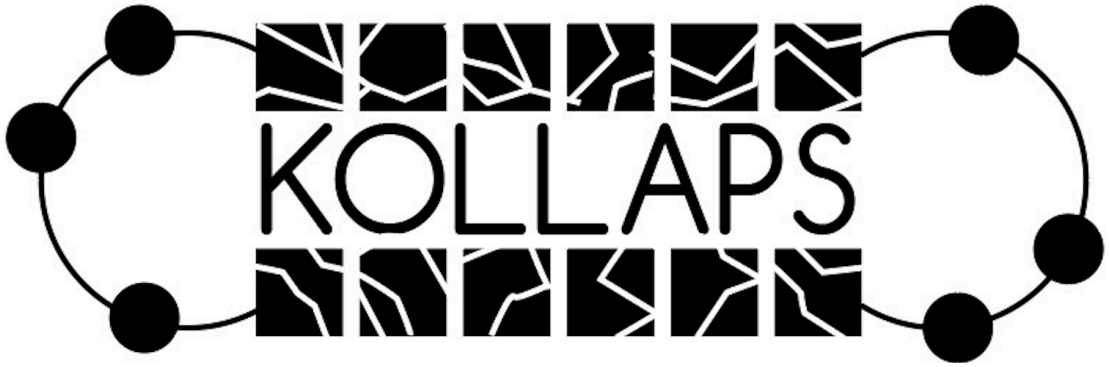


Decentralized and Dynamic Topology Emulation

Paulo Gouveia*, João Neves*, Carlos Segarra†, Luca Liechti†
Shady Issa*, Valerio Schiavoni†, and **Miguel Matos***

*: INESC-ID & IST, University of Lisbon, Portugal
†: University of Neuchâtel, Switzerland

Amazon Found Every 100ms of Latency Cost them 1% in Sales

Nov 10, 2016, 11:43am EST

Why Brands Are Fighting Over Milliseconds

Video news

Buffering reduces video watch time by ~40%, according to research

September 14, 2016 (4 years ago)

Post Mortem: What Yesterday's Network Outage Looked Like

Zalando saw a 0.7% increase in revenue when they shaved 100ms off their load time.

MOTIVATION

- Performance depends heavily on underlying network
- **Variability** and **Failures** are the norm

MOTIVATION

- Performance depends heavily on underlying network
- **Variability** and **Failures** are the norm

- Need for tools for systematic evaluation of distributed applications
- Ability to answer key questions:
  - What is the impact of halving the network latency in application throughput?
  - What is the effect of packet loss?
  - What if …

RELATED WORK

Name	Year	Mode	HW ind.	Orchestration	Concurrent deployments	Path congestion	Link-Level emulation capabilities				Any Language	Topology dynamics	Unit
							Bandwidth	Delay	Packet loss	Jitter			
DelayLine [47]	1994	User	✓	Centralized	✗	✗	✗	✓	✓	✗	✓	✗	P
ModelNet [81]	2002	Kernel	✓	Centralized	✗	✓	✓↯	✓↯	✓↯	✗	✓	✓	P
Nist NET [33]	2003	Kernel	✓	Centralized	✗	✗	✓↯	✓↯	✓↯	✓↯	✓	✗	P
NetEm [45]	2005	Kernel	✓	*(N/A: single link emulation only)*			✗	✓	✓	✓	✓	✗	P
Trickle [39]	2005	User	✓	*(N/A: single link emulation only)*			✓	✓	✗	✗	✓	✗	P
EmuSocket [23]	2006	User	✓	*(N/A: single link emulation only)*			✓↯	✓↯	✗	✗	✓	✗	P
ACIM/FlexLab [71]	2007	Kernel	✓	Centralized	✗	✓	✓↯	✓↯	✓↯	✓↯	✓	✓	V
NCTUns [85]	2007	Kernel	✓	Centralized	✗	✓	✓	✓	✓	✓	✓	✗	P
Emulab [46, 88]	2008	Kernel	✗	Centralized	✗	✓	✓↯	✓↯	✓↯	✗	✓	✓	V
IMUNES [70]	2008	Kernel	✗	Centralized	✗	✗	✓	✓	✓	✗	✓	✗	P
MyP2P-World [75]	2008	User	✓	Centralized	✗	✗	✓	✓	✓	✗	✗	✗	P
P2PLab [61]	2008	Kernel	✓	Centralized	✗	✗	✓	✓	✓	✗	✗	✗	P
Netkit [67]	2008	Kernel	✓	Centralized	✗	✓	✓↯	✓↯	✓↯	✗	✓	✗	V
DFS [79]	2009	User	✓	Centralized	✓	✗	✓↯	✓↯	✓	✓	✗	✓	P
Dummynet [32]	2010	Kernel	✓	Centralized	✗	✗	✓↯	✓↯	✓↯	✗	✓	✗	P
Mininet [53]	2010	Kernel	✓	Centralized	✗	✓	✓↯	✓↯	✓↯	✓↯	✓	✓	P
SliceTime [86]	2011	Kernel	✗	Centralized	✗	✓	✓	✓	✗	✗	✓	✓	V
Mininet-HiFi [44]	2012	Kernel	✓	Centralized	✗	✗	✓↯	✓↯	✓↯	✓↯	✓	✓	C
SplayNet [76]	2013	User	✓	Decentralized	✓	✓	✓	✓	✓	✗	✗	✓	P
MaxiNet [87]	2014	Kernel	✓	Centralized	✗	✓	✓↯	✓↯	✓↯	✓↯	✓	✓	P
Dockemu [80]	2015	User	✓	Centralized	✗	✗	✓	✓	✓	✓	✓	✗	C
EvalBox [77]	2015	Kernel	✓	Centralized	✗	✗	✓↯	✓↯	✓↯	✓↯	✓	✓	P
ContainerNet [65]	2016	Kernel	✓	Centralized	✗	✓	✓↯	✓↯	✓↯	✓↯	✓	✓	C,V
Kathará [30]	2018	Kernel	✓	Centralized	✗	✓	✓↯	✓↯	✓↯	✗	✓	✗	C
Kollaps	**2020**	**Kernel**	✓	**Decentralized**	✓	✓	✓↯	✓↯	✓↯	✓↯	✓	✓	**C,V**

RELATED WORK

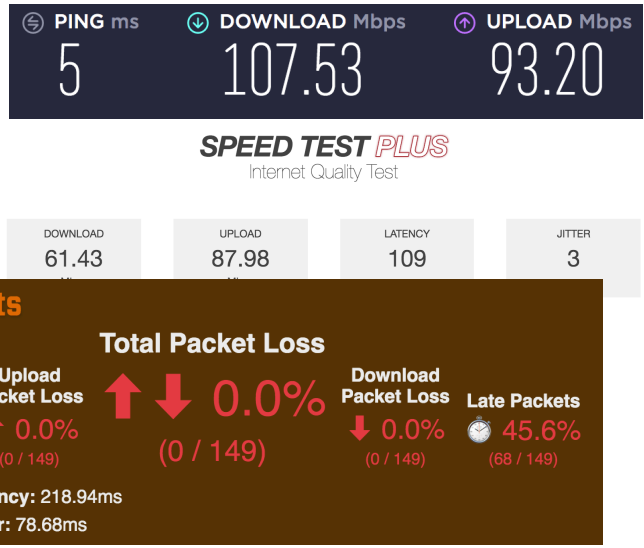
Name	Year	Mode	HW ind.	Orchestration	Concurrent deployments	Path congestion	Link-Level emulation capabilities: Bandwidth	Delay	Packet loss	Jitter	Any Language	Topology dynamics	Unit
DelayLine [47]	1994	User	✓	Centralized	✗	✗	✗	✓	✓	✗	✓	✗	P
ModelNet [81]	2002	Kernel	✓	Centralized	✗	✓	✓↯	✓↯	✓↯	✗	✓	✓	P
Nist NET [33]	2003	Kernel	✓	Centralized	✗	✗	✓↯	✓↯	✓↯	✓↯	✓	✗	P
NetEm [45]	2005	Kernel	✓	*(N/A: single link emulation only)*			✗	✓	✓	✓	✓	✗	P
Trickle [39]	2005	User	✓						✗	✗	✓	✗	P
EmuSocket [23]	2006	User	✓						✗	✗	✓	✗	P
ACIM/FlexLab [71]	2007	Kernel	✓						✓↯	✓↯	✓	✓	V
NCTUns [85]	2007	Kernel	✓						✓	✓	✓	✗	P
Emulab [46, 88]	2008	Kernel	✗						✓↯	✗	✓	✓	V
IMUNES [70]	2008	Kernel	✗						✓	✗	✓	✗	P
MyP2P-World [75]	2008	User	✓						✓	✗	✗	✗	P
P2PLab [61]	2008	Kernel	✓						✓	✗	✗	✗	P
Netkit [67]	2008	Kernel	✓						✓↯	✗	✓	✗	V
DFS [79]	2009	User	✓						✓	✓	✗	✓	P
Dummynet [32]	2010	Kernel	✓						✓↯	✗	✓	✗	P
Mininet [53]	2010	Kernel	✓						✓↯	✓↯	✓	✓	P
SliceTime [86]	2011	Kernel	✗						✗	✗	✓	✓	V
Mininet-HiFi [44]	2012	Kernel	✓						✓↯	✓↯	✓	✓	C
SPLAYNET [76]	2013	User	✓						✓	✗	✗	✓	P
MaxiNet [87]	2014	Kernel	✓	Centralized	✗	✓	✓↯	✓↯	✓↯	✓↯	✓	✓	P
Dockemu [80]	2015	User	✓	Centralized	✗	✗	✓	✓	✓	✓	✓	✗	C
EvalBox [77]	2015	Kernel	✓	Centralized	✗	✗	✓↯	✓↯	✓↯	✓↯	✓	✓	P
ContainerNet [65]	2016	Kernel	✓	Centralized	✗	✓	✓↯	✓↯	✓↯	✓↯	✓	✓	C,V
Kathará [30]	2018	Kernel	✓	Centralized	✗	✓	✓↯	✓↯	✓↯	✗	✓	✗	C
KOLLAPS	**2020**	**Kernel**	✓	**Decentralized**	✓	✓	✓↯	✓↯	✓↯	✓↯	✓	✓	**C,V**

Main limitations:

- scalability/centralization
- accuracy
- dynamics

KOLLAPS IN A NUTSHELL

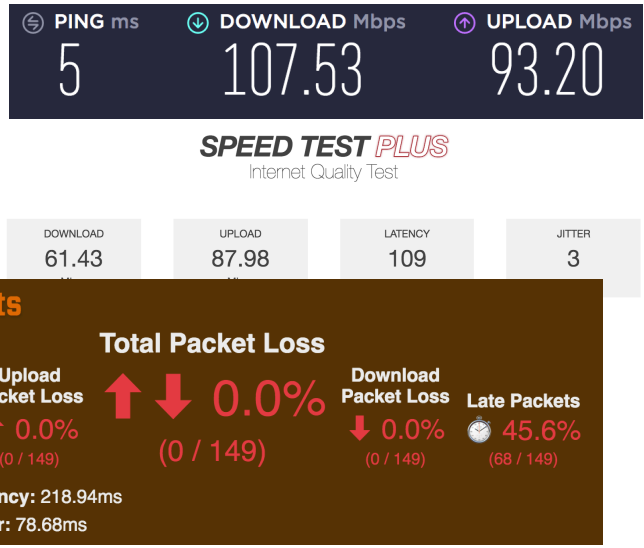
- Applications are concerned about end-to-end network properties
  - bandwidth, latency, jitter, packet loss
- Rather than the network state leading to these properties



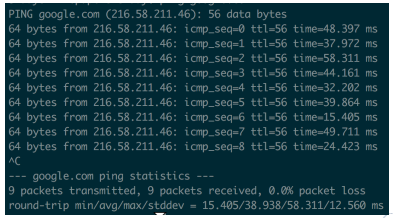

```
PING google.com (216.58.211.46): 56 data bytes
64 bytes from 216.58.211.46: icmp_seq=0 ttl=56 time=48.397 ms
64 bytes from 216.58.211.46: icmp_seq=1 ttl=56 time=37.972 ms
64 bytes from 216.58.211.46: icmp_seq=2 ttl=56 time=58.311 ms
64 bytes from 216.58.211.46: icmp_seq=3 ttl=56 time=44.161 ms
64 bytes from 216.58.211.46: icmp_seq=4 ttl=56 time=32.202 ms
64 bytes from 216.58.211.46: icmp_seq=5 ttl=56 time=39.864 ms
64 bytes from 216.58.211.46: icmp_seq=6 ttl=56 time=15.405 ms
64 bytes from 216.58.211.46: icmp_seq=7 ttl=56 time=49.711 ms
64 bytes from 216.58.211.46: icmp_seq=8 ttl=56 time=24.423 ms
^C
--- google.com ping statistics ---
9 packets transmitted, 9 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 15.405/38.938/58.311/12.560 ms
```

KOLLAPS IN A NUTSHELL

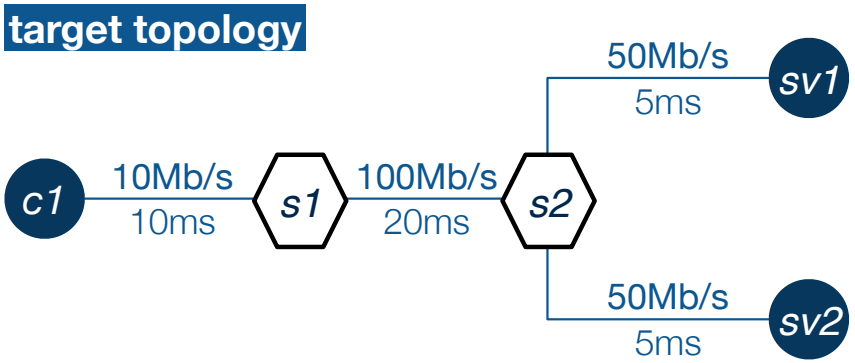
- Applications are concerned about end-to-end network properties
  - bandwidth, latency, jitter, packet loss
- Rather than the network state leading to these properties



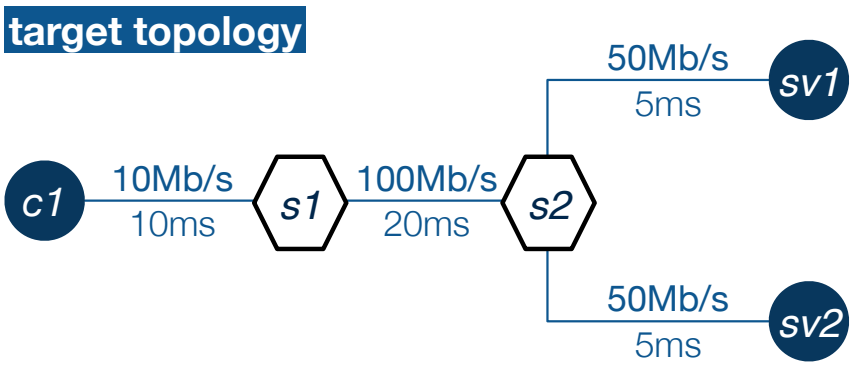
- **Emulate the emerging properties rather than the internal state of the network**
- Allows decentralized highly scalable emulation

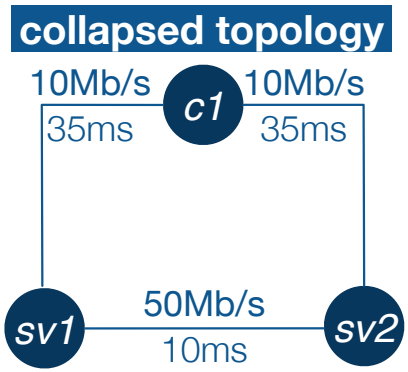


NETWORK COLLAPSING

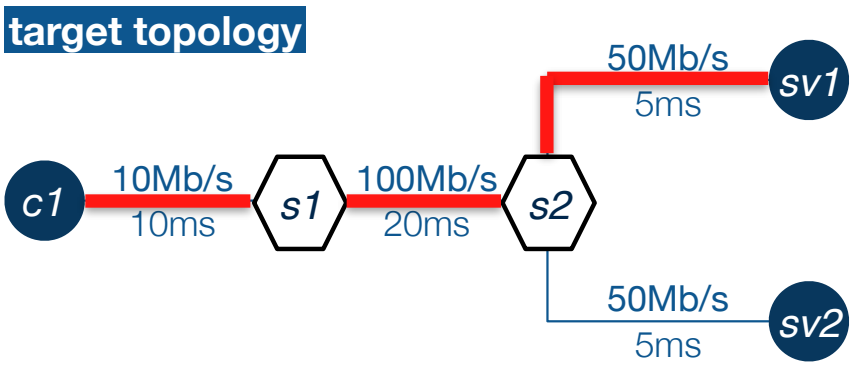


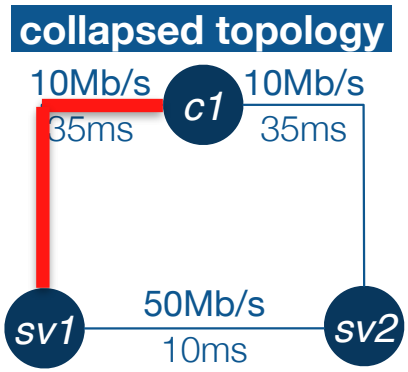
NETWORK COLLAPSING



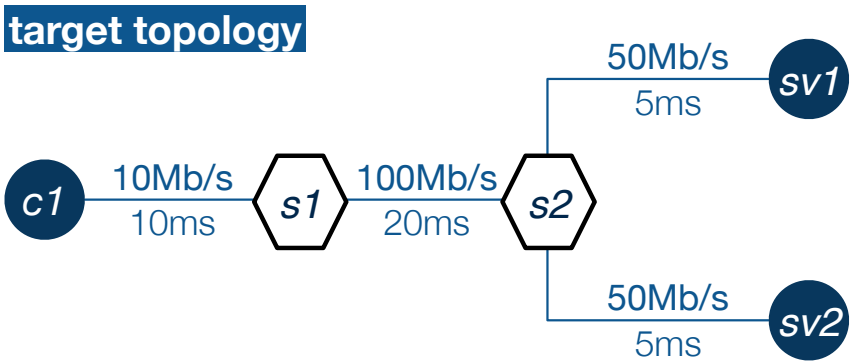


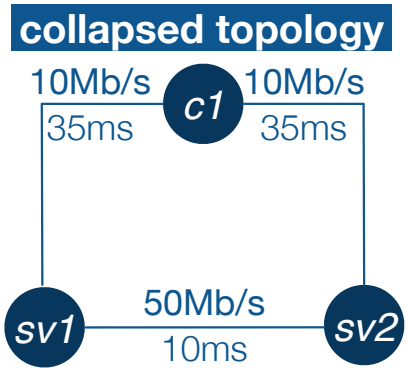
NETWORK COLLAPSING



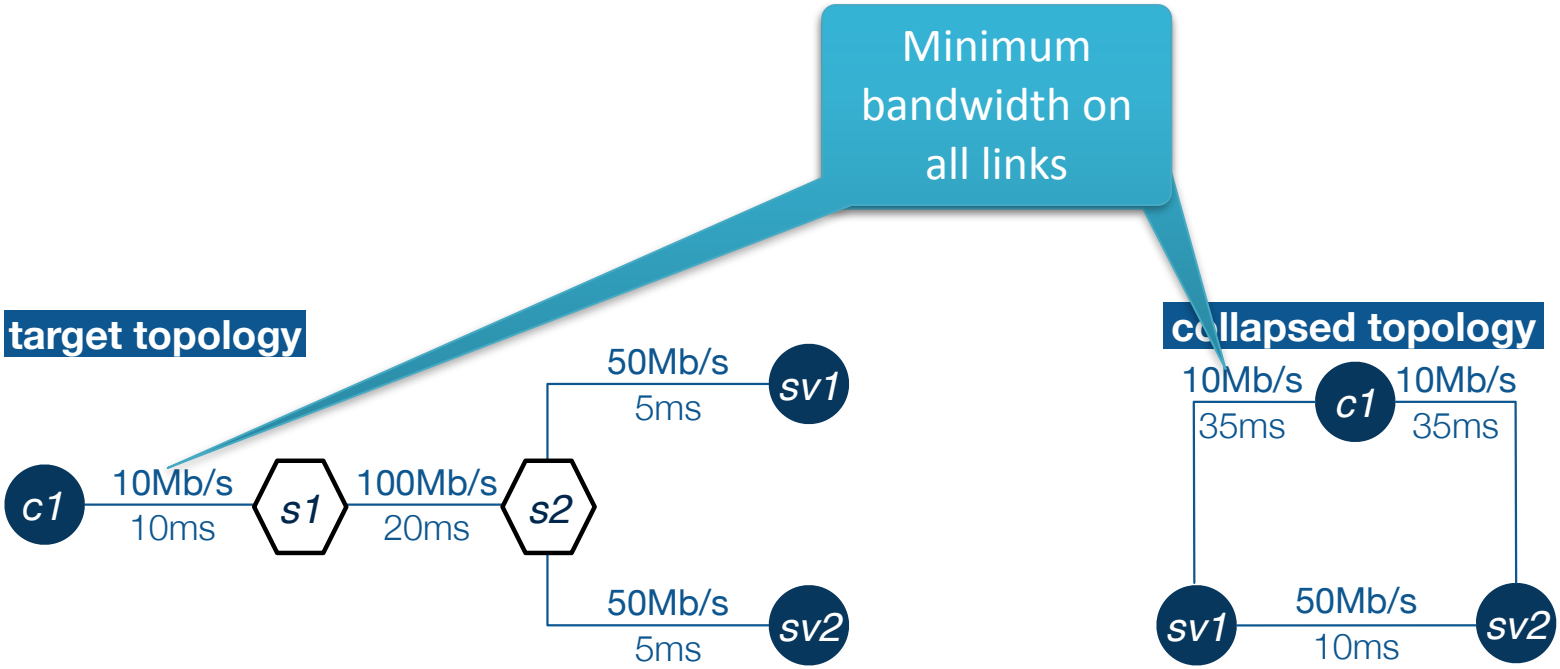


NETWORK COLLAPSING

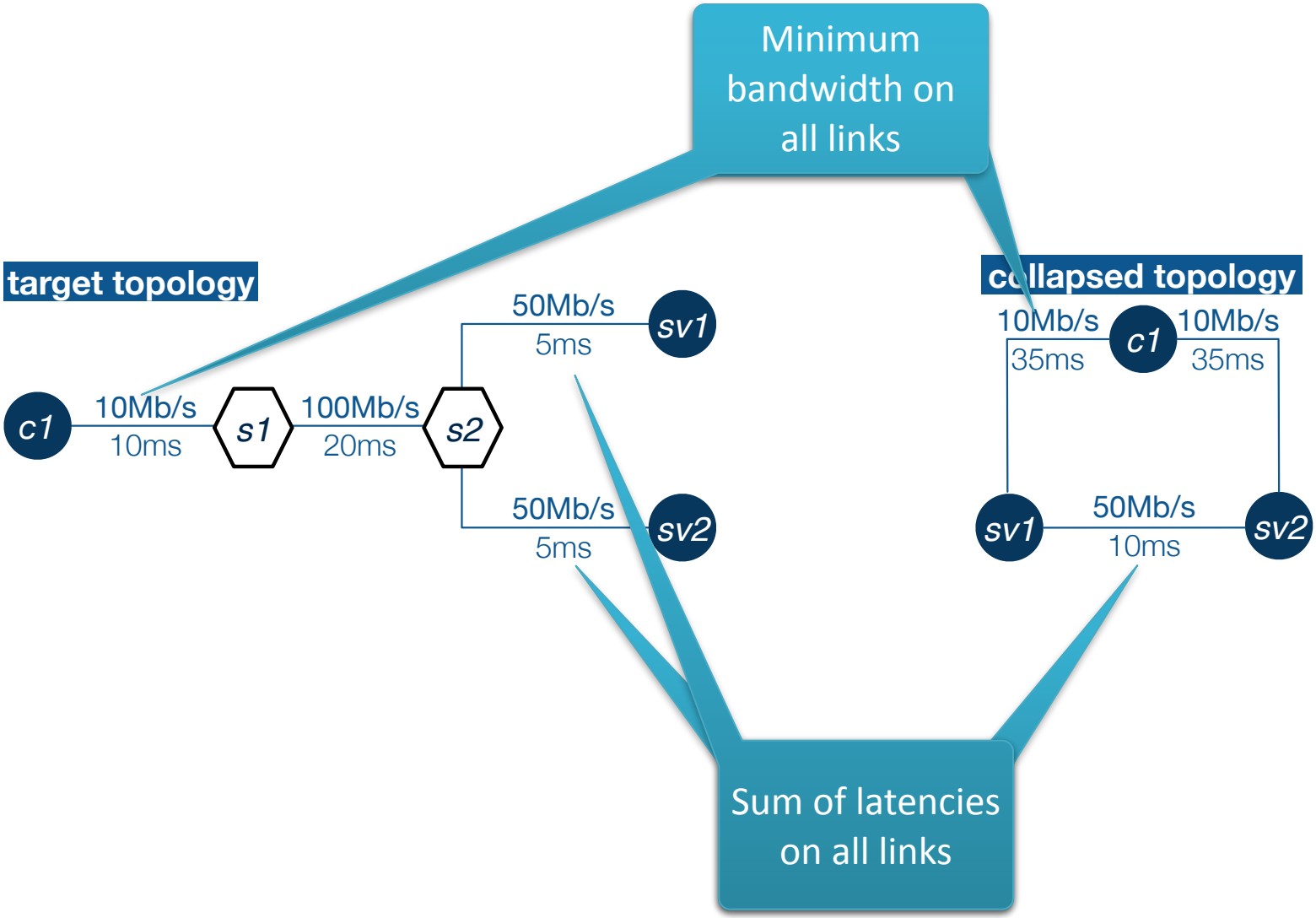




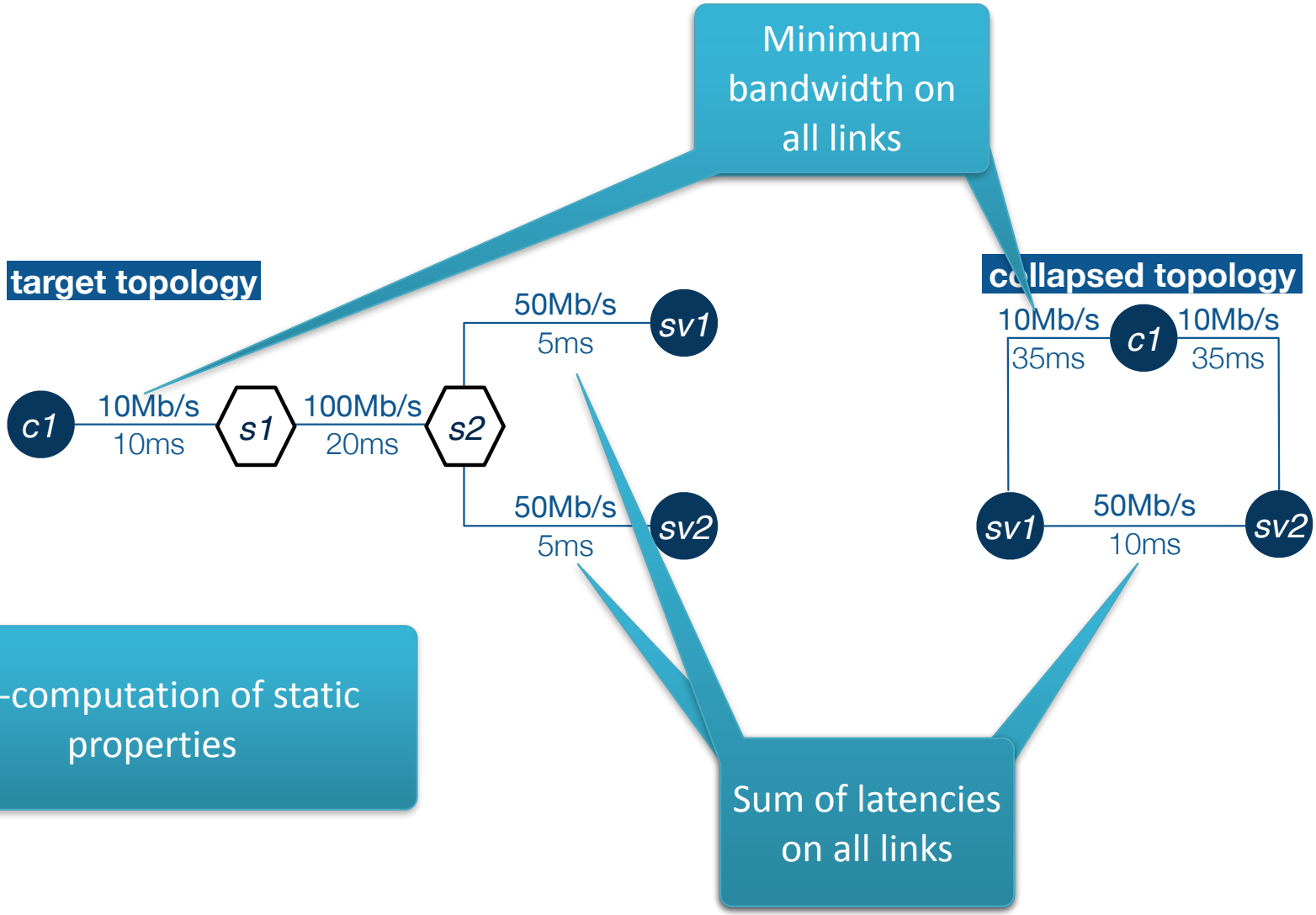
NETWORK COLLAPSING



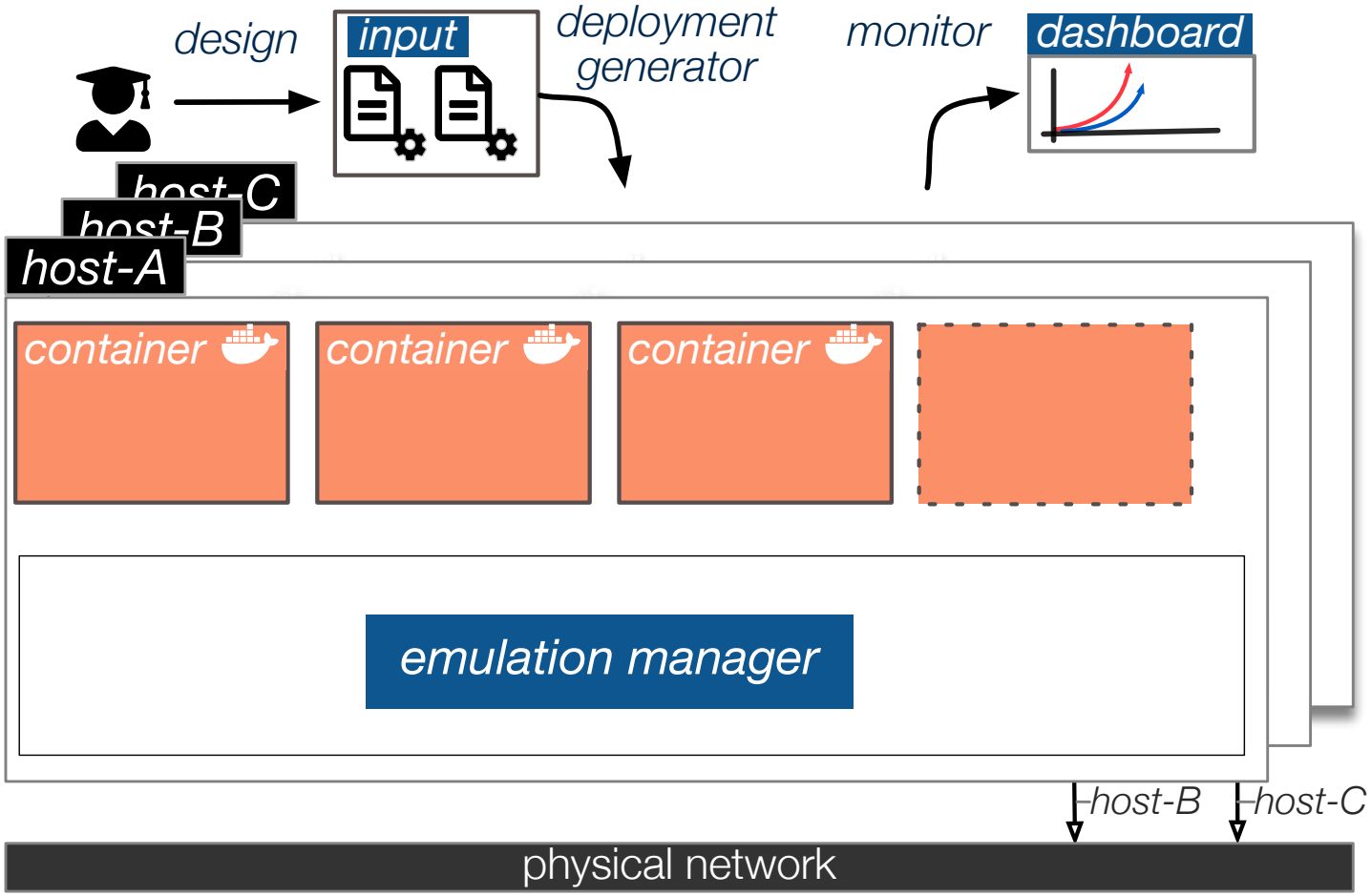
NETWORK COLLAPSING



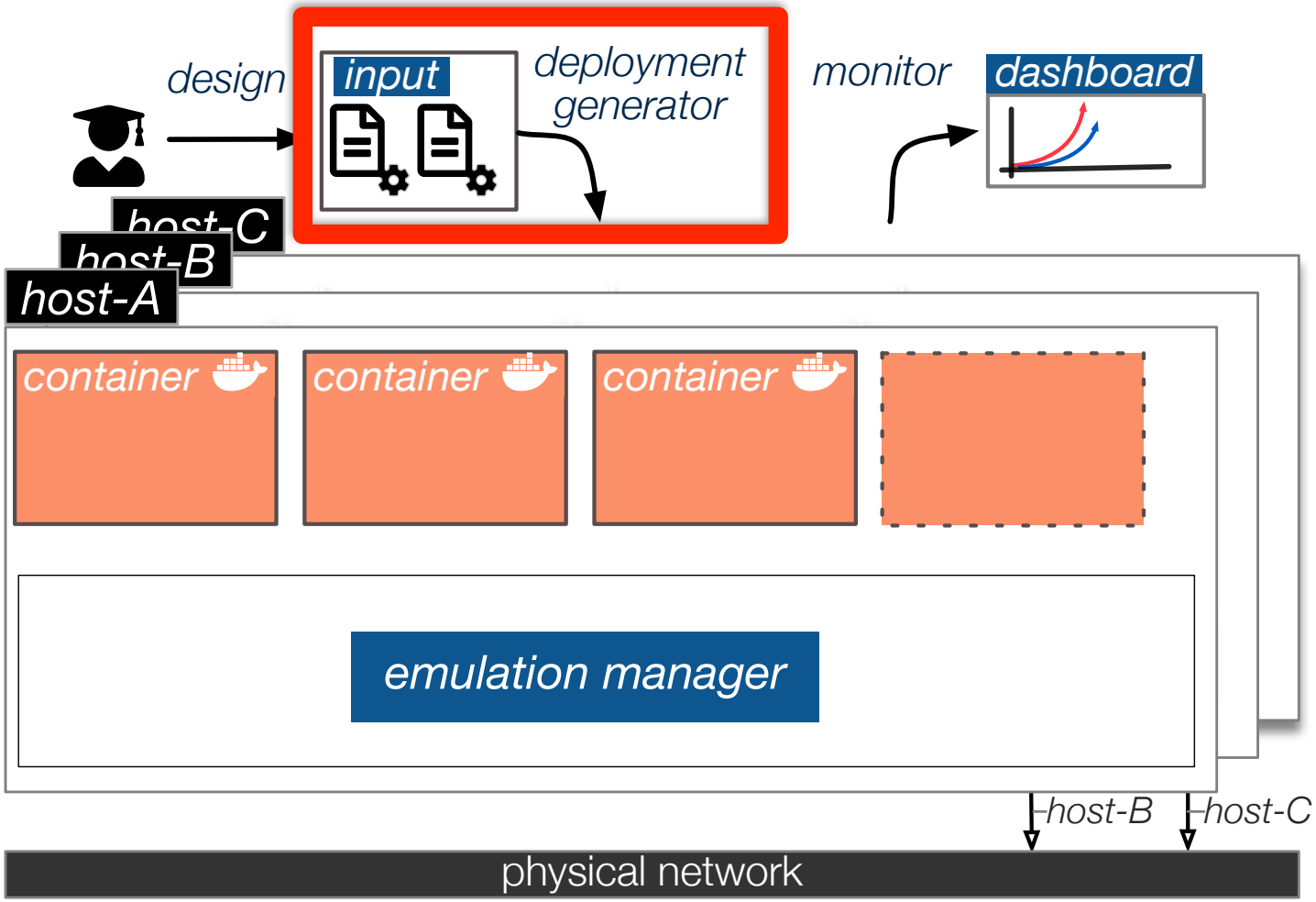
NETWORK COLLAPSING



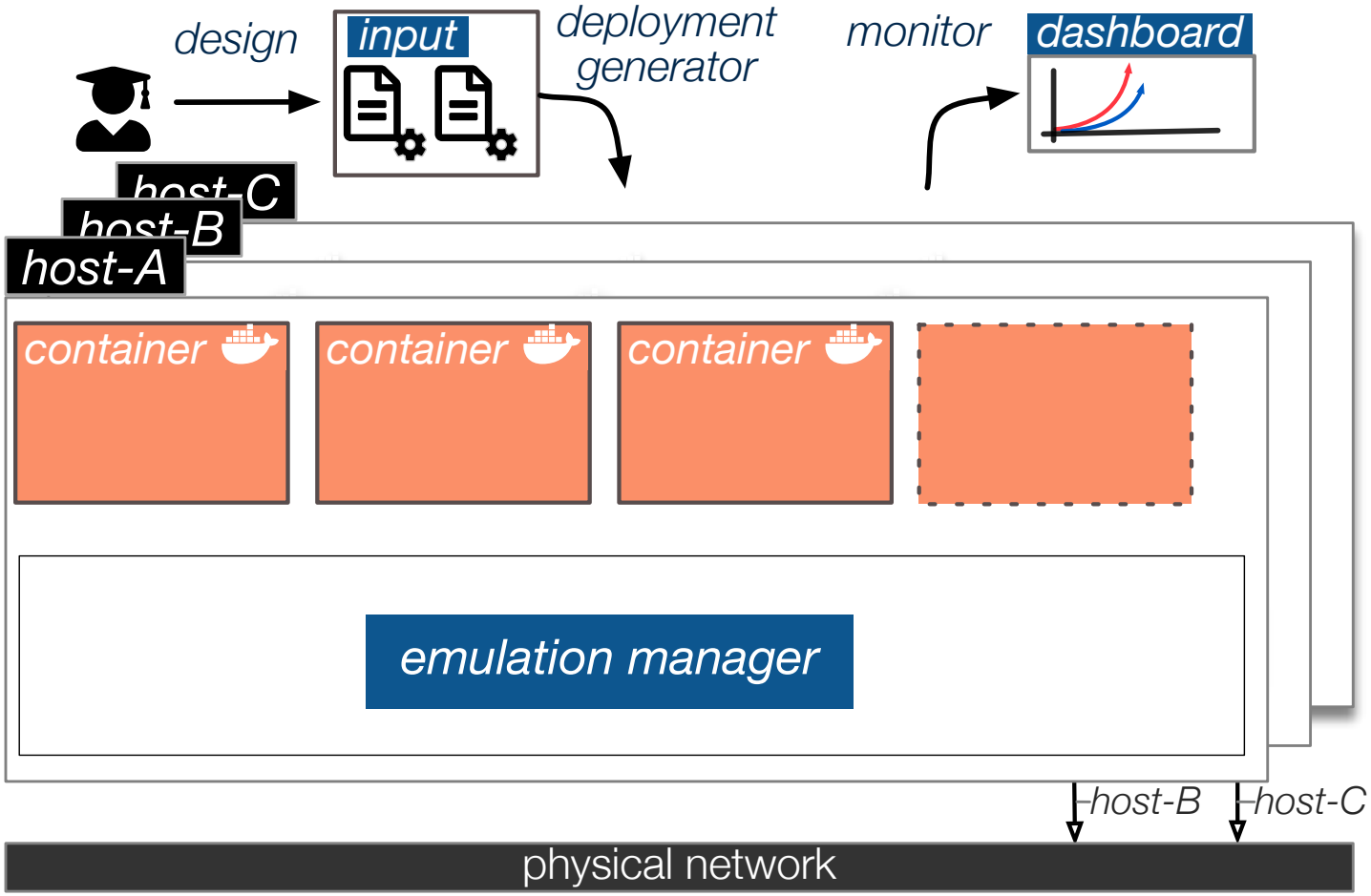
ARCHITECTURE



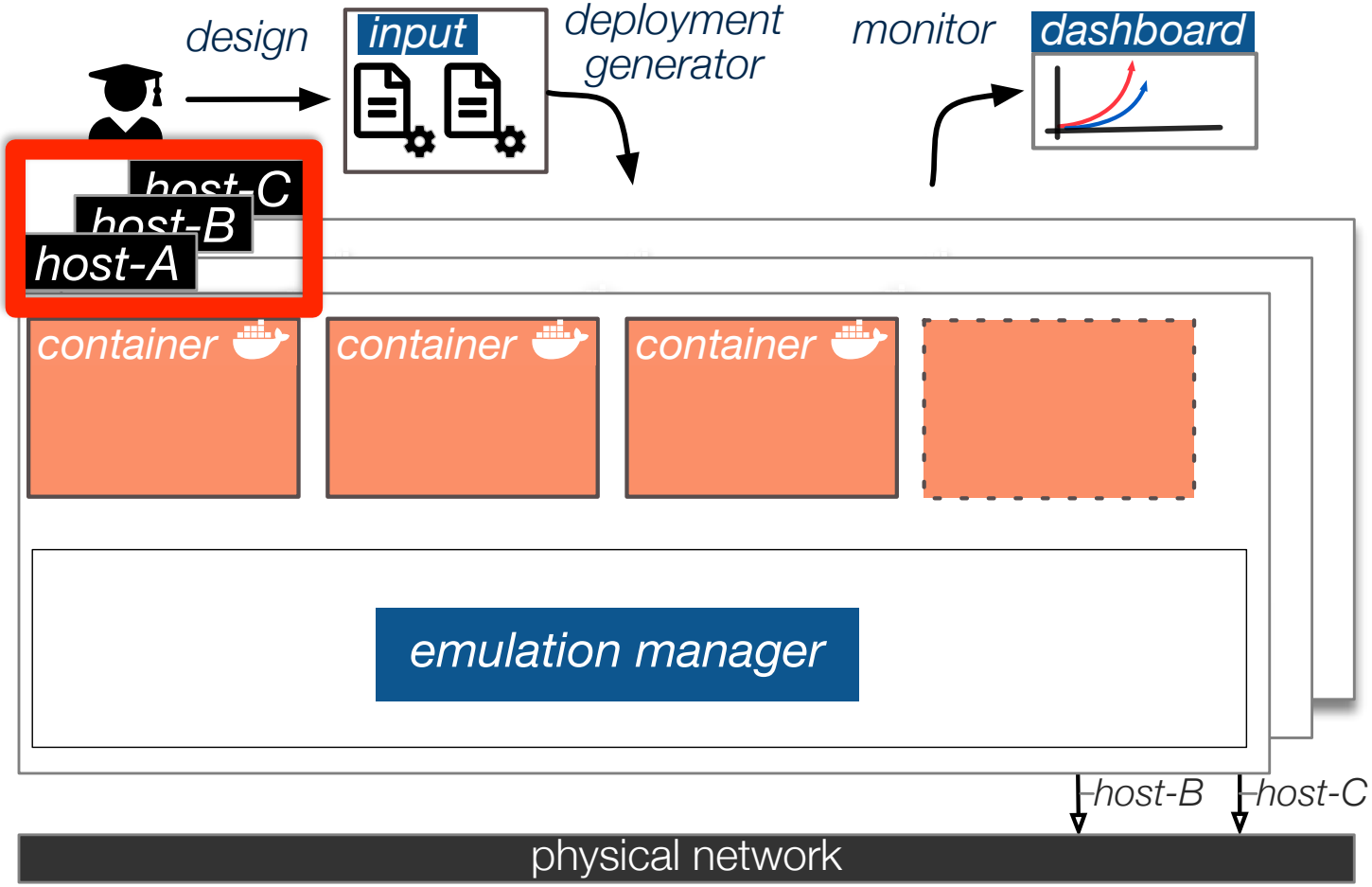
ARCHITECTURE



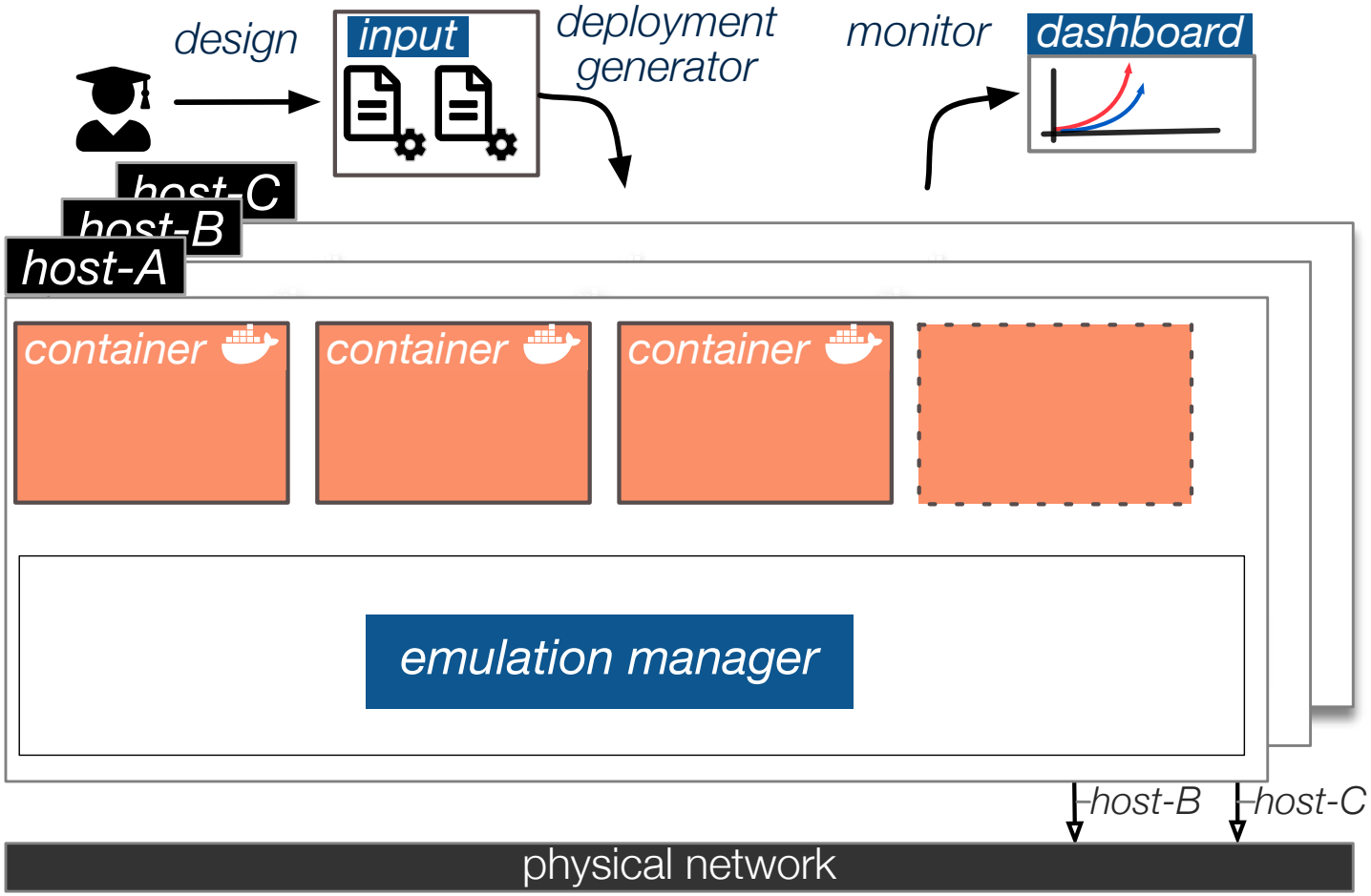
ARCHITECTURE



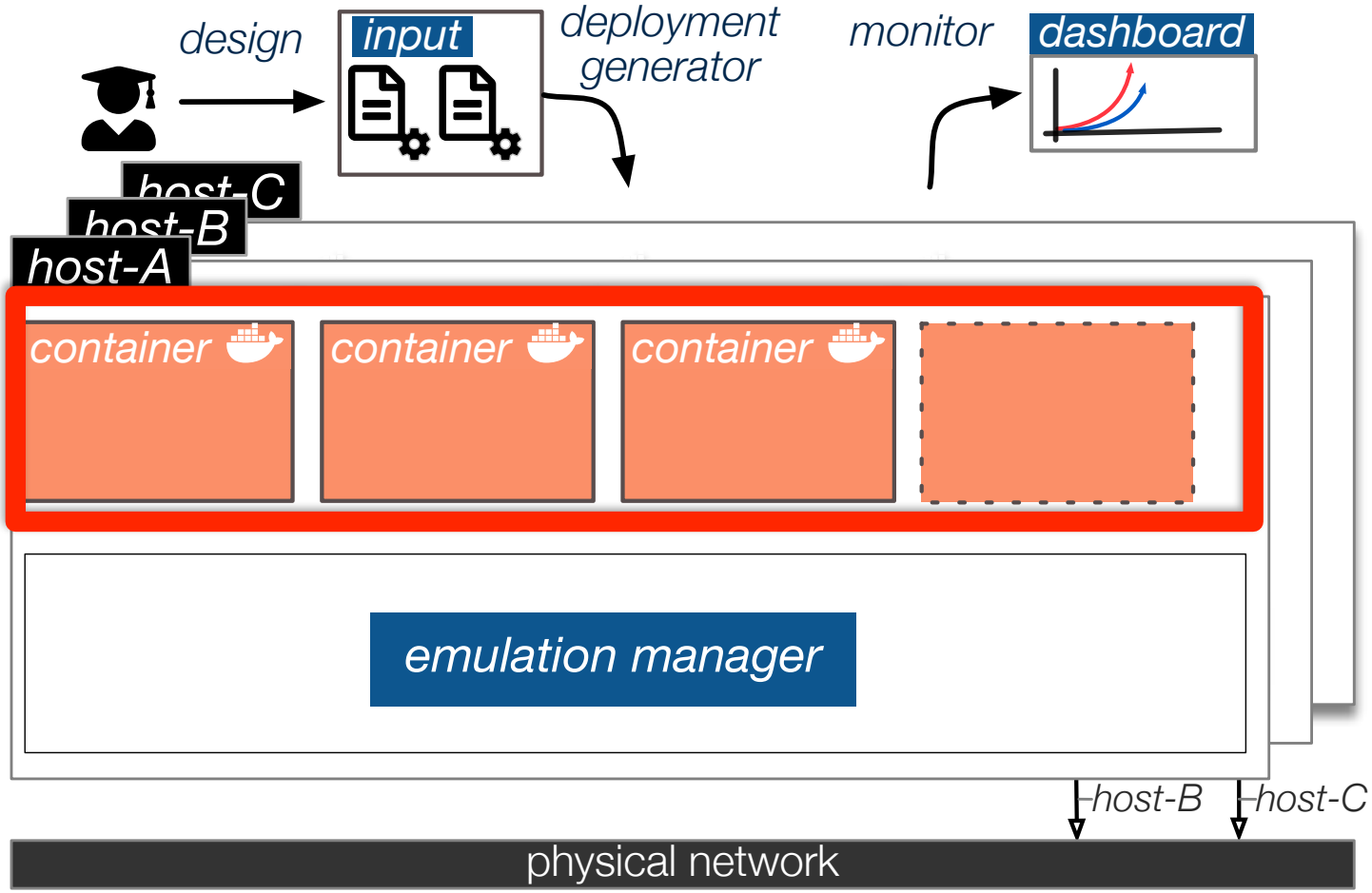
ARCHITECTURE



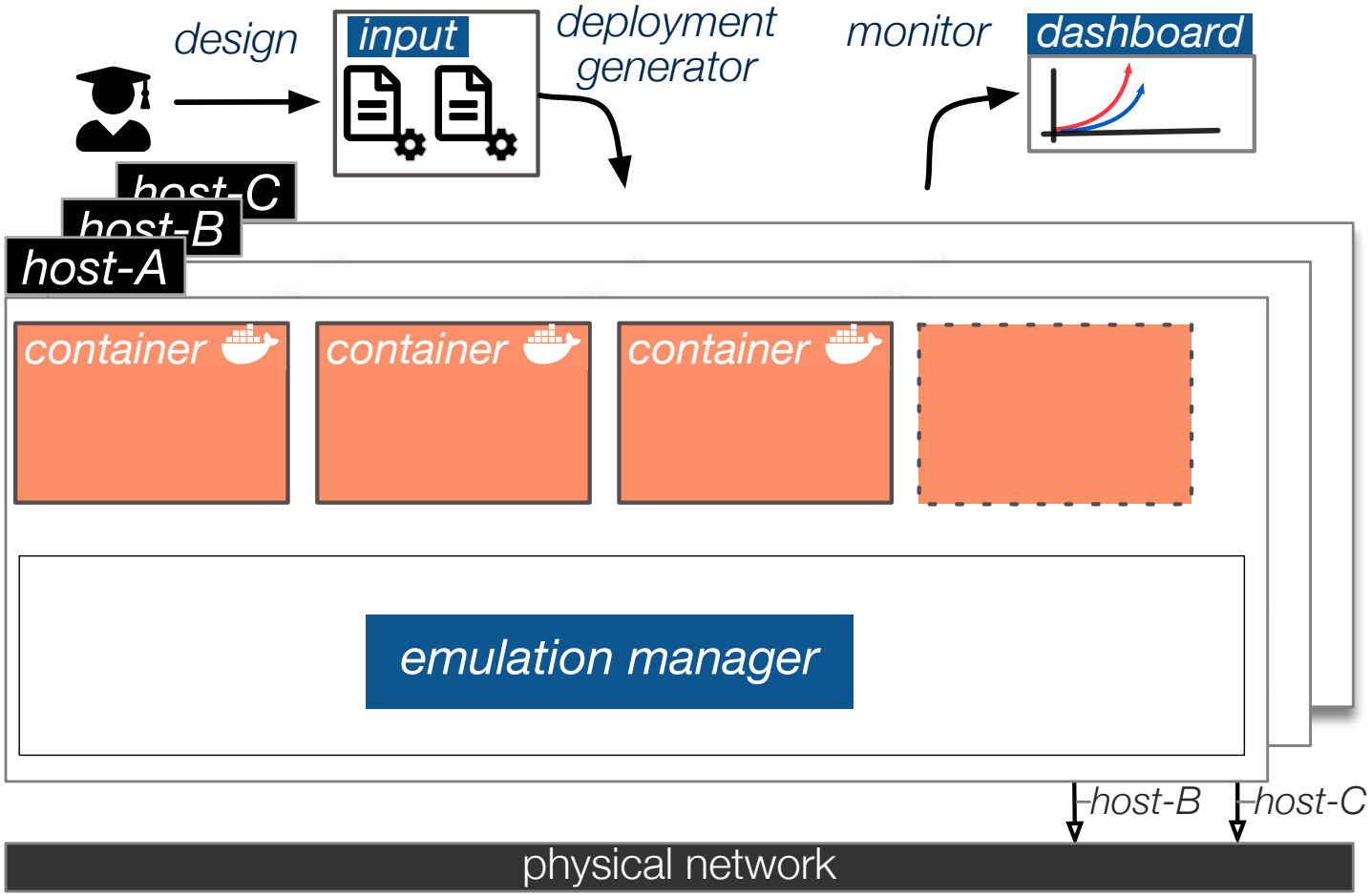
ARCHITECTURE



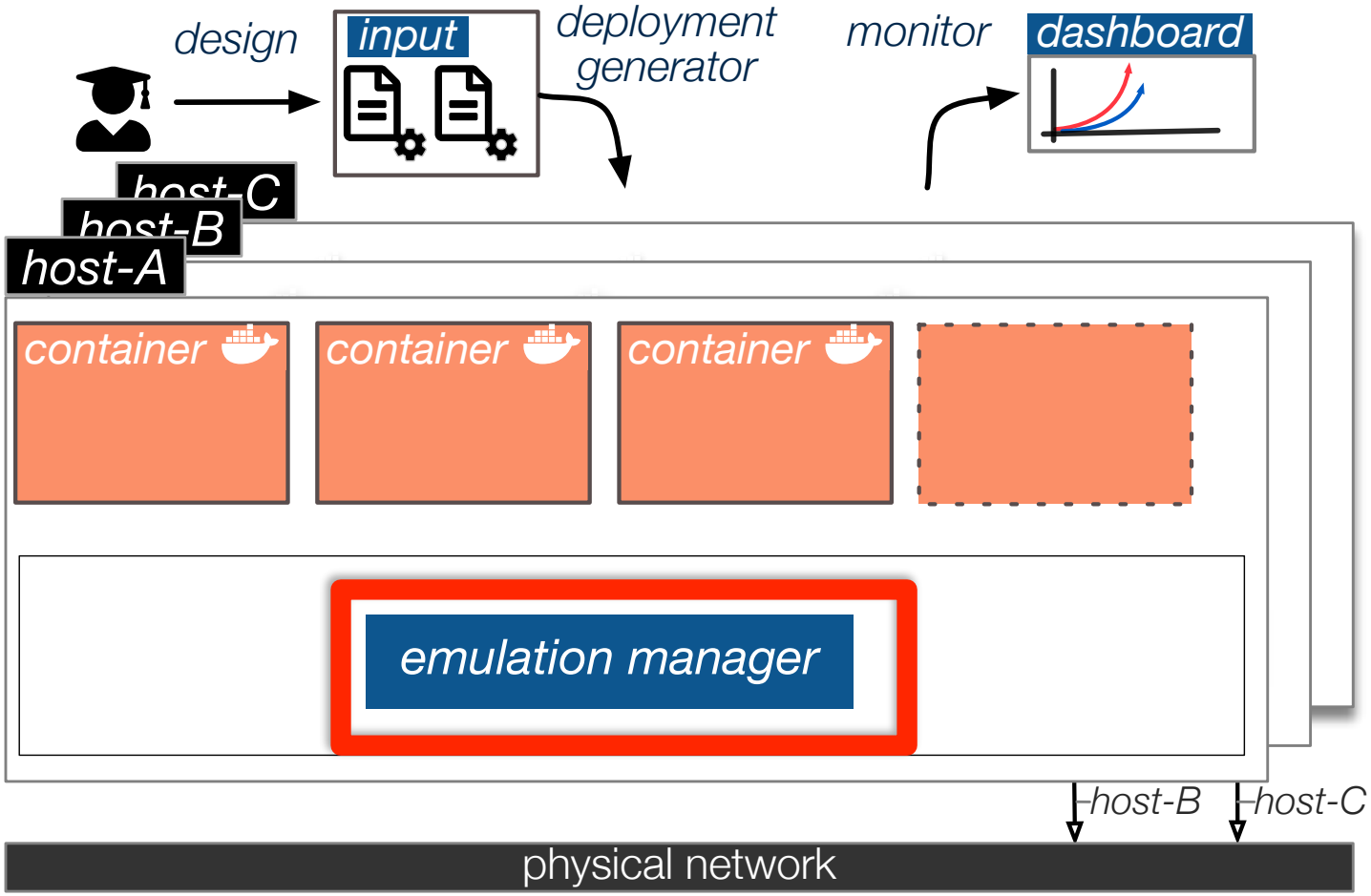
ARCHITECTURE



ARCHITECTURE

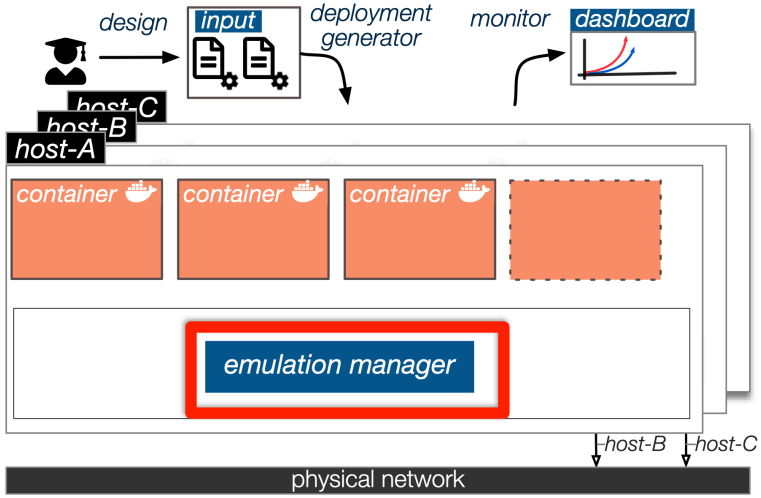


ARCHITECTURE



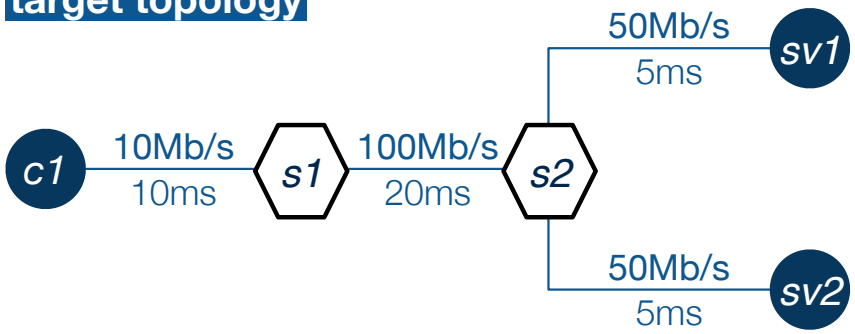
EMULATION MANAGER (EM)

- One instance per physical machine
- Enforces topology properties
  - static properties
  - dynamic properties

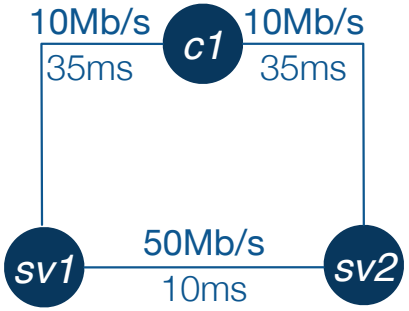


EM: DYNAMIC PROPERTIES

target topology

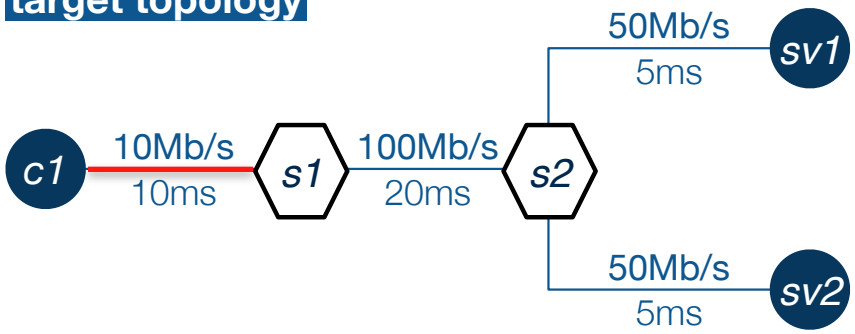


collapsed topology

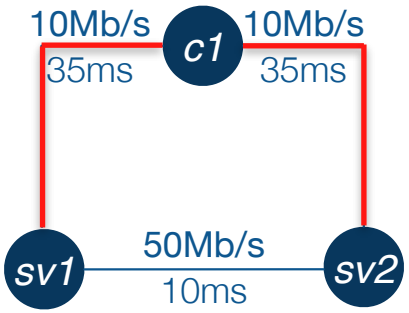


EM: DYNAMIC PROPERTIES

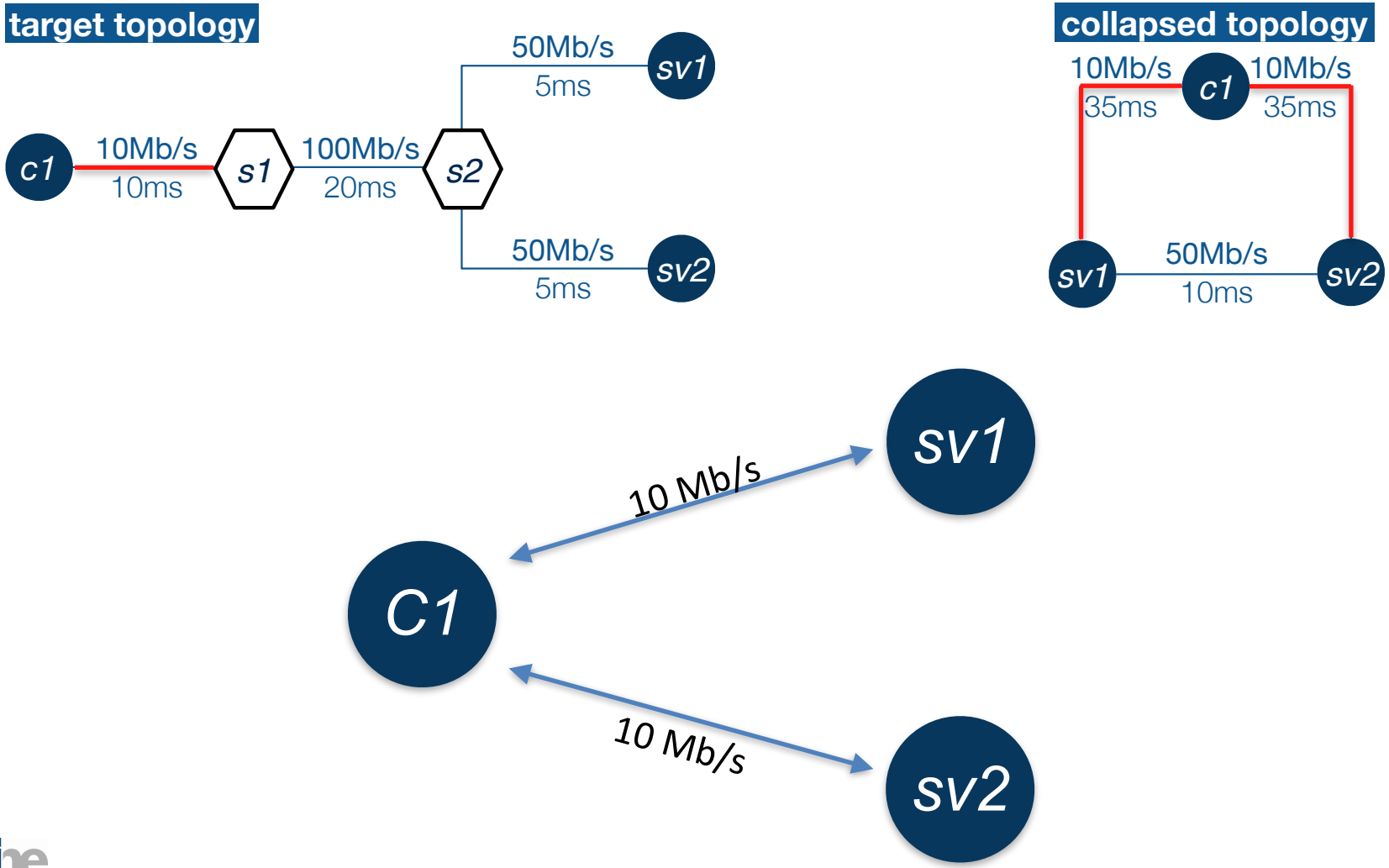
target topology



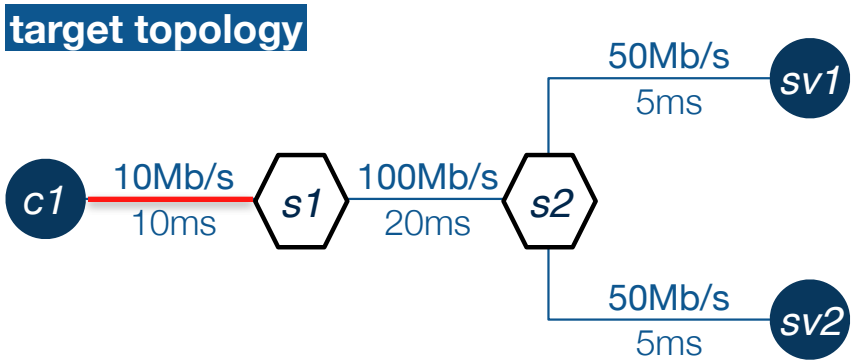
collapsed topology

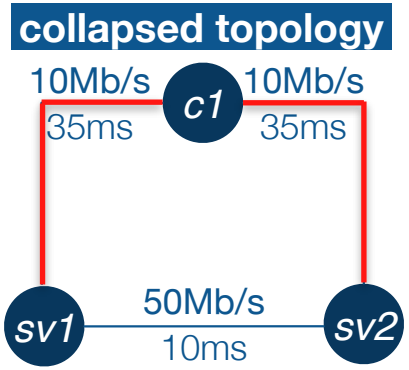


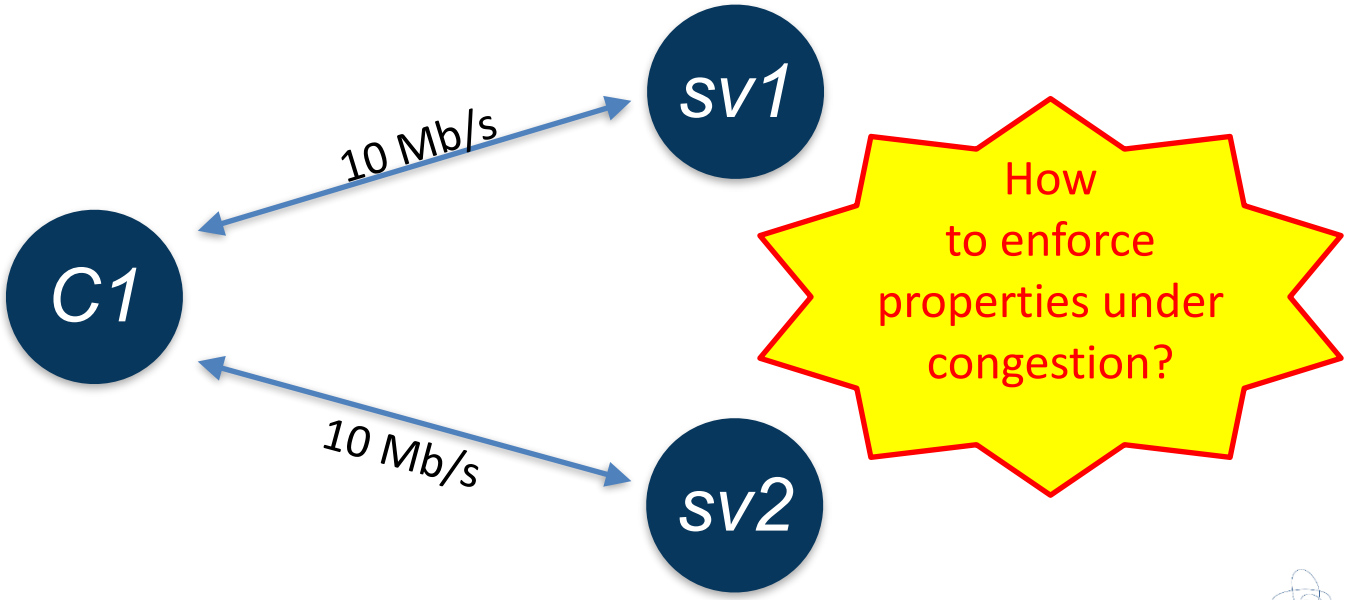
EM: DYNAMIC PROPERTIES



EM: DYNAMIC PROPERTIES







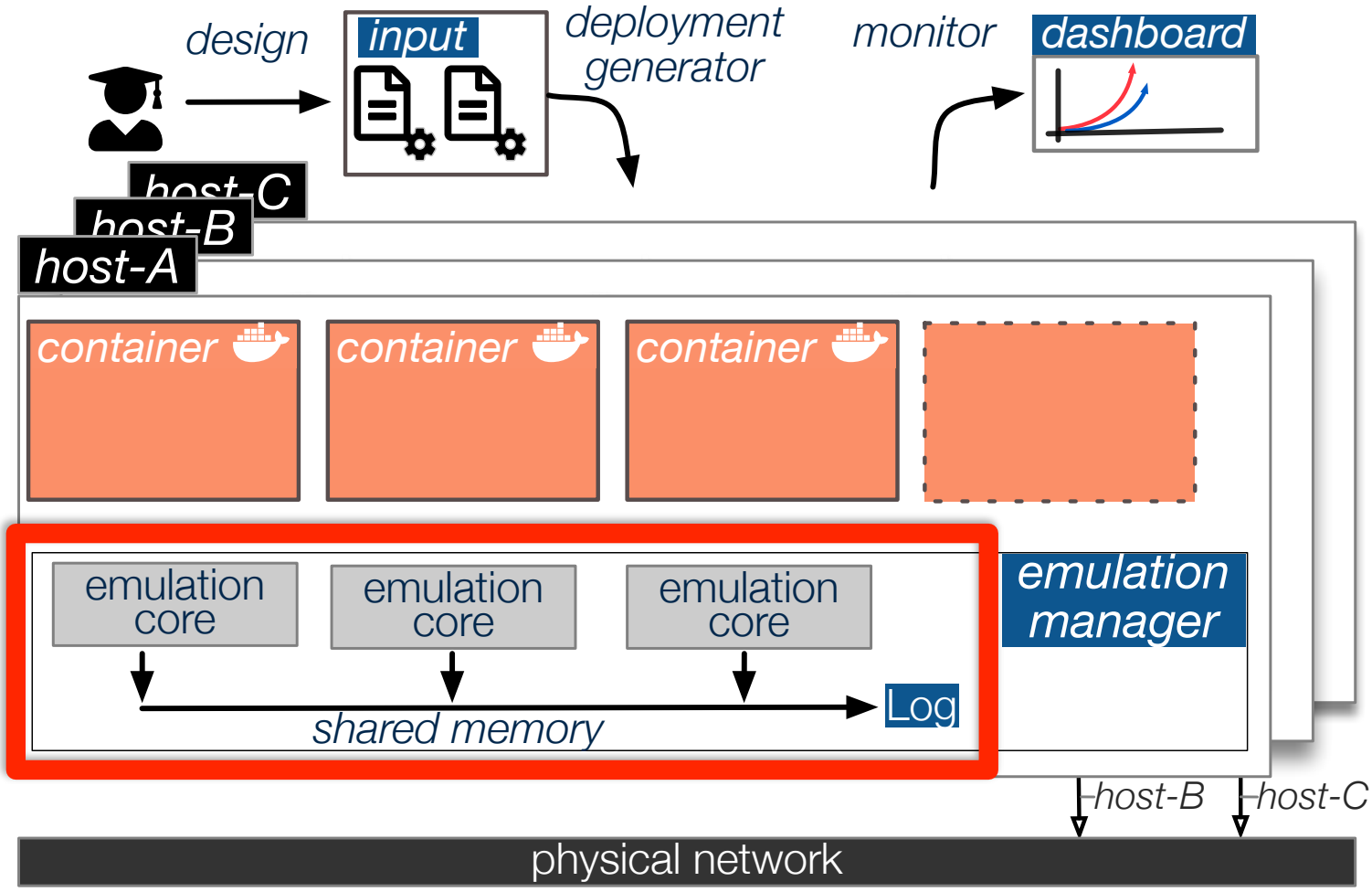
EM: DYNAMIC PROPERTIES

- RTT-Aware Min-Max model:

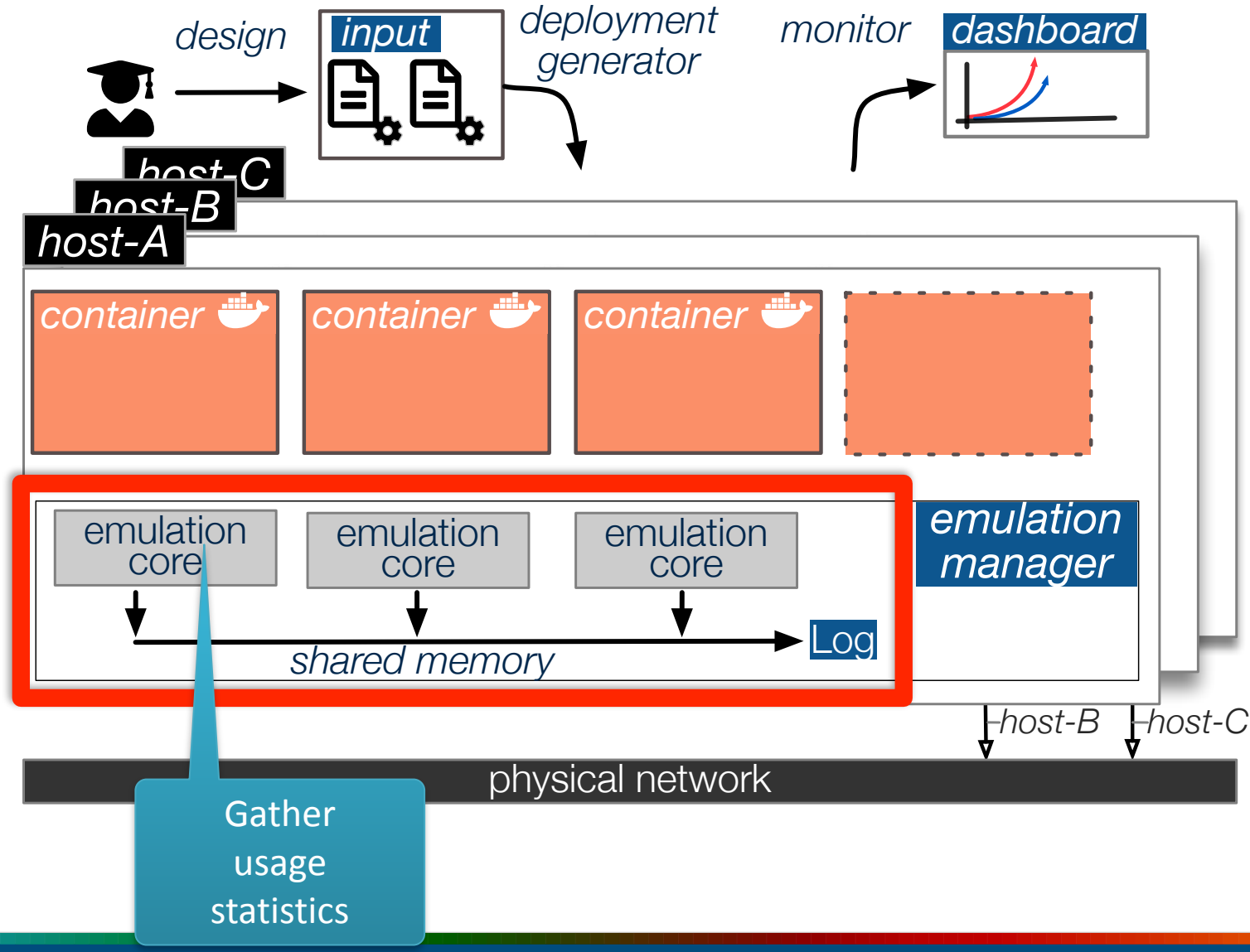
$$Share(f) = \left(RTT(f) \sum_{i=1}^{n} \frac{1}{RTT(f_i)} \right)^{-1}$$

- Intuition
  - Available bandwidth is inversely proportionally to the RTT

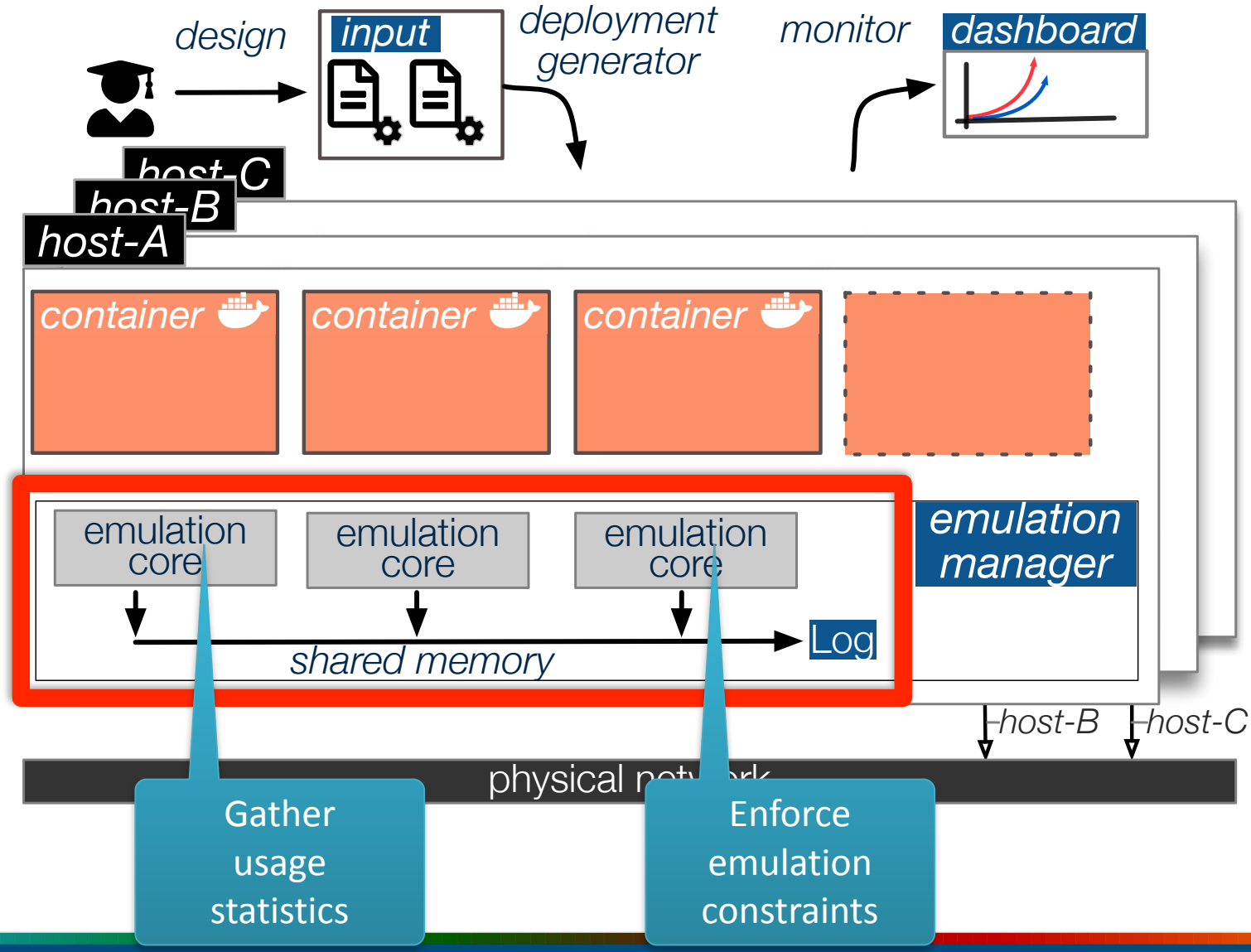
ARCHITECTURE



ARCHITECTURE



ARCHITECTURE



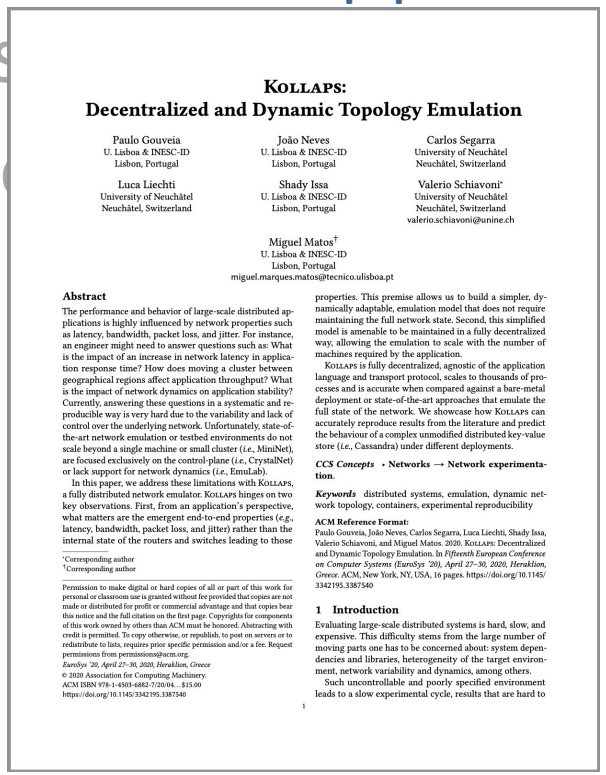
EVALUATION

- Link-level emulation
- Scalability and metadata overhead
- Short- and long-lived connections
- Cubic and Reno congestion control algorithms
- Dynamic behavior
- Large-scale topologies
- Reproducing published results
- Geo-replicated Systems
- What-if use cases

EVALUATION

- Link-level emulation
- Scalability and metadata overhead
- Short- and long-lived connections
- Cubic and Reno congestion contr
- Dynamic behavior
- Large-scale topologies
- Reproducing published results
- Geo-replicated Systems
- What-if use cases

check full paper



KOLLAPS:
Decentralized and Dynamic Topology Emulation

Paulo Gouveia
U. Lisboa & INESC-ID
Lisbon, Portugal

João Neves
U. Lisboa & INESC-ID
Lisbon, Portugal

Carlos Segarra
University of Neuchâtel
Neuchâtel, Switzerland

Luca Liechti
University of Neuchâtel
Neuchâtel, Switzerland

Shady Issa
U. Lisboa & INESC-ID
Lisbon, Portugal

Valerio Schiavoni*
University of Neuchâtel
Neuchâtel, Switzerland
valerio.schiavoni@unine.ch

Miguel Matos†
U. Lisboa & INESC-ID
Lisbon, Portugal
miguel.marques.matos@tecnico.ulisboa.pt

Abstract

The performance and behavior of large-scale distributed applications is highly influenced by network properties such as latency, bandwidth, packet loss, and jitter. For instance, an engineer might need to answer questions such as: What is the impact of an increase in network latency in application response time? How does moving a cluster between geographical regions affect application throughput? What is the impact of network dynamics on application stability? Currently, answering these questions in a systematic and reproducible way is very hard due to the variability and lack of control over the underlying network. Unfortunately, state-of-the-art network emulation or testbed environments do not scale beyond a single machine or small cluster (*i.e.*, MiniNet), are focused exclusively on the control-plane (*i.e.*, CrystalNet) or lack support for network dynamics (*i.e.*, EmuLab).

In this paper, we address these limitations with KOLLAPS, a fully distributed network emulator. KOLLAPS hinges on two key observations. First, from an application's perspective, what matters are the emergent end-to-end properties (*e.g.*, latency, bandwidth, packet loss, and jitter) rather than the internal state of the routers and switches leading to those properties. This premise allows us to build a simpler, dynamically adaptable, emulation model that does not require maintaining the full network state. Second, this simplified model is amenable to be maintained in a fully decentralized way, allowing the emulation to scale with the number of machines required by the application.

KOLLAPS is fully decentralized, agnostic of the application language and transport protocol, scales to thousands of processes and is accurate when compared against a bare-metal deployment or state-of-the-art approaches that emulate the full state of the network. We showcase how KOLLAPS can accurately reproduce results from the literature and predict the behaviour of a complex unmodified distributed key-value store (*i.e.*, Cassandra) under different deployments.

CCS Concepts • **Networks** → **Network experimentation**.

Keywords distributed systems, emulation, dynamic network topology, containers, experimental reproducibility

ACM Reference Format:
Paulo Gouveia, João Neves, Carlos Segarra, Luca Liechti, Shady Issa, Valerio Schiavoni, and Miguel Matos. 2020. KOLLAPS: Decentralized and Dynamic Topology Emulation. In *Fifteenth European Conference on Computer Systems (EuroSys '20), April 27–30, 2020, Heraklion, Greece.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3342195.3387540

*Corresponding author
†Corresponding author

EuroSys '20, April 27–30, 2020, Heraklion, Greece

ACM ISBN 978-1-4503-6882-7/20/04…$15.00
https://doi.org/10.1145/3342195.3387540

1 Introduction

Evaluating large-scale distributed systems is hard, slow, and expensive. This difficulty stems from the large number of moving parts one has to be concerned about: system dependencies and libraries, heterogeneity of the target environment, network variability and dynamics, among others.

Such uncontrollable and poorly specified environment leads to a slow experimental cycle, results that are hard to

1

LARGE-SCALE TOPOLOGIES

- Scale-free networks with random ping requests
- Mean-square error w.r.t. theoretical RTT:

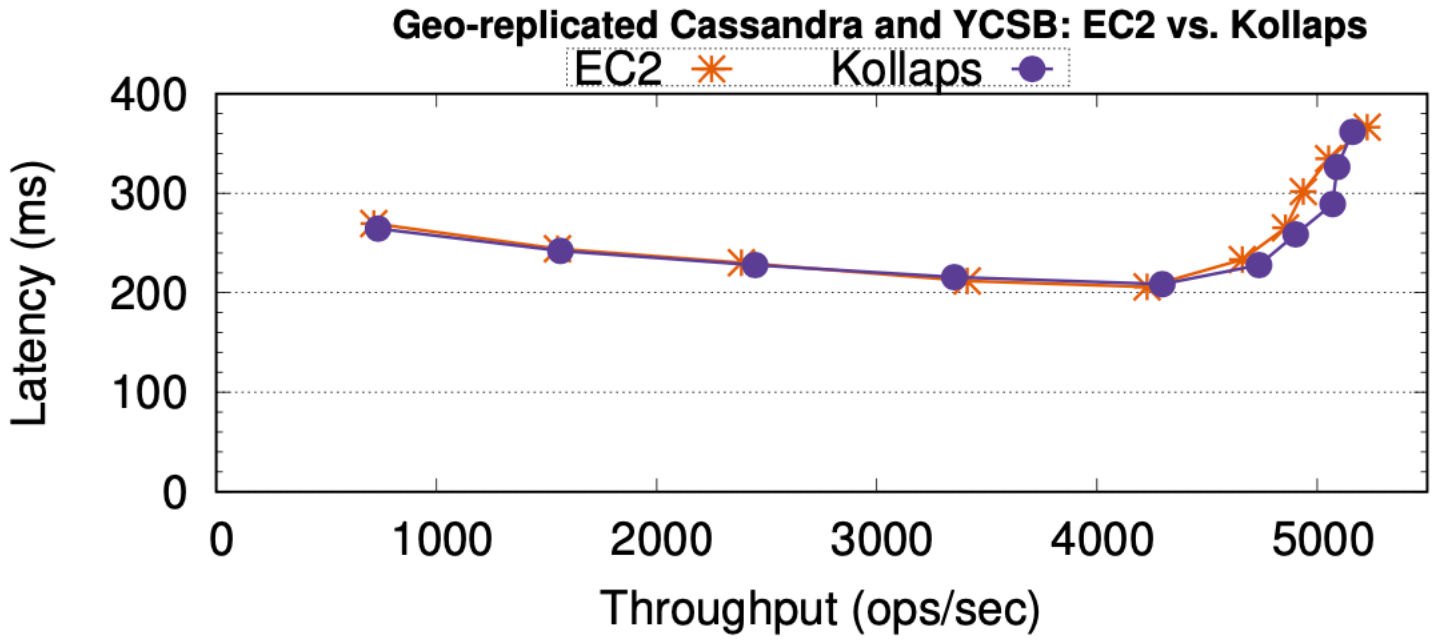
Size (# nodes + # switches)	KOLLAPS	Mininet	Maxinet
1000	0.0261	0.0079	28.0779
2000	0.0384	N/A	347.5303
4000	0.0721	N/A	N/A

GEO-REPLICATED SYSTEM

- Cassandra on EC2 (replication factor: 2)
  - 4 replicas in Frankfurt, 4 replicas in Sydney
  - 4 YCSB clients in Frankfurt
- Repeat experiment with Kollaps on a local cluster

GEO-REPLICATED SYSTEM

- Cassandra on EC2 (replication factor: 2)
  - 4 replicas in Frankfurt, 4 replicas in Sydney
  - 4 YCSB clients in Frankfurt
- Repeat experiment with Kollaps on a local cluster

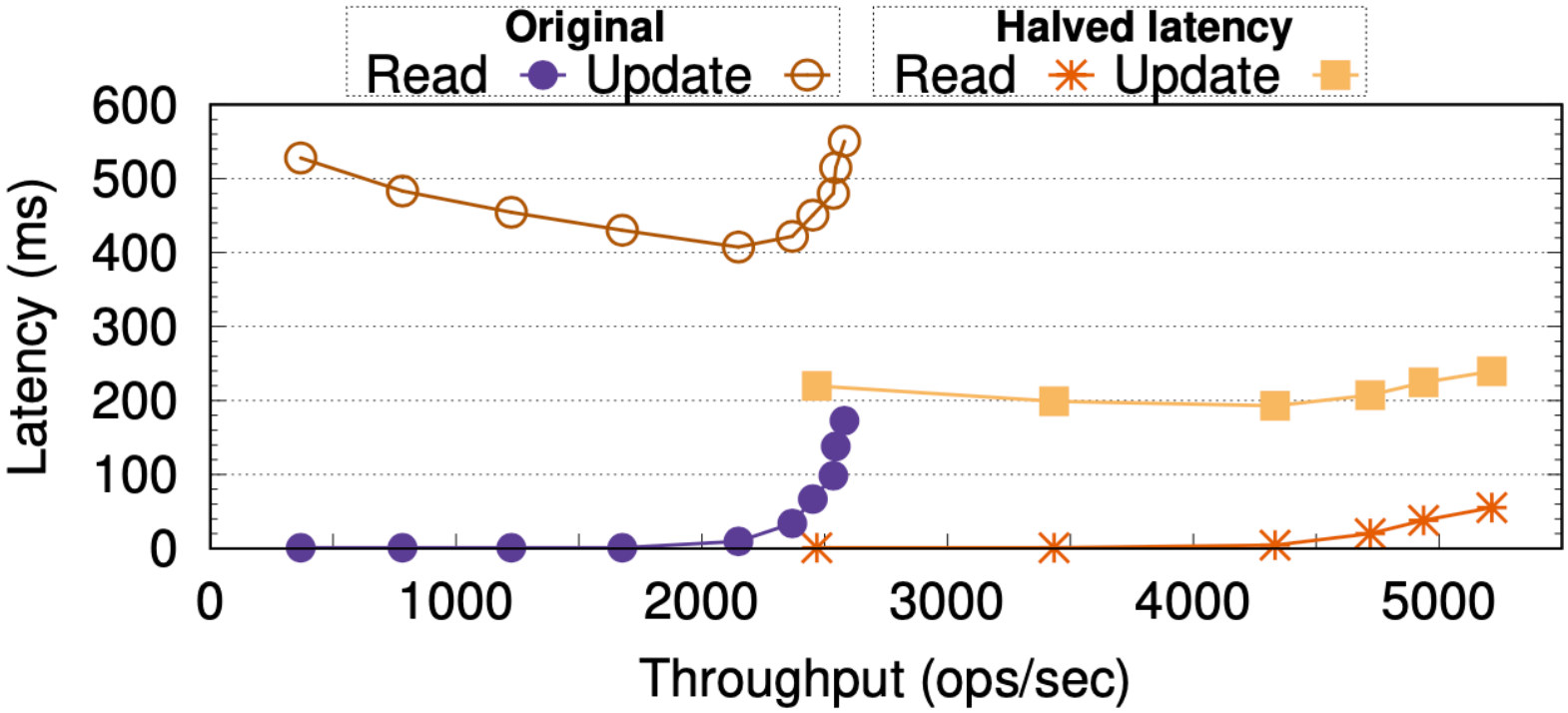


GEO-REPLICATED SYSTEM

- What is the impact of halving the network latency in application throughput?
  - E.g. move replicas from Sidney to Seoul

GEO-REPLICATED SYSTEM

- What is the impact of halving the network latency in application throughput?
  - E.g. move replicas from Sidney to Seoul



CONCLUSION AND FUTURE WORK

- KOLLAPS: a decentralized topology emulator
  - Emulation of emerging end-to-end properties
  - Allows decentralized highly scalable emulation

- Future work:
  - Adding interactive control of experiments
  - Time-dilation to mitigate physical limitations
  - Event-based meta-data dissemination

https://github.com/miguelammatos/Kollaps