

Towards realistic benchmarks for virtual infrastructure resource allocators

Qin Yin, Timothy Roscoe
Systems Group, ETH Zurich

This talk is about



How to benchmark and evaluate
various aspects of
virtual infrastructure resource allocation

Virtual infrastructure resource allocators



- Allow users to reserve or use
 - Combinations of (virtual) nodes, switches, and network links
 - On a variety of virtual infrastructures
 - Ranging from network testbeds to cloud computing facilities

Examples



- Testbeds
 - PlanetLab: best-effort VMs
 - Emulab: guaranteed-share scheduling of VN requests
 - GENI: ProtoGENI, ORCA/BEN, VINI for VN requests
- Grid/cloud computing
 - Globus Condor: batch scheduling of machine requests
 - Amazon EC2: real-time allocation of VMs
 - Private/hybrid clouds like Eucalyptus, OpenNebular allows scheduler module plugins

Why benchmarking

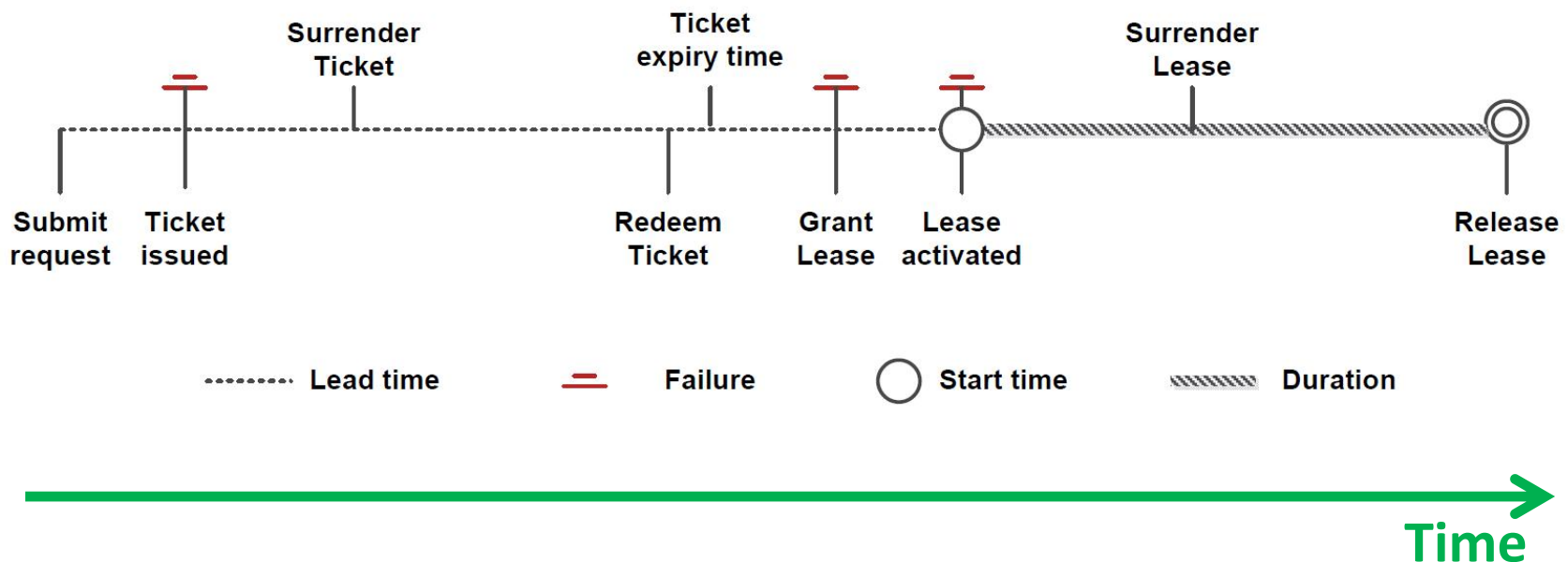


- Benchmarking is essential for
 - Evaluating system performance
 - Debugging
 - Comparing design ideas
- Important and well-established in many areas
 - DBMS, OS, web servers, supercomputers, etc.
- Virtual network resource allocation
 - Relatively new area
 - Lacks realistic benchmarks

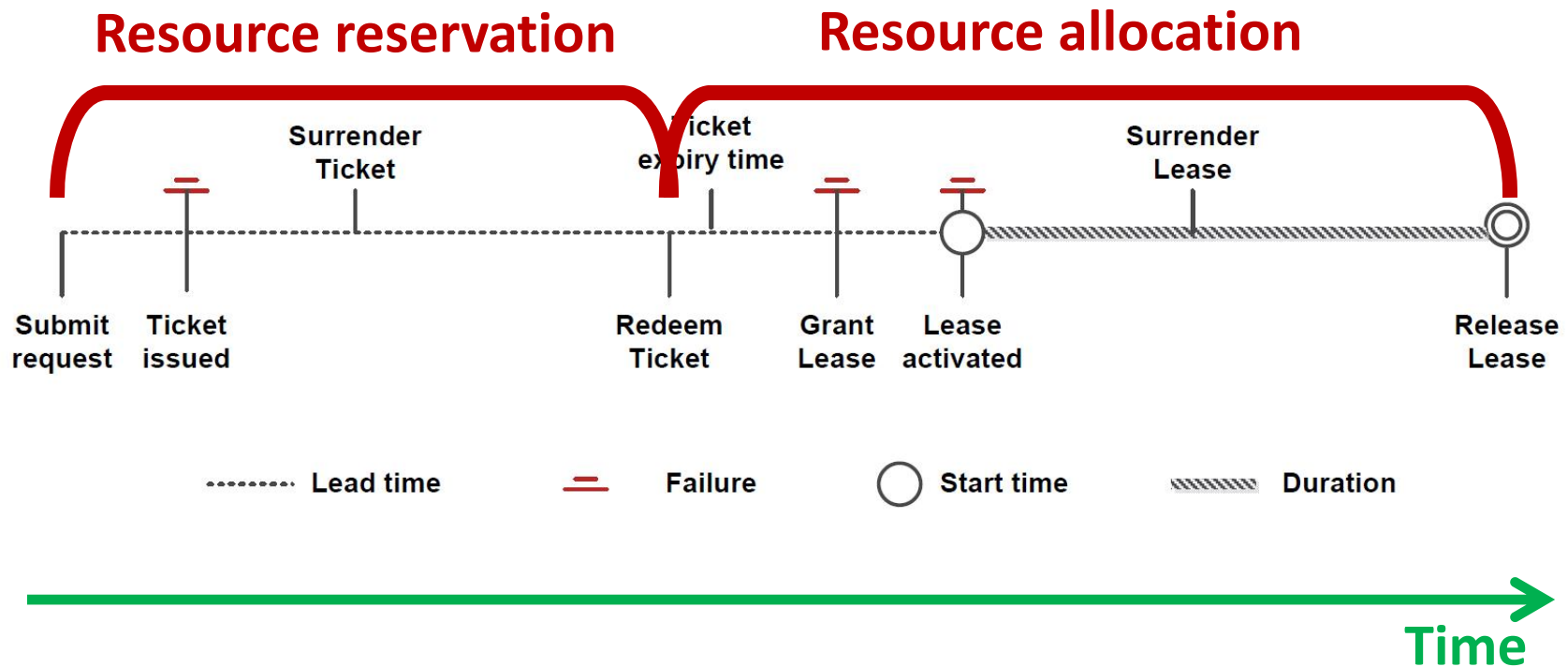
Resource allocation reference model



- Existing resource allocators follow non-strict subset of this model



Resource allocation reference model

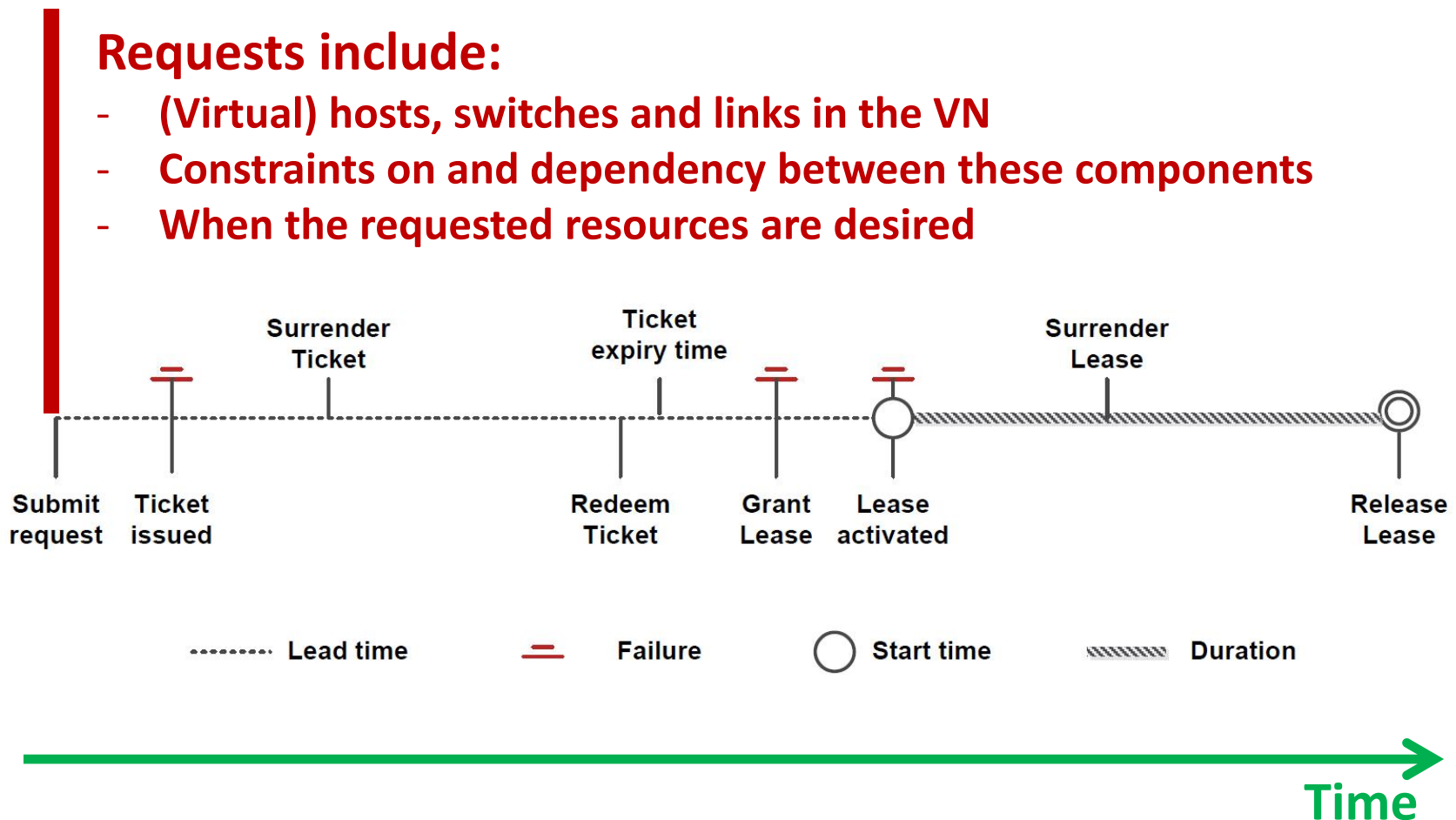


Resource allocation reference model



Requests include:

- (Virtual) hosts, switches and links in the VN
- Constraints on and dependency between these components
- When the requested resources are desired

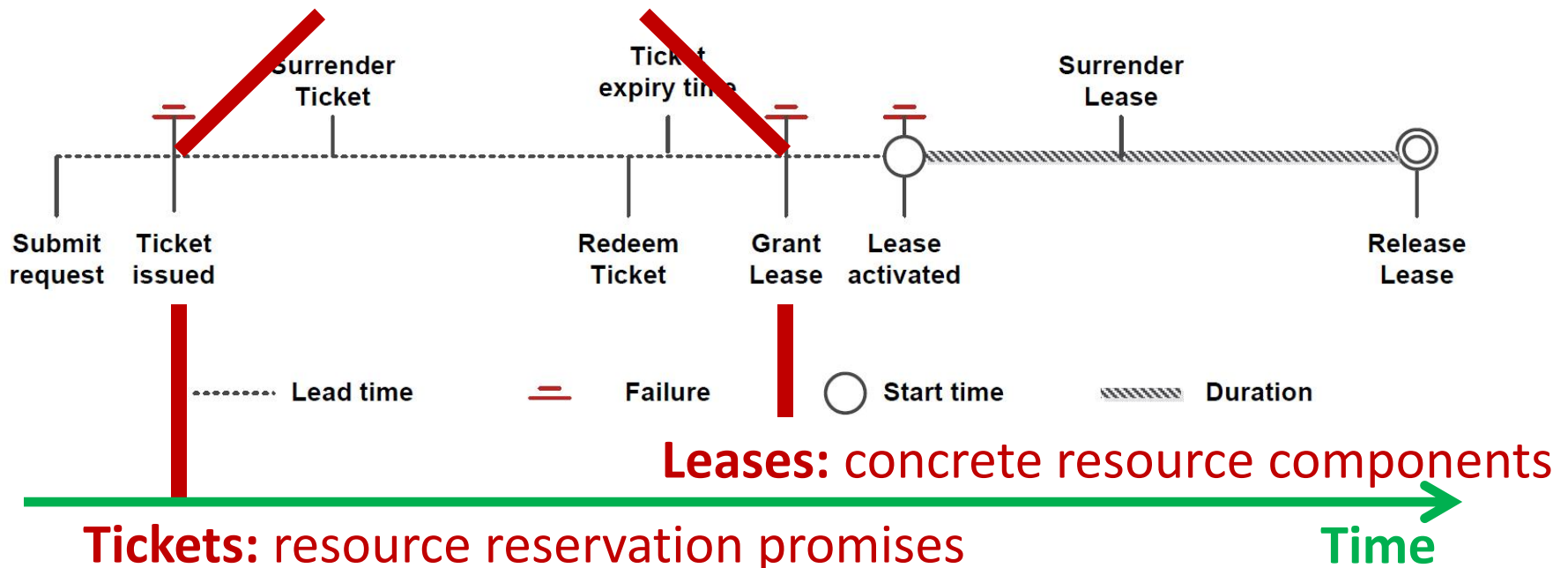


Resource allocation reference model



Online resource allocation:

- Virtual network mapping algorithm
- Satisfies multiple requests while optimizing utilization, revenue, etc.



Large design space leads to



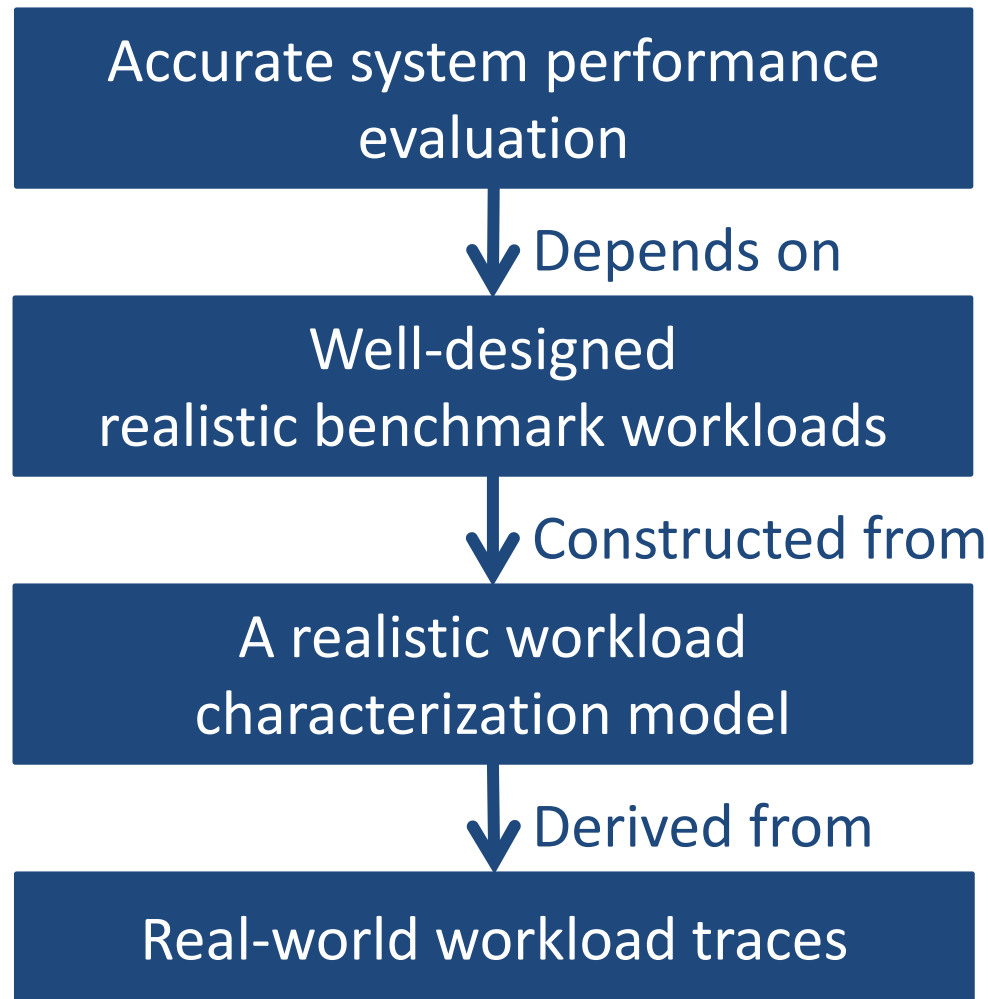
- Different evaluation scenarios
 - Compare virtual network mapping algorithms
 - Investigate dynamic resource allocation behavior
 - Evaluate resource allocation strategies
- Different evaluation criteria
 - Solving time, scalability, revenue-to-cost ratio
 - Number/percentage of successful mappings
 - Resource utilization
- Criteria depend on the evaluation scenario

Synthetic workloads



- Synthetic workloads are useful
- However
 - Randomly generated graphs
 - May not represent real requests or physical infrastructures
 - May miss key aspects in reality that significantly impact system design

Realistic workloads



Methodology for realistic workloads



- Choose parameter set
 - To describe workload behavior
- Collect system workload trace
 - Using monitoring tools
- Analyze collected trace
 - Characterize it using statistical methods
 - Construct workload distribution models
- Generate realistic workloads

Parameters

- VN topology requests
- VN request arrivals
- Duration
- Lead time
- $\Pr(\text{ticket expiry})$
- $\Pr(\text{ticket surrender})$
- $\Pr(\text{lease surrender})$
- ...

In our workload generator

- VN topology requests
- VN request arrivals
- Duration
- Lead time
- Pr(ticket expiry)
- Pr(ticket surrender)
- Pr(lease surrender)
- ...

Sampled from the complete 5-year Emulab VN request set

In our workload generator

- VN topology requests
 - **VN request arrivals**
 - Duration
 - Lead time
 - Pr(ticket expiry)
 - Pr(ticket surrender)
 - Pr(lease surrender)
 - ...
- Request timestamps missing from the Emulab trace
 - Common assumption: Poisson distribution
 - Use different arrival rates to vary request loads

In our workload generator

- VN topology requests
 - VN request arrivals
 - **Duration**
 - Lead time
 - Pr(ticket expiry)
 - Pr(ticket surrender)
 - Pr(lease surrender)
 - ...
- No duration information from the Emulab trace
 - Assume **Gamma distribution**
 - 1) Used to model demand distribution in queuing systems
 - 2) VN request durations are skewed and heavy-tailed [Jelena et.al.' 12]

In our workload generator

- VN topology requests
 - VN request arrivals
 - Duration
 - **Lead time**
 - Pr(ticket expiry)
 - Pr(ticket surrender)
 - Pr(lease surrender)
 - ...
- Emulab supports no reservation
 - No information available
 - Assume Gamma distribution

In our workload generator

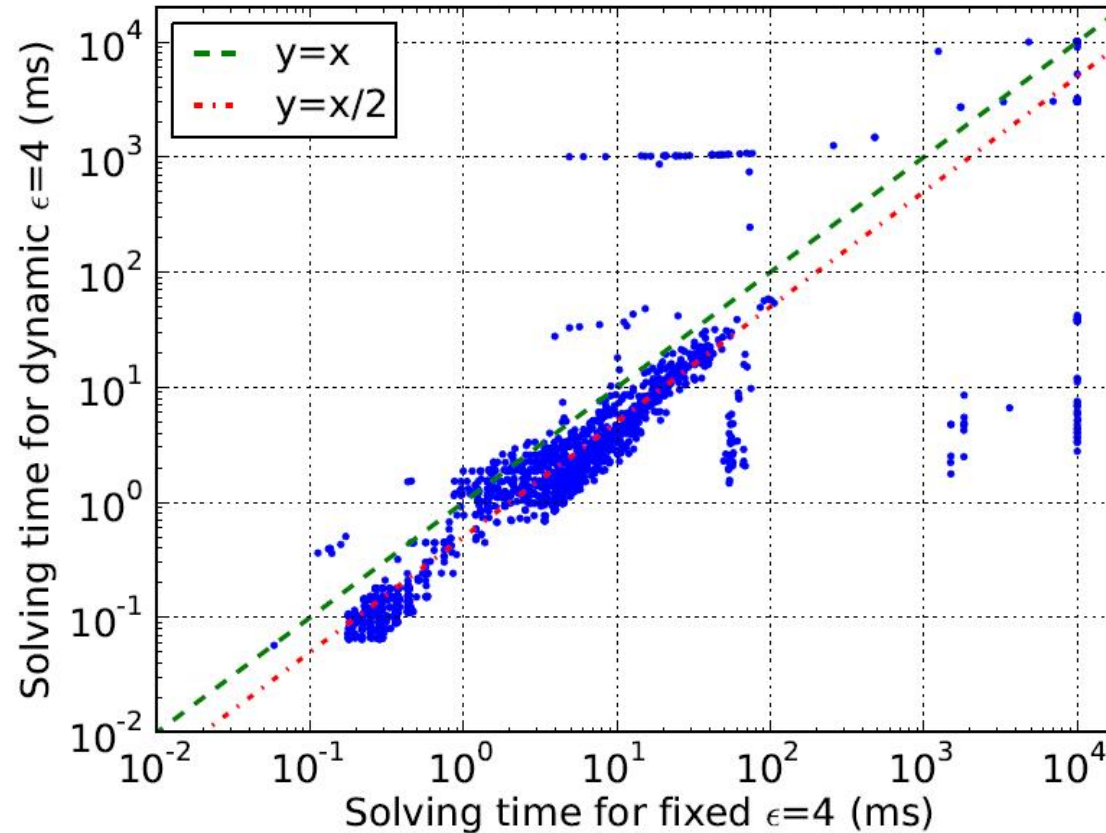
- VN topology requests
 - VN request arrivals
 - Duration
 - Lead time
 - Pr(ticket expiry)
 - Pr(ticket surrender)
 - Pr(lease surrender)
 - ...
- No information available
 - Simpler assumption:
 - No tickets are expired
 - No tickets surrendered
 - No leases surrendered

Comparing VN mapping algorithms



- General approach
 - Various topology requests
 - Various physical networks
 - Possible criteria: solving time, scalability, revenue-to-cost ratio, etc.
- Our experimental setting:
 - VN requests sampled from real Emulab request stream
 - Physical network based on the largest ProtoGENI site
 - Criteria: solving time

Comparing two VNM algorithms



Details in Tridentcom12 paper

Investigating dynamic allocation behavior

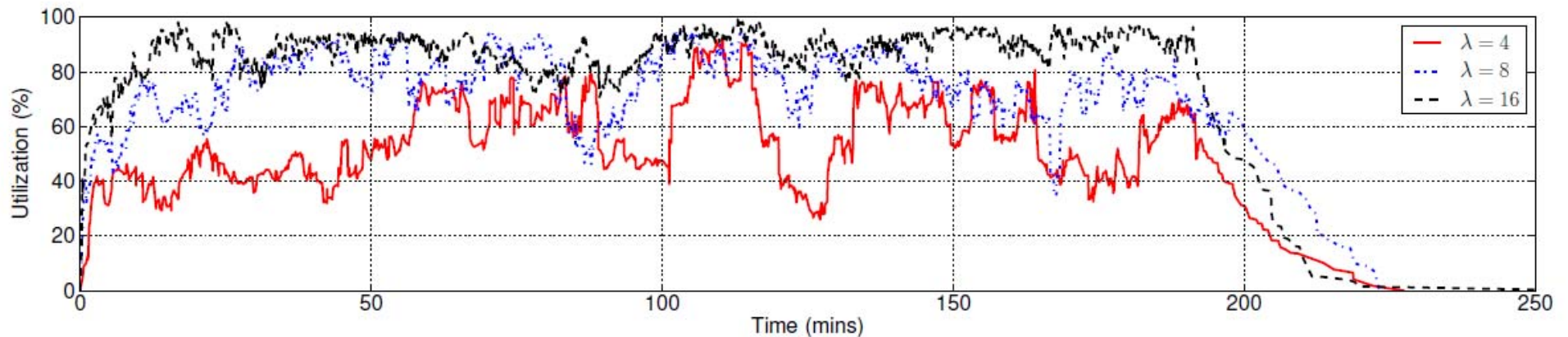


- Sequence of requests
 - Resource reservation requests
 - Redeem requests
 - Release requests
- Physical networks
- Example criteria:
 - Resource utilization
 - Percentage of successful mappings
 - etc.

Dynamic behavior under varying loads

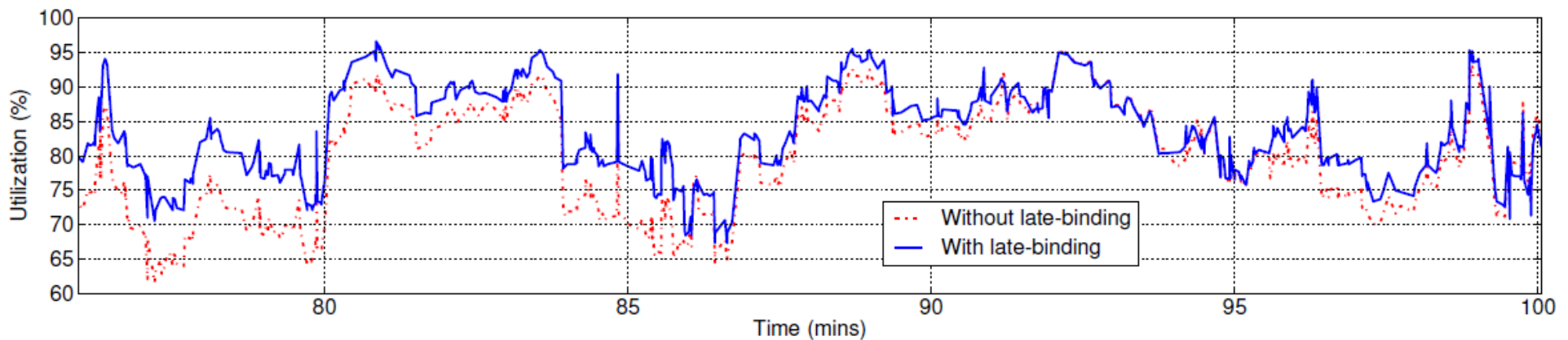


- Use distributions to annotate VN requests with
 - When to request/release resources
- Physical network based on the largest ProtoGENI site
- Criteria: resource utilization



Evaluating resource allocation strategies

- Use distributions to annotate VN requests with
 - When to reserve/allocate/release resources
- Physical network based on the largest ProtoGENI site
- Criteria: resource utilization



Discussion



- Our first step towards realistic benchmarks for virtual network resource allocation
- Limitations
 - Built from one testbed workload trace
 - Assumes some distributions due to lack of information
- Open questions
 - How much do testbed workloads change over time?
 - How to validate the distribution assumptions?
 - How better this is compared to synthetic workloads?
 - How applicable it is to the cloud computing?
 - ...

Discussion



- Our first step towards realistic benchmarks for virtual network resource allocation
- Limitations
 - Built from one testbed workload trace
 - Assumes some distributions due to lack of information
- Open questions
 - How much do testbed workloads change over time?
 - How to validate the distribution assumptions?
 - How better this is compared to synthetic workloads?
 - How applicable it is to the cloud computing?
 - ...

Many thanks!