h
v
v
v
v
vlan
vlan
Main OVS bridge
vlan
vlan
Main OVS bridge
GRE tunnels
All VMs connected to one VLAN for management

Notation for Delay

- If two pubds/caches/... are both connected to routing by a solid line, then the traffic between them is routed, i.e. goes StarBED to Junosphere back to StarBED, inducing a very large delay.
- Sequential router hops stay within Junosphere/Dallas, so do not add significantly more delay

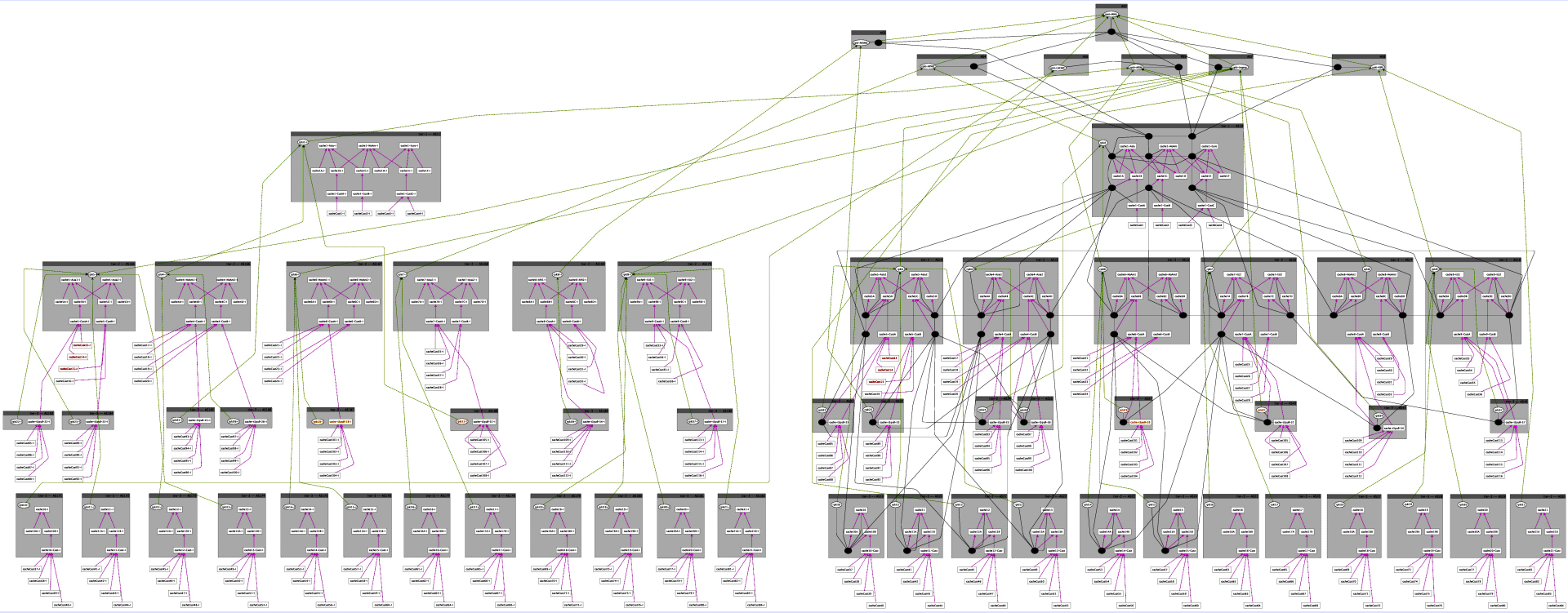
Creating Objects

We will buy a two or three star dinner for the code to take a real BGP table dump, Route Views or whatever, and create a hierarchy of well aggregated certificate requests and subsequent ROAs.

Creating Objects

- 1,500 ROAs on Start
  - 250-270 per RIR for ISPs who use RIR web pages
  - 45 per Tier-1 ISP
  - 10 per Tier-2 ISP
  - 1-2 per Tier3 ISP
- 1,500 more fed slowly during a run
- Using Same Distribution

Two Tier-1 Model

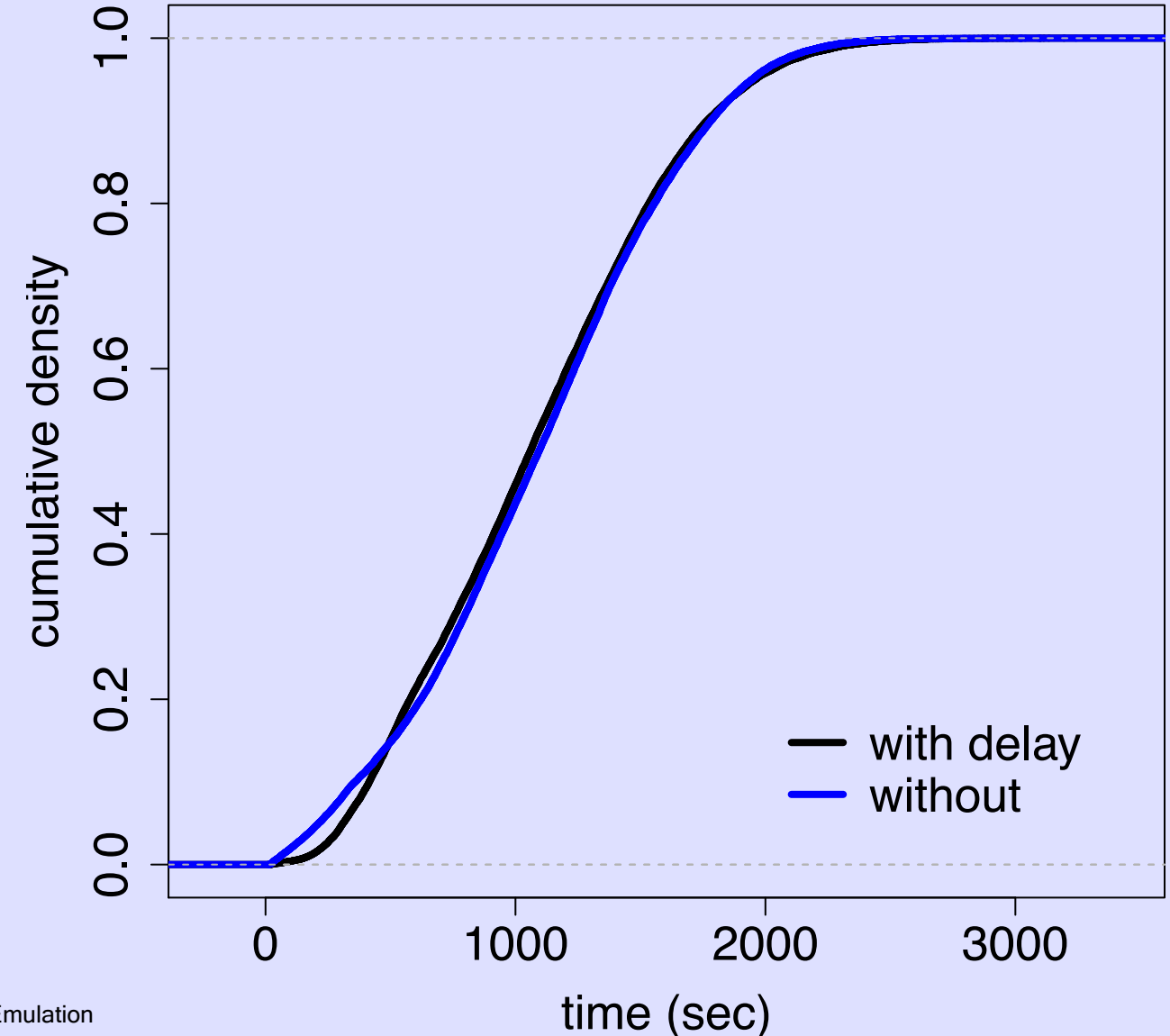


One Without Delay One With

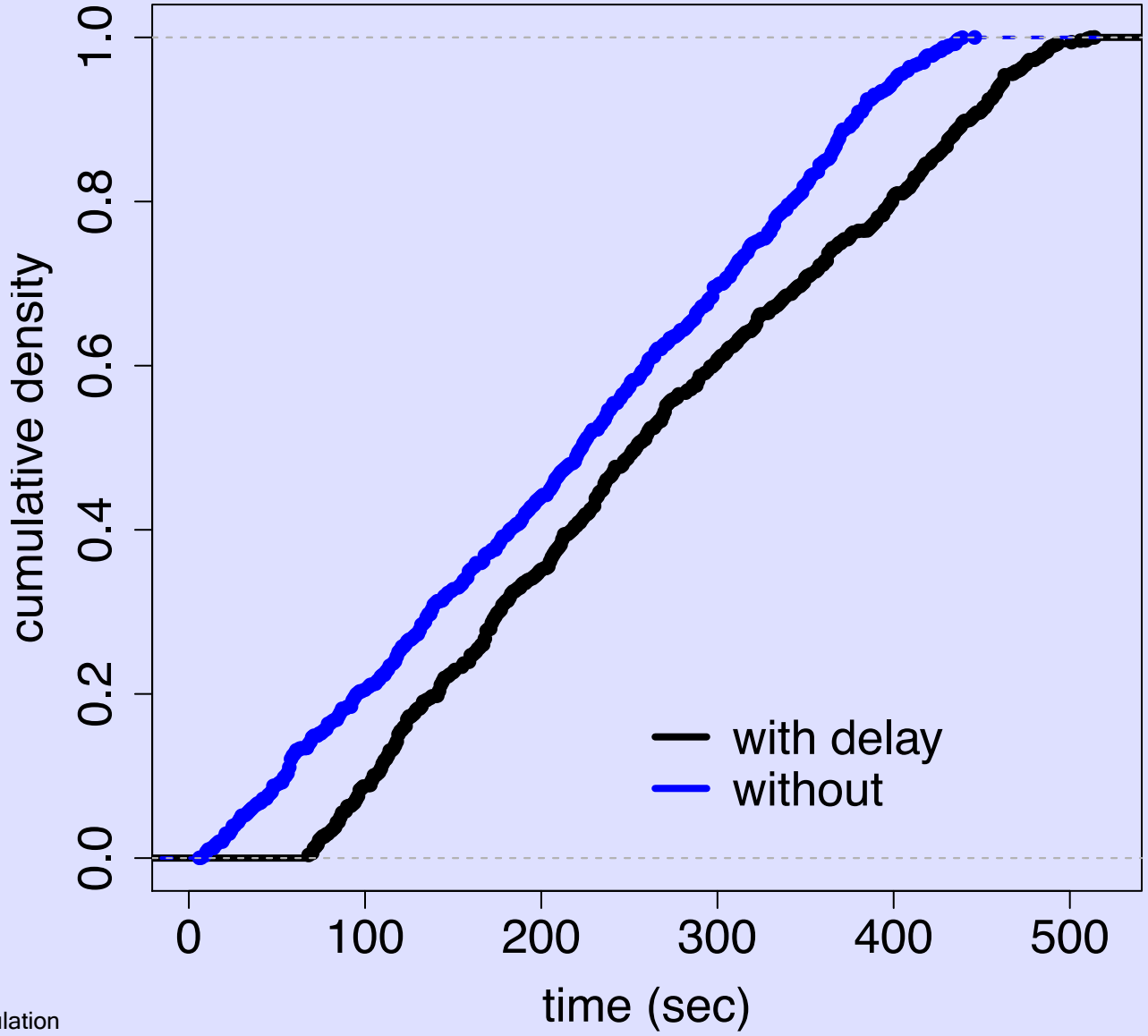
Running the Model

- About one hour to run and upload to StarBed in Japan from Loughborough
- 150MB Uploaded
- 1:1 Time Ratio, so it runs for a full day
- Produces 1-3.3G of log files
- Which we then have to transfer the logs to a compute server
- Analysis of logs takes 42 minutes

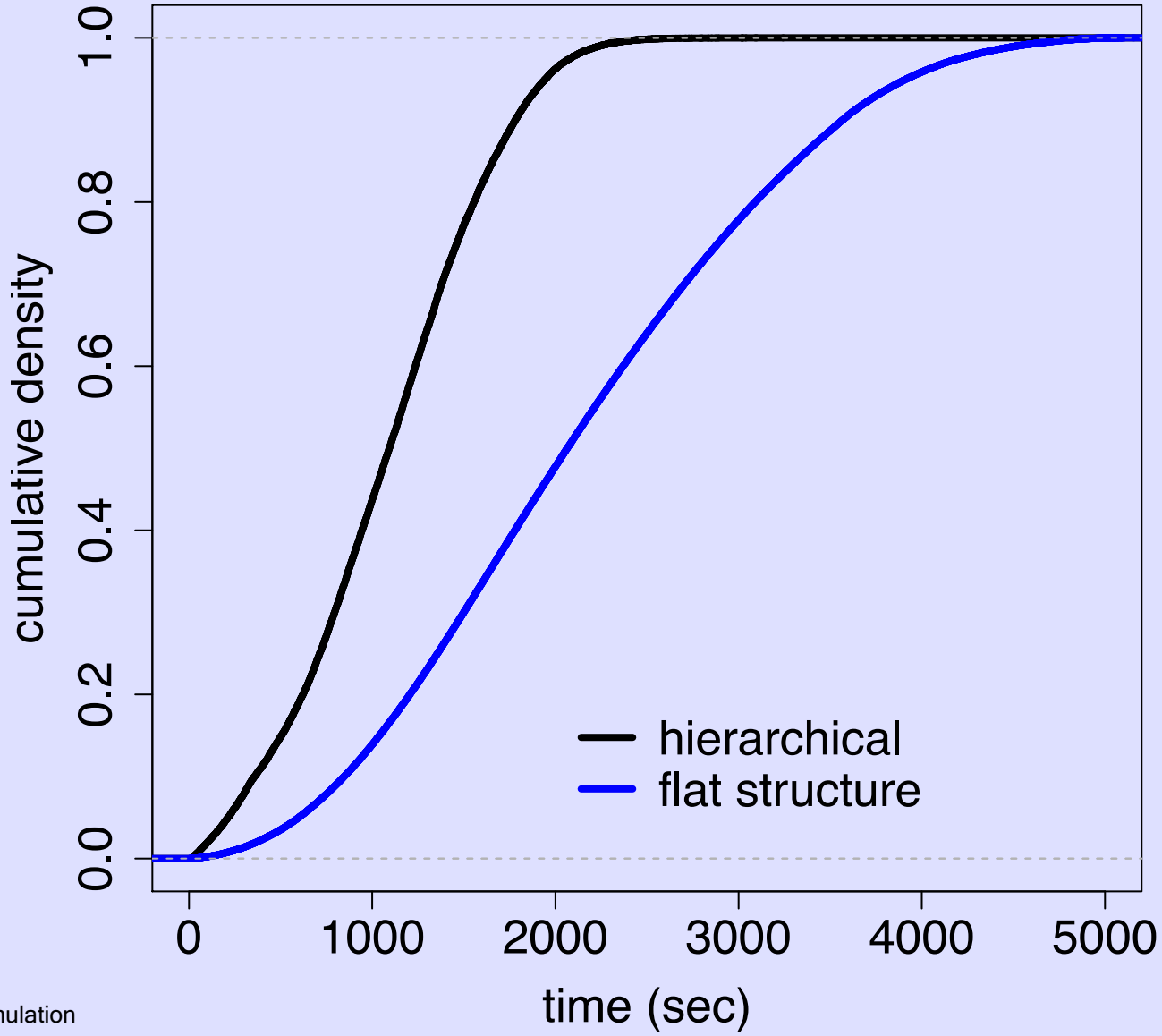
RIRs to All Caches



RIPE to Gatherers



Flat vs Hierarchic



Thanks

- StarBED
- Juniper and Cisco
- DHS [0]
- University of Adelaide
- Loughborough University, Purdue, & IIJ

[0] THIS WORK IS SPONSORED IN PART BY THE DEPARTMENT OF HOMELAND SECURITY UNDER AN INTERAGENCY AGREEMENT WITH THE AIR FORCE RESEARCH LABORATORY (AFRL).