



UNIVERSITY
OF AMSTERDAM

docker



Docker Overlay Networks

Performance analysis in high-latency environments

Students: Siem Hermans
Patrick de Niet

Supervisor: Dr. Paola Grosso



Research Project 1
System and Network Engineering

Research question

“What is the performance of various Docker overlay solutions when implemented in high latency environments and more specifically in the GÉANT Testbeds Services (GTS)?”

Related Work

Internal

- Claassen, J. (2015, July). Container Network Solutions. Retrieved January 31, 2016, from <http://rp.delaat.net/2014-2015/p45/report.pdf>.
- Rohprimardho, A. (2015, August). Measuring The Impact of Docker on Network I/O Performance. Retrieved January 31, 2016, from <http://rp.delaat.net/2014-2015/p92/report.pdf>.

External

- Kratzke, N. (2015). About Microservices, Containers and their Underestimated Impact on Network Performance. CLOUD COMPUTING 2015, 180.
- Barker, S. K., & Shenoy, P. (2010, February). Empirical evaluation of latency-sensitive application performance in the cloud. In Proceedings of the first annual ACM SIGMM conference on Multimedia systems (pp. 35-46). ACM.

Docker - Concepts

Basics

- Containerization
- Gaining traction
- Performance increases

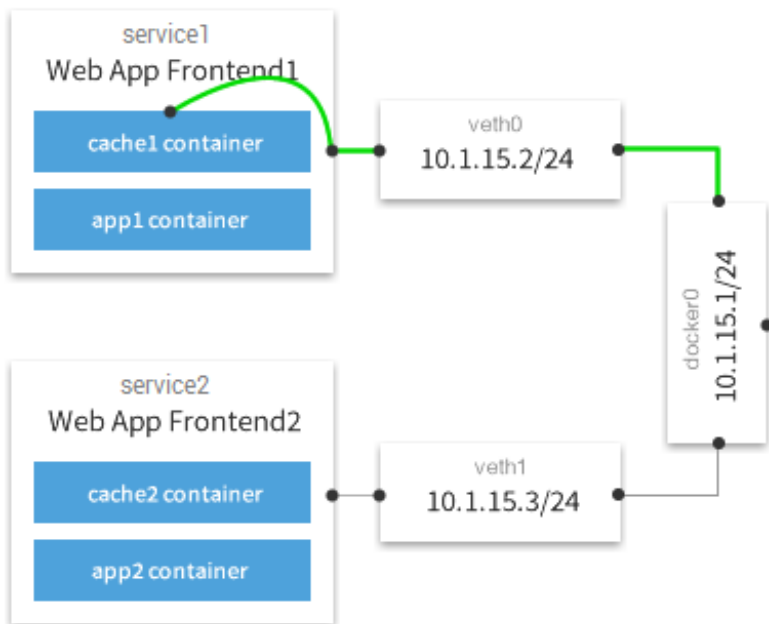
-
- Role of Docker

Virtual Machine

Container

Multi-host networking

- Virtual networks that span underlying hosts
- Powered by `libnetwork`



Overlay solutions

Libnetwork

(Native overlay driver)



- Based on SocketPlane
- Integrating OVS APIs in Docker
- VXLAN based forwarding

Weave Net



- Previously routing based on `pcap`. Now uses OVS.
- Libnetwork plugin
- VXLAN based forwarding

Kratzke, N. (2015).

Flannel



- Flanneld agent
- No integration with `libnetwork`
- Subnet per host
- UDP or VXLAN forwarding

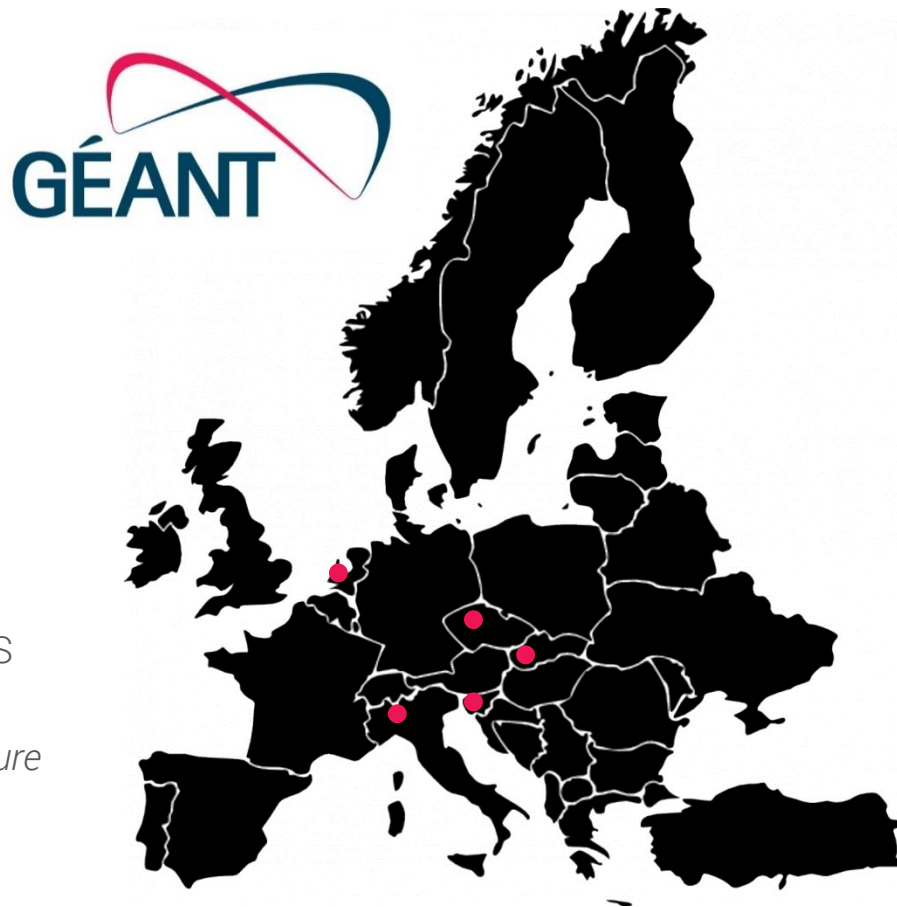
Project Calico



- Technically not an overlay
- Routing via BGP
- Segmentation via iptables
- State distribution via BGP route reflectors
- No tunneling

GÉANT - Introduction

- European research community
 - Amsterdam
 - Bratislava
 - Ljubljana
 - Milan
 - Prague
- GÉANT Testbeds Service (GTS)
- OpenStack platform, interconnected by MPLS
- KVM for compute nodes
- Resembles IaaS providers; *Shared infrastructure*

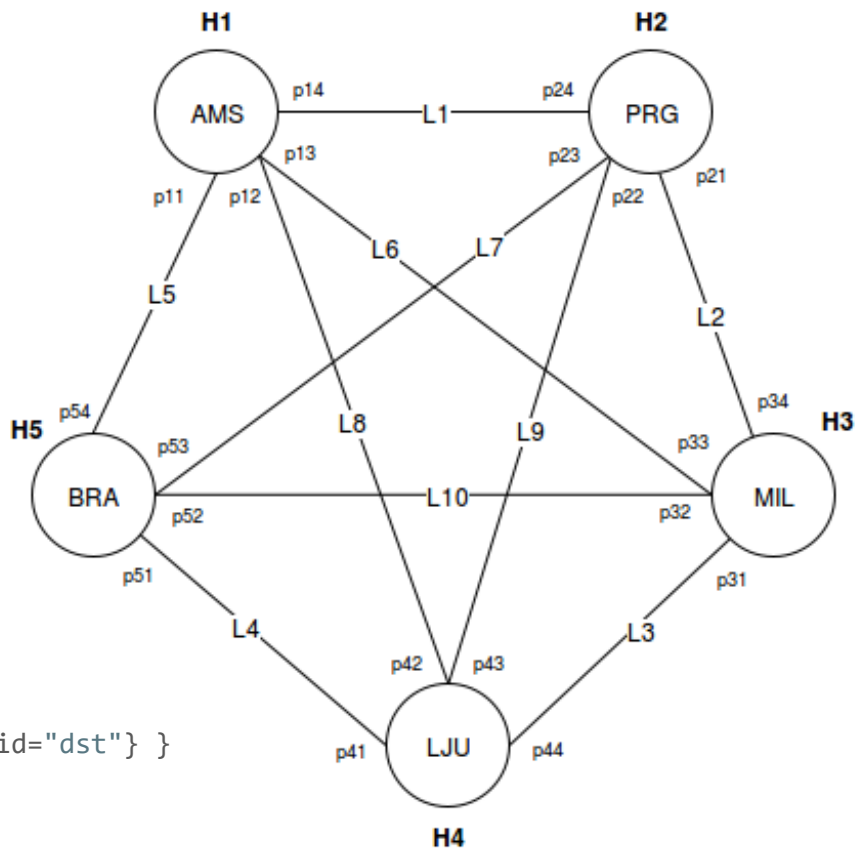


Topologies (1)

- Four full mesh instances
 - DSL 2.0 grammar (JSON)
- Local site; Feasibility evaluation

DSL

```
FullMesh {
  id="FullMesh_Dispersed"
  host { id= "h1" location= "AMS"
    port { id="port11"}
    port { id="port12"} }
  link { id="l1" port {id="src"} port {id="dst"} }
  adjacency h1.port14, l1.src
  adjacency h2.port24, l1.dst
} {...}
```

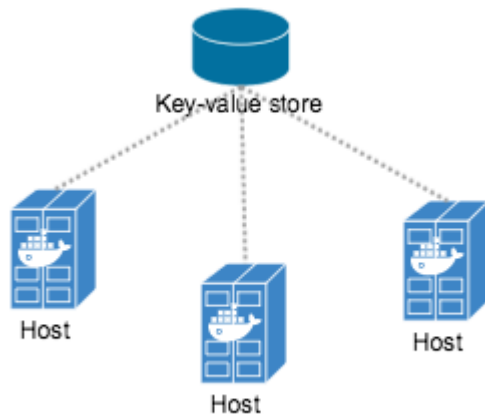


Topologies (2)

- Scaling up from single-site feasibility check
 - Calico dropped
- Full mesh divided in:
 1. **Point-to-point**, synthetic benchmarks
 2. **Star topology**, real-world scenario

Setup

- Flannel VXLAN tunneling
- Key-value store placement
 - Storing network state
 - Separate distributed system



Methodology - Performance

Synthetic benchmark (PtP)

- Placement of nodes

Netperf

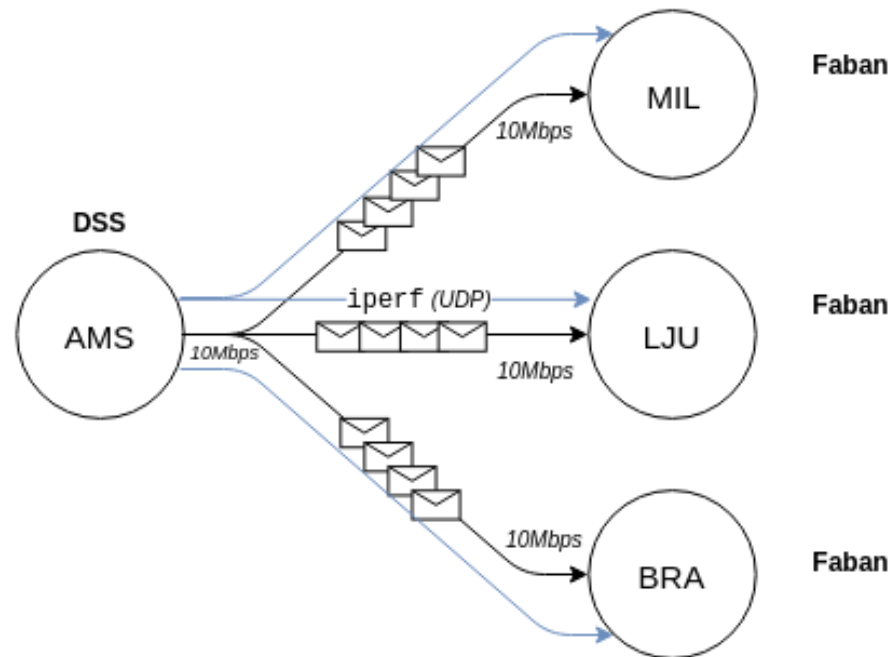
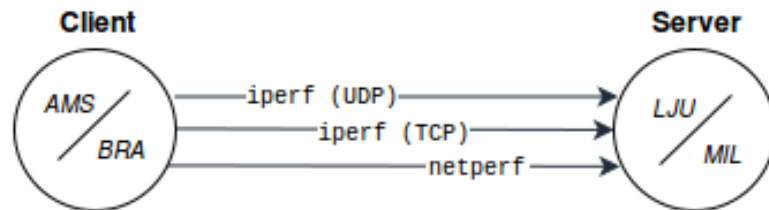
- Latency
- Jitter

Iperf

- TCP/UDP throughput
- Jitter

Latency sensitive application (Media streaming)

- Darwin Streaming Server, Faban RTSP clients
 - Jitter (with netperf)
 - Bitrate



Results - GÉANT



Documentation



Provisioning



Access



Setup



VPN

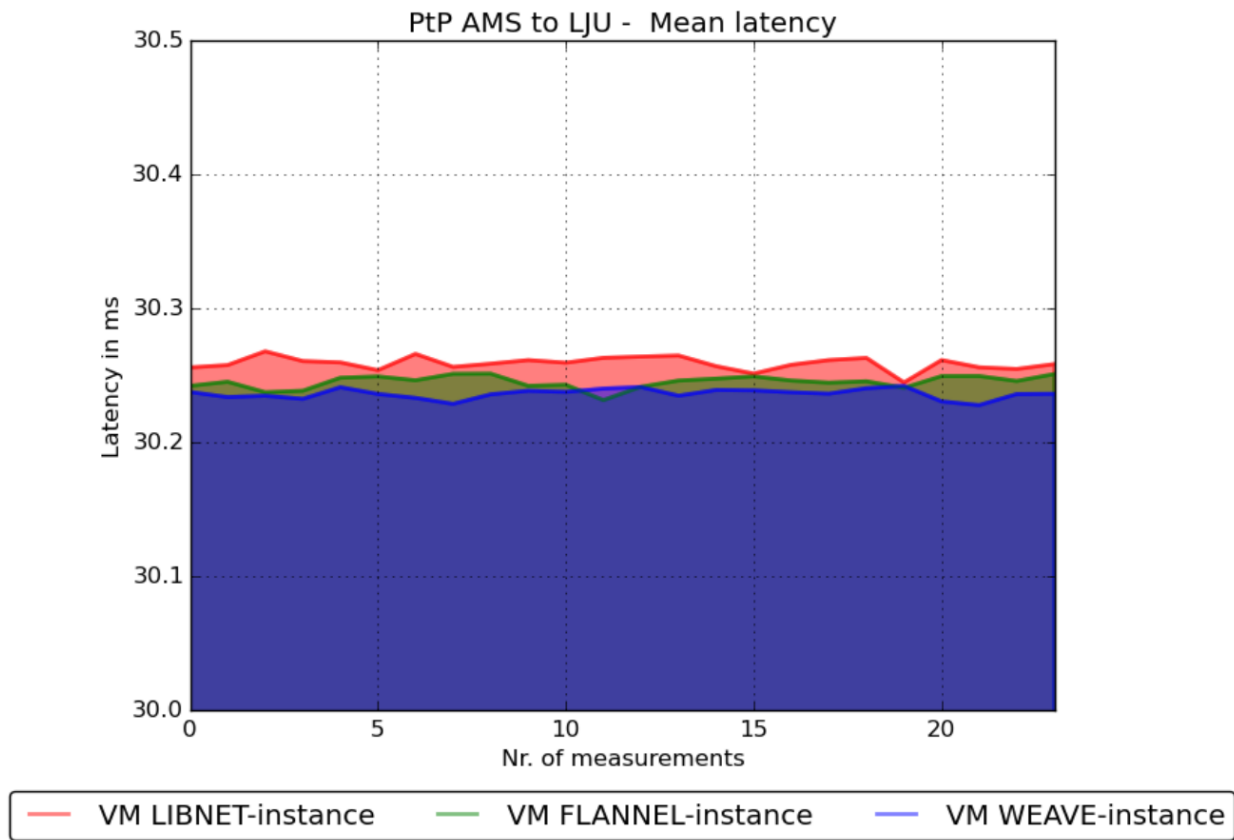


Resources



Support

Results - PtP VM to VM Latency

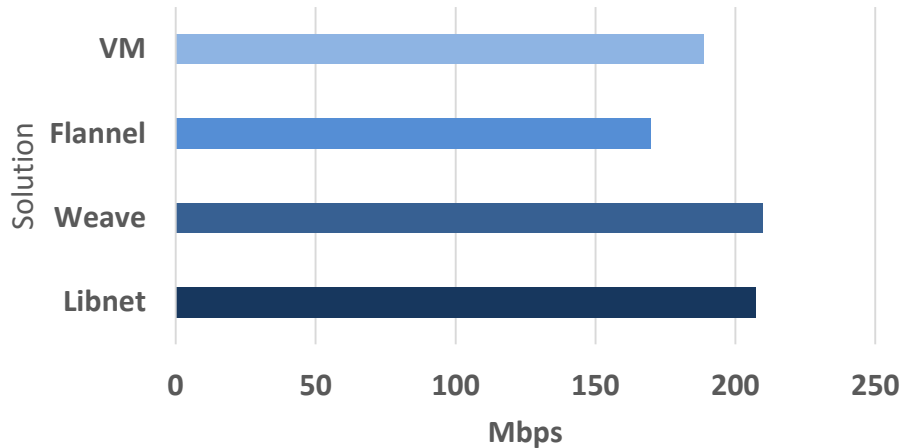


Results - PtP Docker to Docker Latency

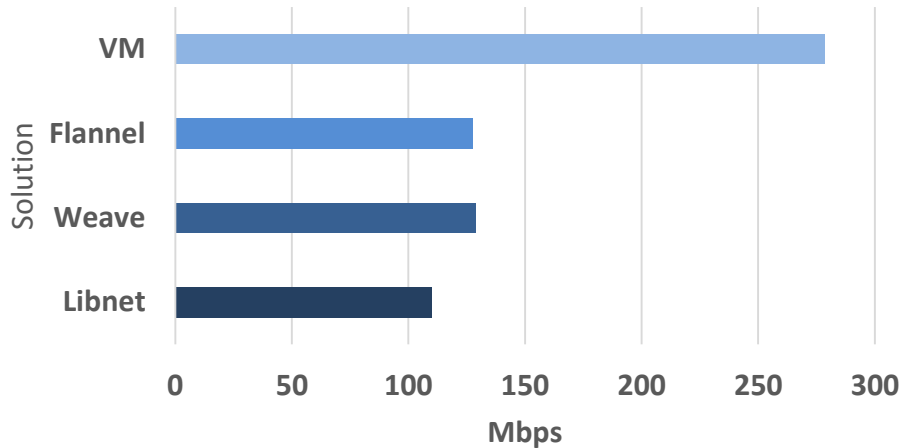
Circuit	Topology	In Milliseconds (ms)		
		Min. Latency	Mean Latency	99 th % Latency
AMS – MIL	LIBNET	36.3	36.5	37.0
	WEAVE	36.2	36.5	37.0
	FLANNEL	42.5	42.9	43.0
AMS – LJU	LIBNET	30.1	30.3	31.0
	WEAVE	29.8	30.3	31.0
	FLANNEL	29.8	30.3	31.0
AMS – BRA	LIBNET	17.6	17.7	18.0
	WEAVE	17.4	17.7	18.0
	FLANNEL	17.4	17.7	18.0
MIL – LJU	LIBNET	61.8	62.1	62.4
	WEAVE	59.6	59.8	60.0
	FLANNEL	55.6	55.8	56.0
MIL – BRA	LIBNET	12.7	13.0	14.0
	WEAVE	12.9	13.1	14.0
	FLANNEL	12.9	13.1	14.0
BRA – LJU	LIBNET	47.1	47.4	48.0
	WEAVE	43.1	59.5	130.0
	FLANNEL	43.1	43.4	44.0

Results - PtP Throughput

AMS to BRA TCP Throughput

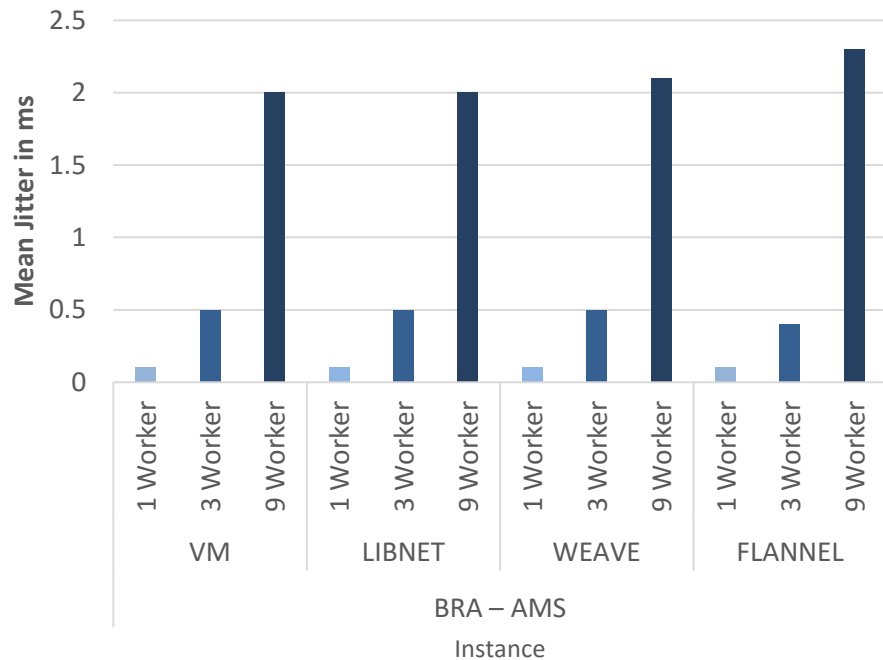


AMS to BRA UDP Throughput

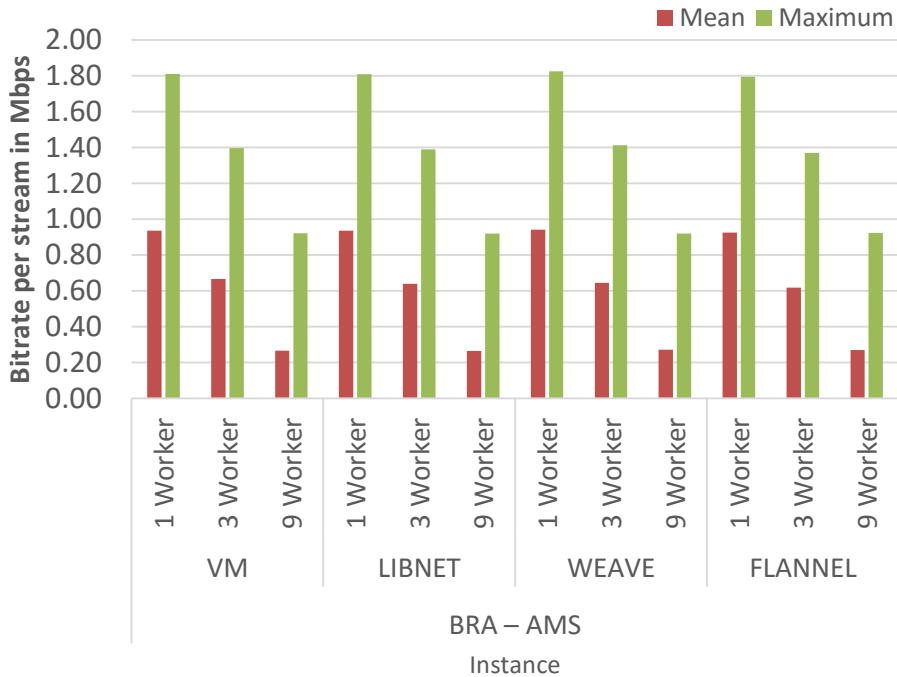


Results - Streaming Experiment

BRA - AMS Concurrency Jitter



BRA - AMS Concurrency Bitrate



Conclusion & Future Work

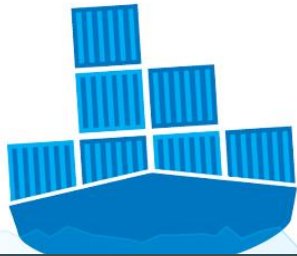
-
- Measurements currently only valid within GTS environment;
 - Reconduct performance analysis in heavily shared environment (e.g. *Amazon EC2*)
 - Perform experiments with more compute resources (*CPU capping*)
 - Anomalies in throughput performance not identified (*UDP, TCP*)
 - Similar behavior discovered in the work of J. Claassen
 - Ideally more measurements to increase accuracy
 - No significant performance degradations by implementing Docker overlays within GTS
 - Use Weave ideally within the GTS environment



UNIVERSITY
OF AMSTERDAM

Research Project 1
System and Network Engineering

Questions ?



Thank you

A: Science Park 904, Amsterdam NH
W: rp.delaat.net



github.com/siemhermans/gtsp perf

siem.hermans@os3.nl
patrick.deniet@os3.nl

