Self-* Networks

Stefan Schmid

Internet Architecture and Management (INET)



Formerly: Anja Feldmann

Research Vision

- Our vision is that *networked systems* should become self-*: i.e., self-optimizing, self-repairing, self-configuring
 - Why? Networks are a *critical infrastructure* but complex many outages due to *human errors*
- Topics:
 - algorithms and mechanisms to design and operate communication networks
 - network architectures and protocols for future communication technologies
 - *performance evaluation* of networked and distributed systems
 - network security
 - software-defined networks, network virtualization, optical networks, wireless and cellular networks, peerto-peer networks, cryptocurrency networks
- Accordingly, we are currently particularly interested in *automated and data-driven* approaches

Collection of data: e.g., datacenter traffic Metrics, e.g., information-theoretic measures of spatial and temporal structure, also visualization Algorithm and protocol design, formal analysis



System design and evaluation

Collection of data: e.g., datacenter traffic Metrics, e.g., information-theoretic measures of spatial and temporal structure, also visualization

Algorithm and protocol design, formal analysis

System design and evaluation



DC Traces

The following tables list the datacenter traces used in the publication: On the Complexity of Traffic Traces and Implications To reference this website, please use: bibtex

File Name	Source Information	Туре	Lines	Size	Direct Download	Torrent Download
hpc_exact_boxlib_multigrid_c_large.csv	High Performance Computing Traces	Traces	17.947.800	151.3 MB	Download	Download
hpc_exact_boxlib_cns_nospec_large.csv	High Performance Computing Traces	Traces	1.108.068	9.3 MB	Download	Download
hpc_cesar_nekbone.csv	High Performance Computing Traces	Traces	21.745.229	184.0 MB	Download	Download
hpc_cesar_mocfe.csv	High Performance Computing Traces	Traces	2.713.600	22.0 MB	Download	Download

Several TB of data

Collection of data: e.g., datacenter traffic Metrics, e.g., information-theoretic measures of spatial and temporal structure, also visualization

Algorithm and protocol

design, formal analysis

System design and evaluation



Collection of data: e.g., datacenter traffic Metrics, e.g., information-theoretic measures of spatial and temporal structure, also visualization

Algorithm and protocol design, formal analysis

System design and evaluation

E.g., ERC project (in a second...)

Collection of data: e.g., datacenter traffic Metrics, e.g., information-theoretic measures of spatial and temporal structure, also visualization



Algorithm and protocol design, formal analysis





Self-adjusting datacenter rack (EXIST startup, load-balancer for Spotify, etc.)

Example 1: ERC Project AdjustNet

Explosive growth...

- Increasing popularity of datacentric applications related to health, business, science, social networking etc.
- Leads to *much traffic*, especially *to*, *from*, and *inside* datacenters
- Interconnecting network is critical infrastructure for application performance

Aggregate server traffic in **Google's datacenter fleet**



Source: Jupiter Rising. SIGCOMM 2015.

... but much structure!

"less than 1% of the rack pairs account for 80% of the total traffic"

"only a few ToR switches are hot and most of their traffic goes to a few other ToRs"

"over 90% bytes in elephant flows"

ProjecToR @ SIGCOMM 2016 Understanding Data Center Traffic Characteristics @ WREN 2009



Traffic matrix of two different distributed ML applications (GPU-to-GPU):



Two different ways to generate *same traffic matrix* (same non-temporal structure)

Complexity Map



Today's datacenters

- Traditional networks are optimized for the "worst-case" (all-to-all communication traffic)
- Example, fat-tree topologies: provide full bisection bandwidth





Complexity Map



Our goal: Design self-adjusting networks which leverage *both* dimensions of structure!





Enabler: Reconfigurable Optical Technologies



- Even the **physical** layer becomes reconfigurable
- E.g., free-space optics, 60GHz, optical circuit switches

Enabler: Reconfigurable Optical Technologies



- Even the **physical** layer becomes reconfigurable
- E.g., free-space optics, 60GHz, optical circuit switches

Example 2: WWTF Project WhatIf











Routers and switches store list of forwarding rules, and conditional failover rules.





Sysadmin responsible for:

• **Reachability:** Can traffic from ingress port A reach egress port B?



Sysadmin responsible for:

- **Reachability:** Can traffic from ingress port A reach egress port B?
- **Loop-freedom:** Are the routes implied by the forwarding rules loop-free?



Sysadmin responsible for:

- **Reachability:** Can traffic from ingress port A reach egress port B?
- **Loop-freedom:** Are the routes implied by the forwarding rules loop-free?
- **Policy:** Is it ensured that traffic from A to B never goes via C?



Sysadmin responsible for:

- **Reachability:** Can traffic from ingress port A reach egress port B?
- **Loop-freedom:** Are the routes implied by the forwarding rules loop-free?
- **Policy:** Is it ensured that traffic from A to B never goes via C?
- Waypoint enforcement: Is it ensured that traffic from A to B is always routed via a node C?



Sysadmin responsible for:

- **Reachability:** Can traffic from ingress port A reach egress port B?
- **Loop-freedom:** Are the routes implied by the forwarding rules loop-free?
- **Policy:** Is it ensured that traffic from A to B never goes via C?
- Waypoint enforcement: Is it ensured that traffic from A to B is always routed via a node C?

... and everything even under multiple failures?!

Can we automate such tests or even self-repair?

Can we automate such tests or even self-repair?



Yes! Automated What-if Analysis Tool for MPLS and SR in *polynomial time*.

Leveraging Automata-Theoretic Approach



Pushdown Automaton (PDA) and Prefix Rewriting Systems Theory

MPLS configurations, Segment Routing etc.

Tool and Query Language

Part 1: Parses query and constructs Push-Down System (PDS)



• In Python 3

Part 2: Reachability analysis of constructed PDS

• Using *Moped* tool

Regular query language



query processing flow

Example: Traversal Testing With 2 Failures

Traversal test with k=2: Can traffic starting with [] go through s5, under up to k=2 failures?



Thanks. Questions?