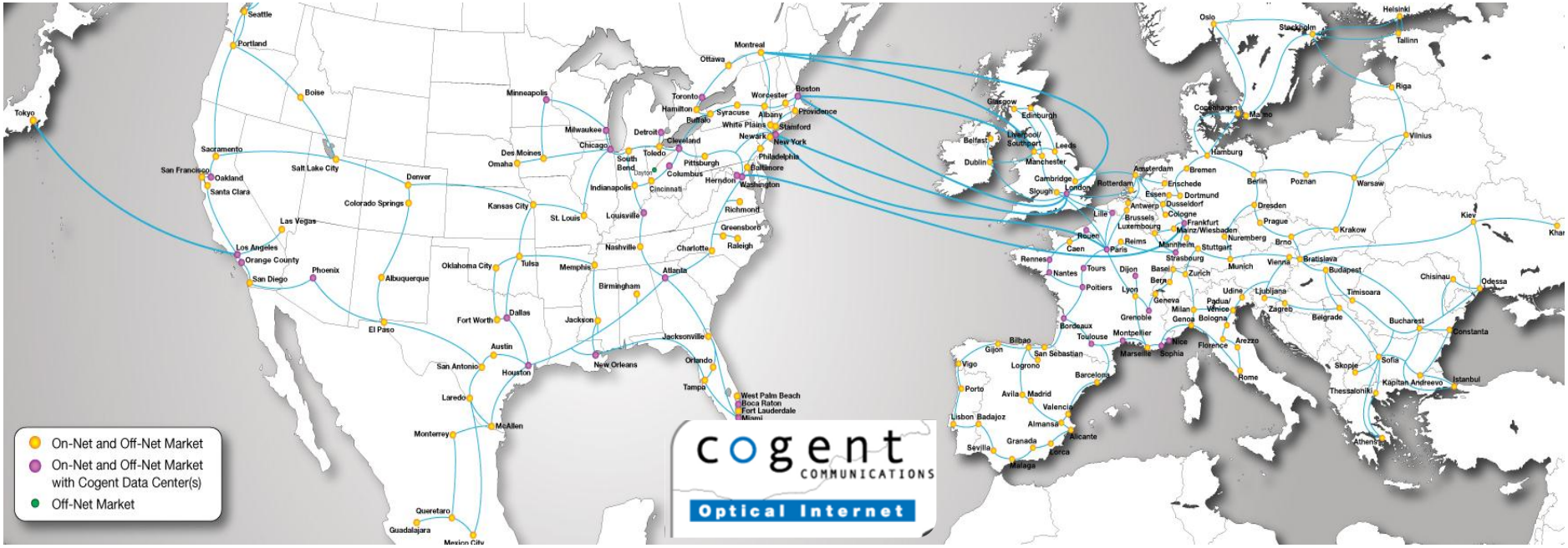
When NFV Meets SDN: a Short Circuit or Sparkling Fireworks?

Danny Raz

Bell Labs Israel (Technion)

Based on joint work with: Rami Cohen, Lian Lewin-Eytan, Ariel Orda, Amir Nahir, and Seffi Naor

The Network


On-Net and Off-Net Market
On-Net and Off-Net Market with Cogent Data Center(s)
Off-Net Market
cogent COMMUNICATIONS
Optical Internet
Seattle
Portland
Boise
Tokyo
Sacramento
San Francisco
Oakland
Santa Clara
Salt Lake City
Denver
Colorado Springs
Las Vegas
Los Angeles
Orange County
San Diego
Phoenix
Albuquerque
Oklahoma City
Tulsa
Kansas City
St. Louis
Minneapolis
Des Moines
Omaha
Chicago
Milwaukee
Detroit
Cleveland
Toledo
South Bend
Indianapolis
Louisville
Nashville
Memphis
Birmingham
Atlanta
Charlotte
Raleigh
Greensboro
Richmond
Pittsburgh
Columbus
Cincinnati
Herndon
Washington
Baltimore
Philadelphia
New York
Newark
Stamford
White Plains
Albany
Syracuse
Buffalo
Hamilton
Toronto
Ottawa
Montreal
Boston
Worcester
Providence
Dallas
Fort Worth
Jackson
El Paso
Austin
San Antonio
Houston
New Orleans
Jacksonville
Orlando
Tampa
West Palm Beach
Boca Raton
Fort Lauderdale
Miami
Laredo
McAllen
Monterrey
Queretaro
Guadalajara
Mexico City
Oslo
Stockholm
Helsinki
Tallinn
Riga
Vilnius
Copenhagen
Malmo
Glasgow
Edinburgh
Belfast
Liverpool/ Southport
Leeds
Dublin
Manchester
Hamburg
Amsterdam
Bremen
Berlin
Poznan
Warsaw
Cambridge
Slough
London
Rotterdam
Enschede
Essen
Dortmund
Dusseldorf
Antwerp
Brussels
Luxembourg
Cologne
Frankfurt
Mainz/Wiesbaden
Dresden
Prague
Krakow
Kiev
Lille
Rouen
Caen
Paris
Reims
Mannheim
Stuttgart
Nuremberg
Brno
Strasbourg
Vienna
Bratislava
Budapest
Rennes
Nantes
Tours
Dijon
Basel
Bern
Zurich
Munich
Chisinau
Odessa
Poitiers
Lyon
Geneva
Udine
Ljubljana
Timisoara
Padua/ Venice
Milan
Genoa
Bologna
Zagreb
Belgrade
Bucharest
Grenoble
Bordeaux
Toulouse
Montpellier
Nice
Florence
Arezzo
Constanta
Bilbao
Gijon
San Sebastian
Marseille
Sophia
Sofia
Skopje
Vigo
Logrono
Barcelona
Rome
Kapitan Andreevo
Istanbul
Porto
Avila
Madrid
Valencia
Thessaloniki
Lisbon
Badajoz
Almansa
Alicante
Granada
Sevilla
Cordoba
Malaga
Athens

The Network

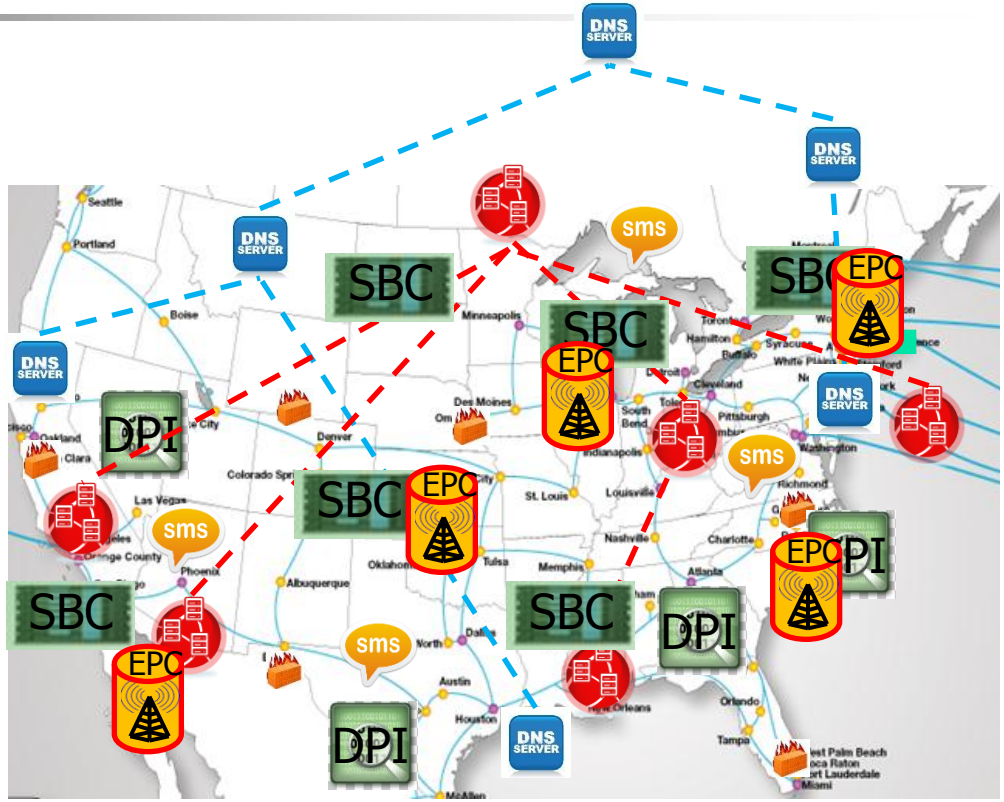
- Basically
  - Transport information from place to place
  - Transport bits from place to place
  - Transport packets from place to place

The Network

- Basically
  - Transport information from place to place
  - Transport bits from place to place
  - Transport packets from place to place
- Actually
  - People can talk (video-conf)
  - People can text (or Whatsapp)
  - Communities can be formed
  - Machines can share state
  - Applications can …. (real time traffic, public transportation, ….)

The Network

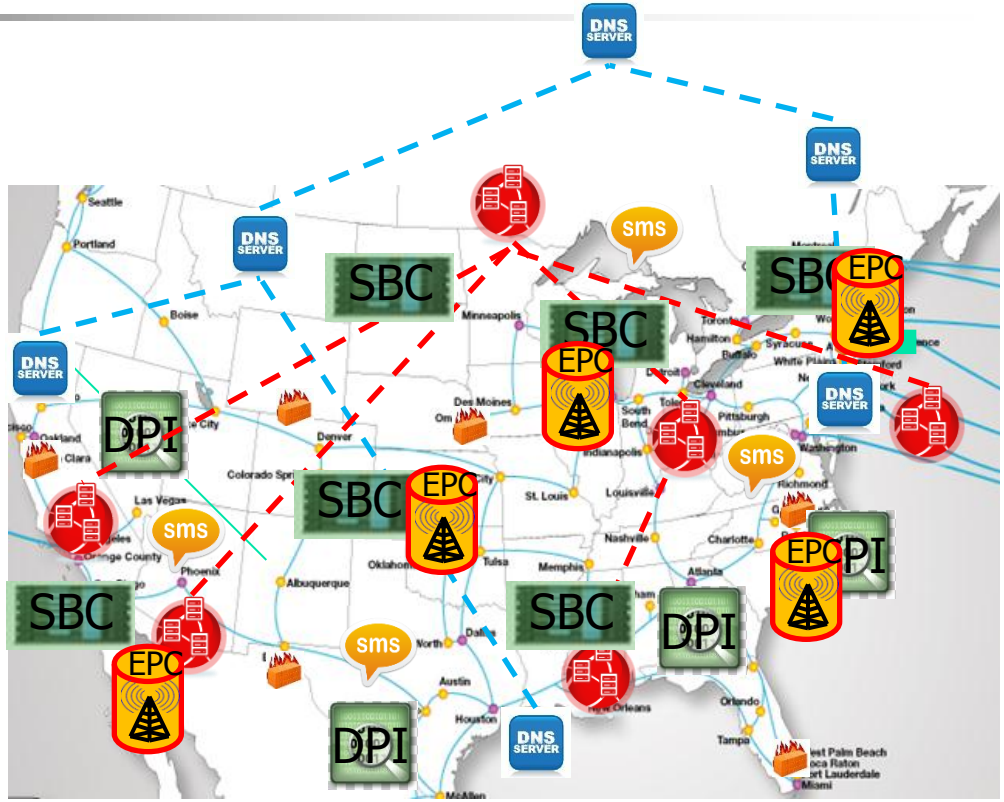
- Much more than just
  - Transport packets from place to place
- Actually
  - People can talk (video-conf)
  - People can text (or whatsup)
  - Communities can be formed
  - Machines can share state
  - Applications can …. (real time traffic, public transportation, ….)



PCE LTE TE
PDN-GW S-GW
SGSN/GGSN
SIP NAT RSVP

The Network is a service

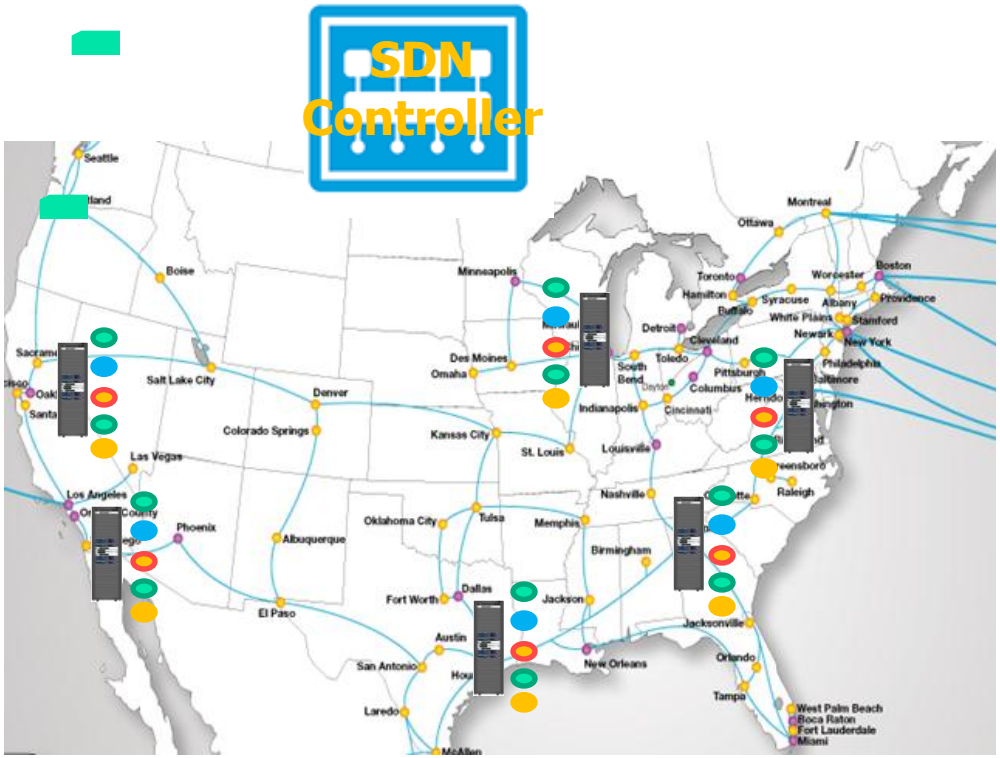
- A Network Service
  - Composed of one or more network functions
  - Service function chaining
- Currently
  - Functions (and services) are implemented via dedicated hardware located on the flow path



The Network is a service

- A Network Service
  - Composed of one or more network functions
  - Service function chaining

- Distributed Cloud Networking
  - Functions (and services) are implemented on COST servers located at mini) data centers distributed within the network
  - Traffic is send to these servers using the control mechanism of SDN

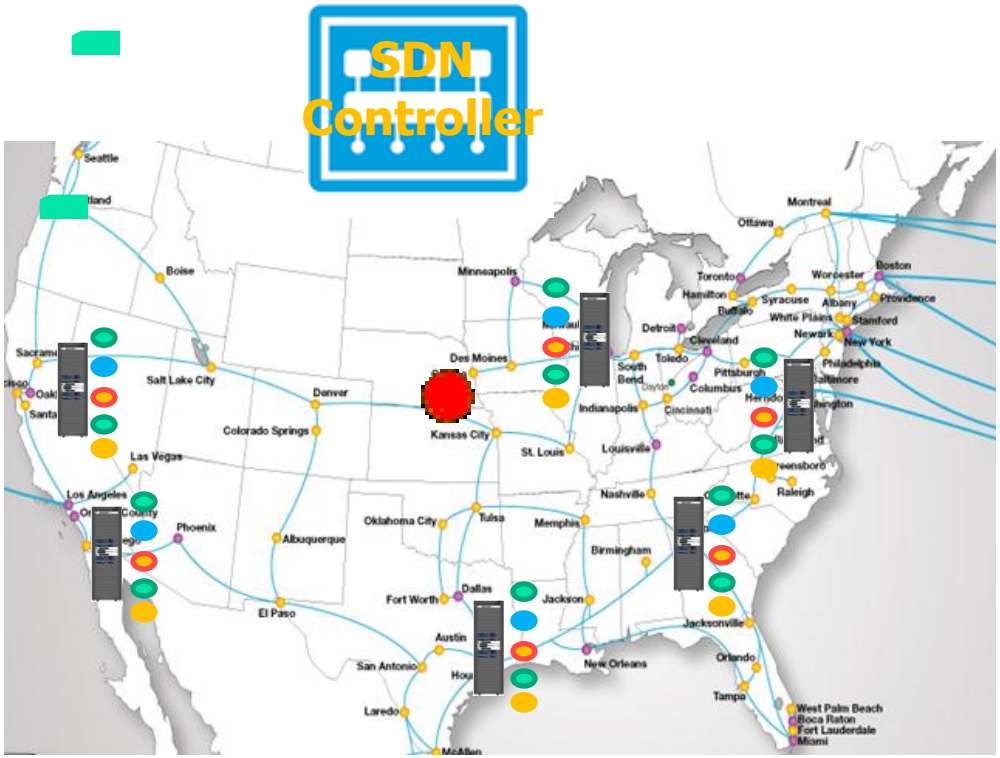


PCE LTE TE
PDN-GW S-GW
SGSN/GGSN
SIP NAT RSVP

Distributed Cloud Networking

NFV + SDN

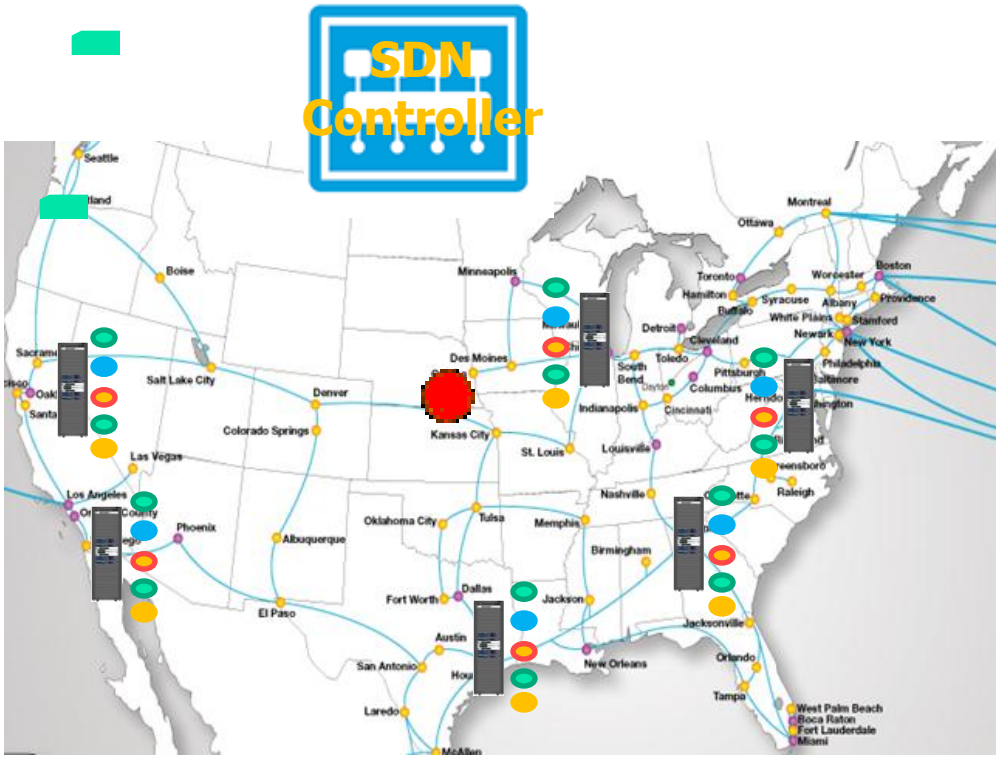
- Lots to gain
  - Use COTS silicon - Reduced Capex
  - Easy provisioning - reducing time to market
  - Easier operation – reduced Opex



Distributed Cloud Networking

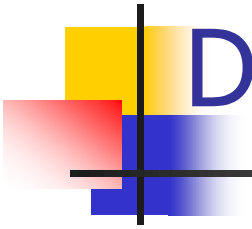
NFV + CDN

- Not so easy
  - Use COTS silicon – What about performance
  - Easy provisioning? – managing and orchestrating distributed clouds
  - Easier operation? – elasticity and QoS

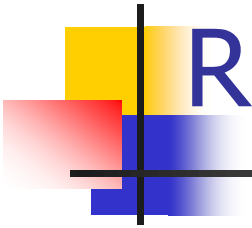


PCE LTE TE
PDN-GW S-GW
SGSN/GGSN
SIP NAT RSVP

Distributed Cloud Networking



- This is what this talk is all about:
  - Reduce the barriers so as to realize the promises of the new era of networking
  - Try to identify ***game changer*** research challenges and address them

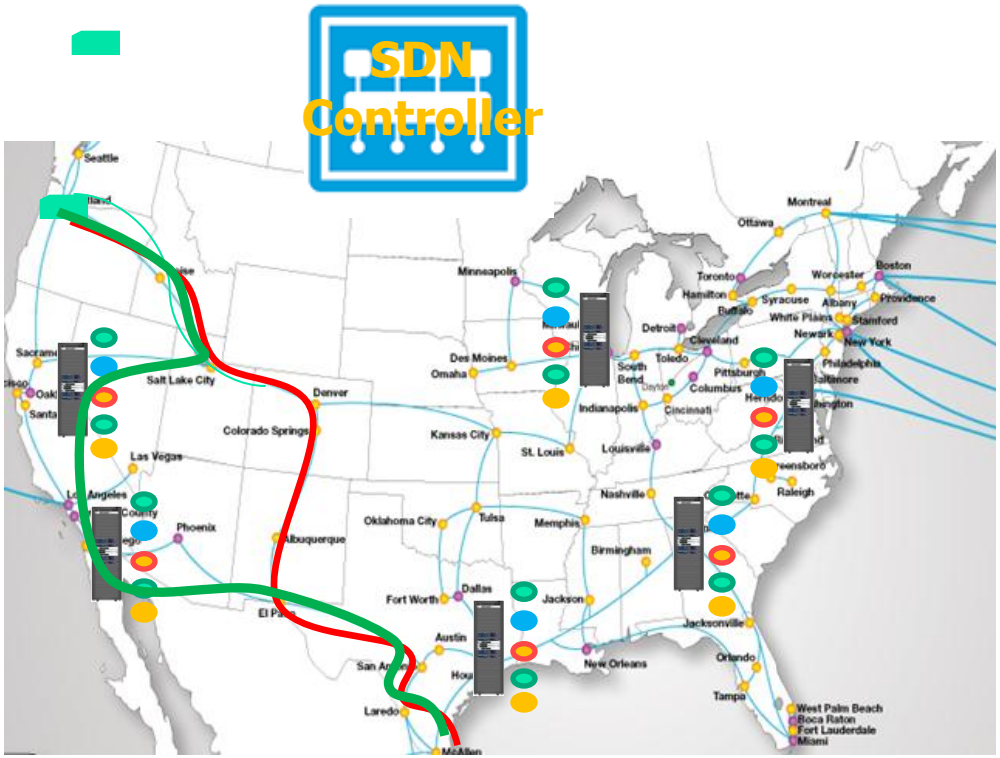


Rest of the talk

- Two concrete examples:
  - placement of network functions
  - fully distributed elasticity
- And then open problems
  - research scope characterization (time-line, applicability)
  - specific problems

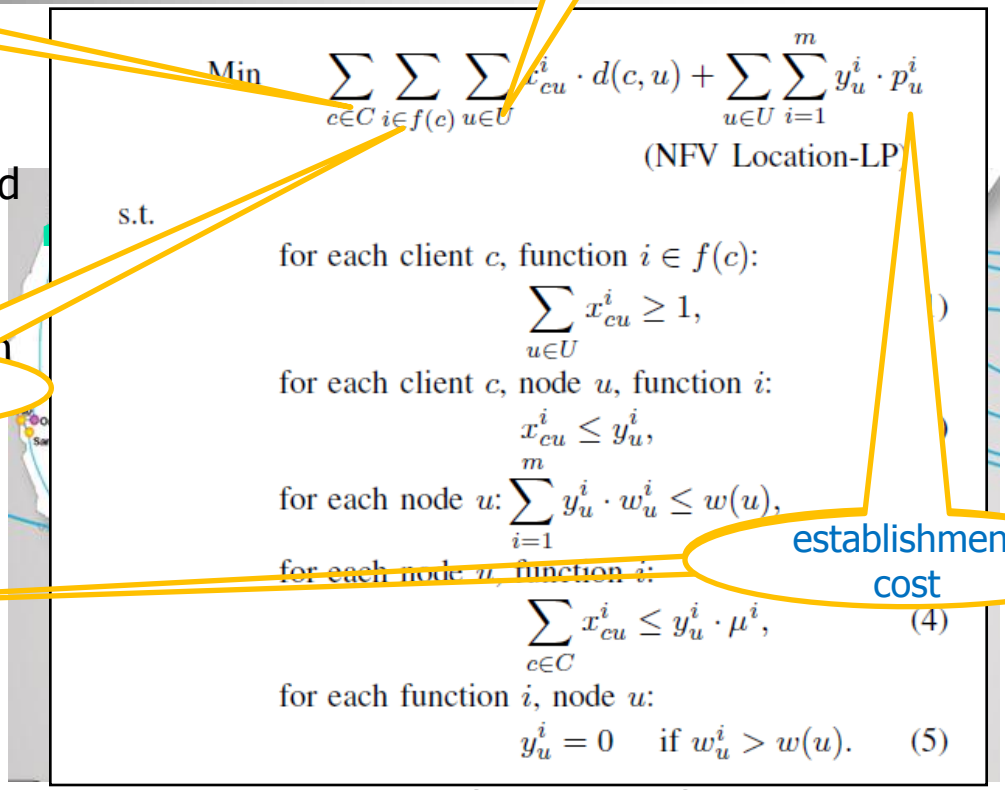
Near Optimal Placement of Network Functions

- Distributed Cloud Networking
  - Functions (and services) are implemented on COST servers located at mini) data centers distributed within the network
  - Traffic is send to these servers using the control mechanism of SDN
- Where to place each service
  - one place (globally) or
  - in each location , or
  - As Needed = depends on demand



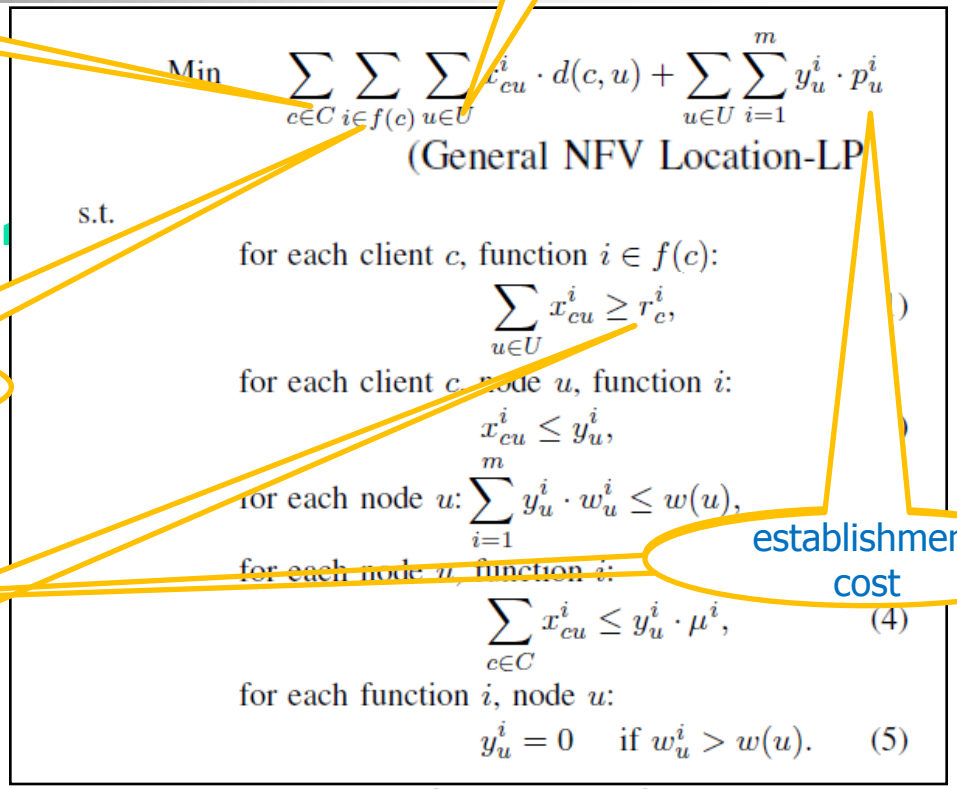
Model: problem definition

- Input
  - A set of flows, each with a path and a demand for each of the possible network functions.
  - A set of datacenters locations () with a size.
  - A set of network functions realizations , each with capacity (amount of clients to be served), size, and establishment cost .
- OUTPUT
  - A placement of copies of the realization of the network functions and a rerouting of the flow into the DCs such that:
    - The demand for each flow and for each function is satisfies, the size constraints are met, and the overall cost is minimal



Model: problem definition

- Input
  - A set of flows, each with a path and a demand for each of the possible network functions.
  - A set of datacenters locations () with a size.
  - A set of network functions realizations , each with capacity (amount of clients to be served), size, and establishment cost .
- OUTPUT
  - A placement of copies of the realization of the network functions and a rerouting of the flow into the DCs such that:



$$\text{Min} \quad \sum_{c \in C} \sum_{i \in f(c)} \sum_{u \in U} x^i_{cu} \cdot d(c,u) + \sum_{u \in U} \sum_{i=1}^{m} y^i_u \cdot p^i_u$$

(General NFV Location-LP)

s.t.

for each client c, function $i \in f(c)$:

$$\sum_{u \in U} x^i_{cu} \geq r^i_c,$$

for each client c, node u, function i:

$$x^i_{cu} \leq y^i_u,$$

for each node u: $\sum_{i=1}^{m} y^i_u \cdot w^i_u \leq w(u)$,

for each node u, function i:

$$\sum_{c \in C} x^i_{cu} \leq y^i_u \cdot \mu^i, \qquad (4)$$

for each function i, node u:

$$y^i_u = 0 \quad \text{if } w^i_u > w(u). \qquad (5)$$

node = DC location

flows=clients

i is a network function

establishment cost

flow's demand for function *i*

- The demand for each flow and for each function is satisfies, the size constraints are met, and the overall cost is minimal

Main theoretical result

- notes
  - If there is only one network function then this problem is actually the well known facility location problem .
  - If there are no network distances this problem reduces to the well known generalized assignment problem (GAP).

Min $$\sum_{c\in C}\sum_{i\in f(c)}\sum_{u\in U} x^i_{cu}\cdot d(c,u) + \sum_{u\in U}\sum_{i=1}^{m} y^i_u \cdot p^i_u$$

(General NFV Location-LP)

s.t.

for each client c, function $i \in f(c)$:
$$\sum_{u\in U} x^i_{cu} \geq r^i_c, \quad (1)$$

for each client c, node u, function i:
$$x^i_{cu} \leq y^i_u, \quad (2)$$

for each node u: $$\sum_{i=1}^{m} y^i_u \cdot w^i_u \leq w(u), \quad (3)$$

for each node u, function i:
$$\sum_{c\in C} x^i_{cu} \leq y^i_u \cdot \mu^i, \quad (4)$$

for each function i, node u:
$$y^i_u = 0 \quad \text{if } w^i_u > w(u). \quad (5)$$

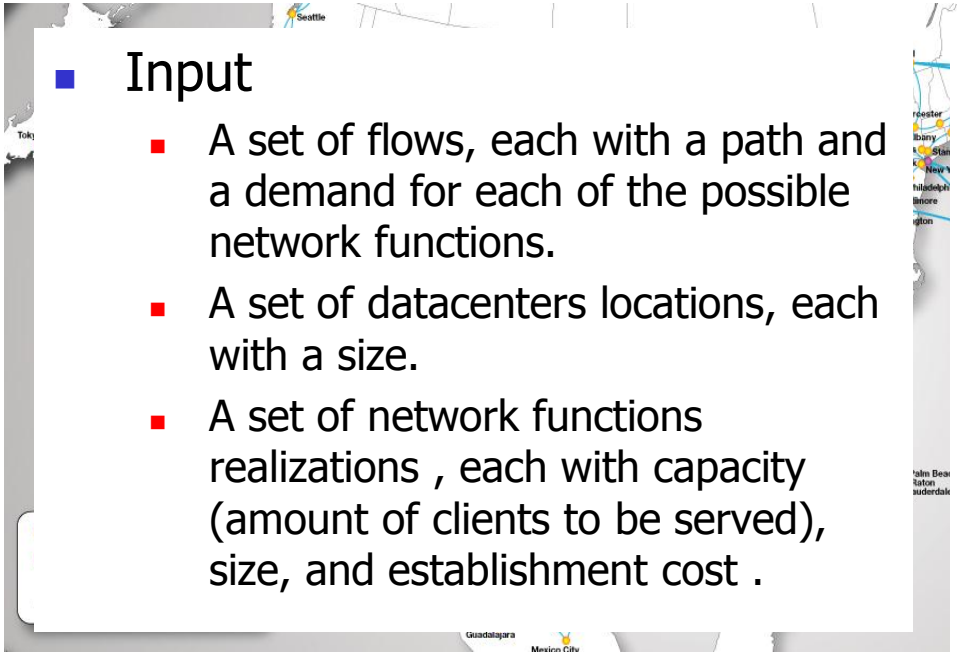
Theorem:

There exists a bi-criteria (O(1), O(1)) approximation algorithm for the General NFV location problem

Experimental evaluation

This network covers:

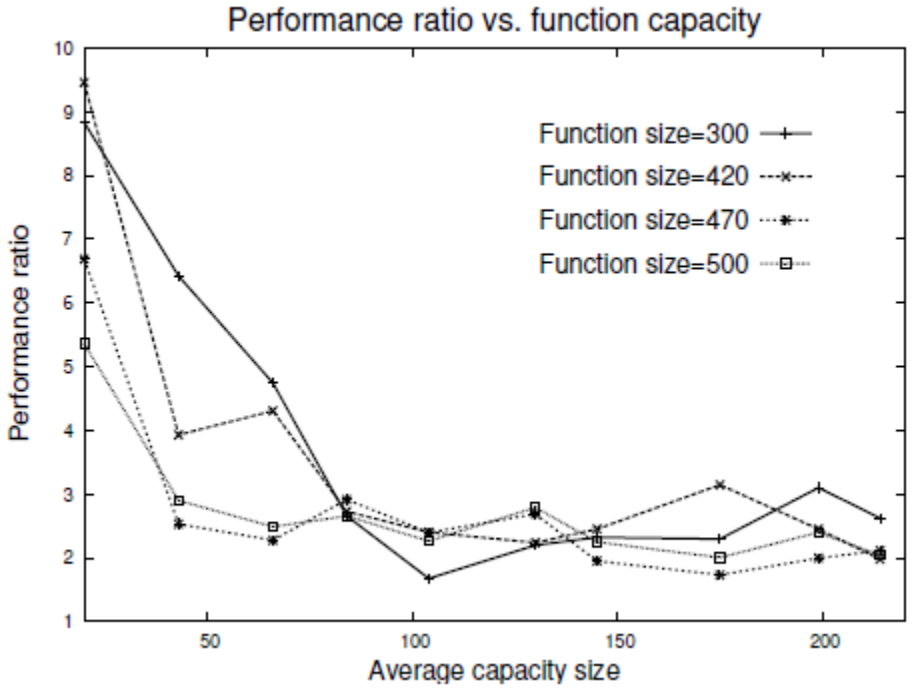
- 195 access locations (mostly within Europe and North America), about 260 links and almost 40 data centers



- Input
  - A set of flows, each with a path and a demand for each of the possible network functions.
  - A set of datacenters locations, each with a size.
  - A set of network functions realizations , each with capacity (amount of clients to be served), size, and establishment cost .

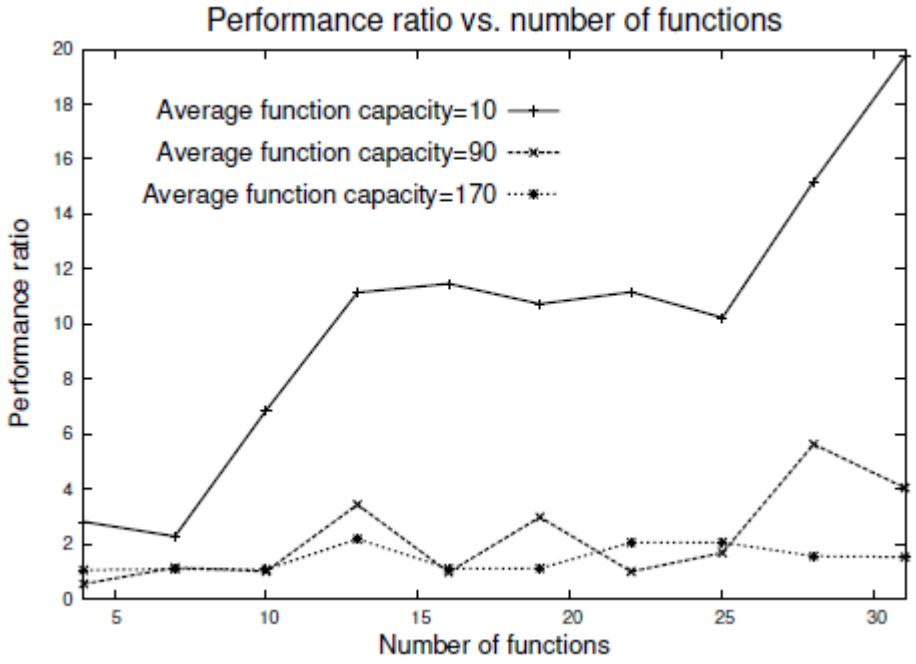
- selected 400 random pairs of (source, destination), and determined a shortest path between each source and destination, unit demand per flow.
- Each such flow is associated with 1-4 network functions that were chosen randomly from a set of 30.
- The size of a network function varies.
- The size of data center was randomly selected in the range 200-500.
- Opening cost was constant.

Experimental evaluation (2)



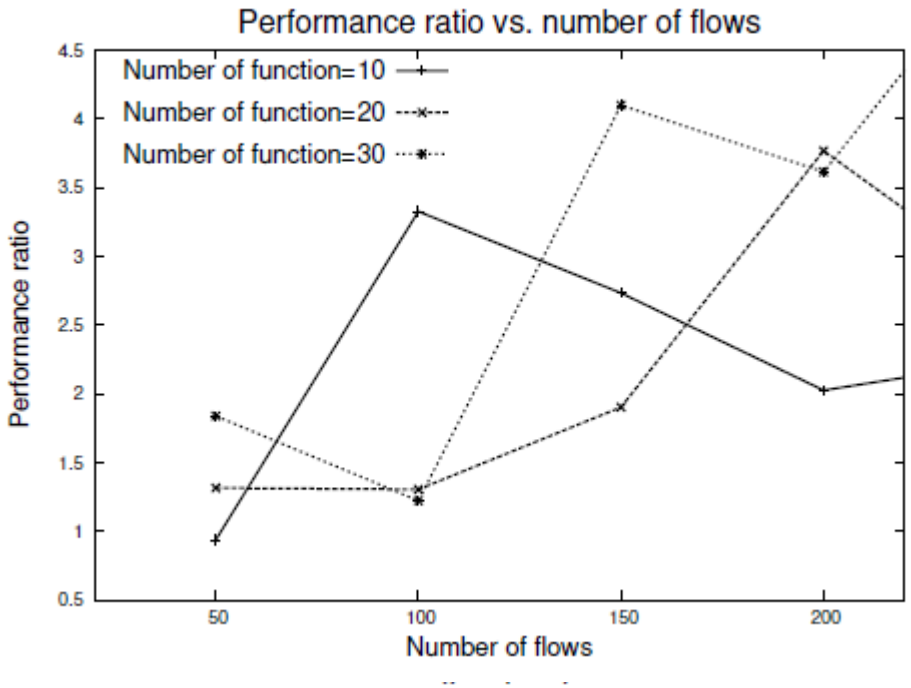
- Greedy
  - Go over all network function in an arbitrary order
  - For each such function
  - Find in a greedy way the best placement to satisfy the flows' demand

Experimental evaluation (3)



- Greedy
  - Go over all network function in an arbitrary order
  - For each such function
  - Find in a greedy way the best placement to satisfy the flows' demand

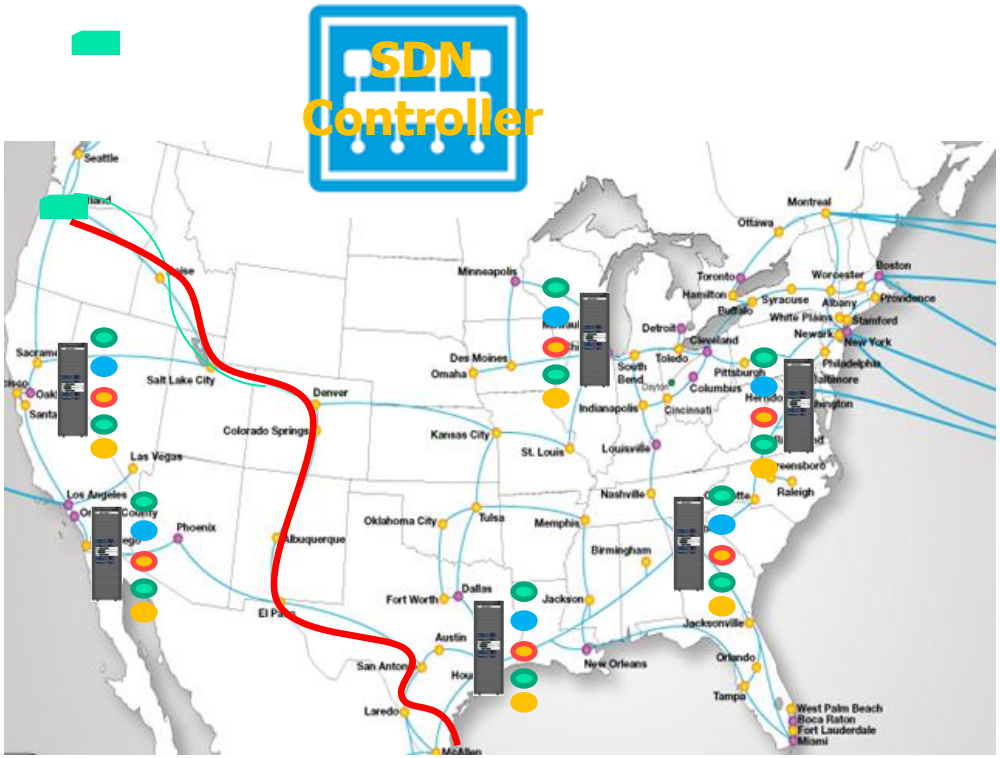
Experimental evaluation (4)



- Greedy
  - Go over all network function in an arbitrary order
  - For each such function
  - Find in a greedy way the best placement to satisfy the flows' demand

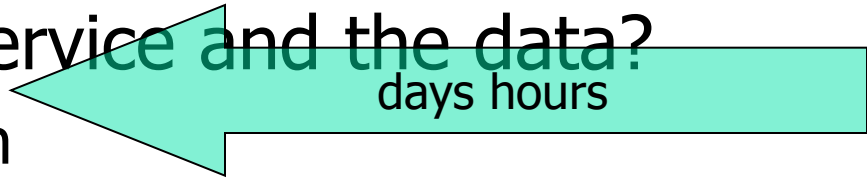
Conclusions

- Can model the important aspects of NFV placement
- Results depends on the settings
- More to do:
  - Service function chaining



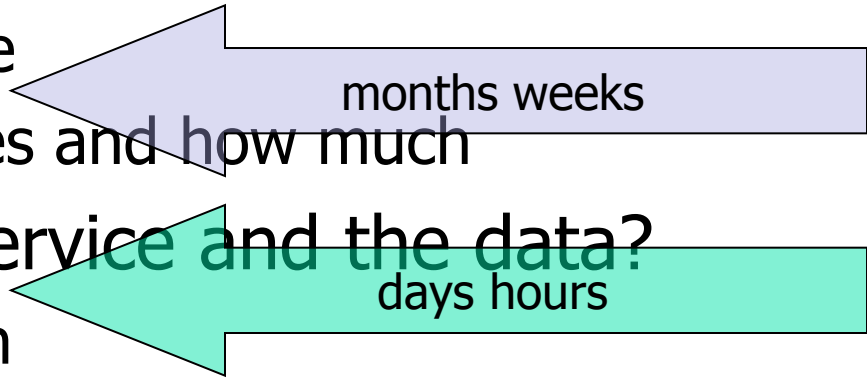
Resource allocation in the Cloud

- Where to acquire resources (CPU, Storage)?
  - building the next node
  - getting more resources and how much
- Where to place the service and the data?
  - service/data migration
- Which location should serve a specific request?
  - Load balancing



Resource allocation in the Cloud

- Where to acquire resources (CPU, Storage)?
  - building the next node
  - getting more resources and how much
- Where to place the service and the data?
  - service/data migration
- **How much resources are needed to provide the needed QoS for the current load?**
  - **elasticity**
- Which location should serve a specific request?
  - load balancing

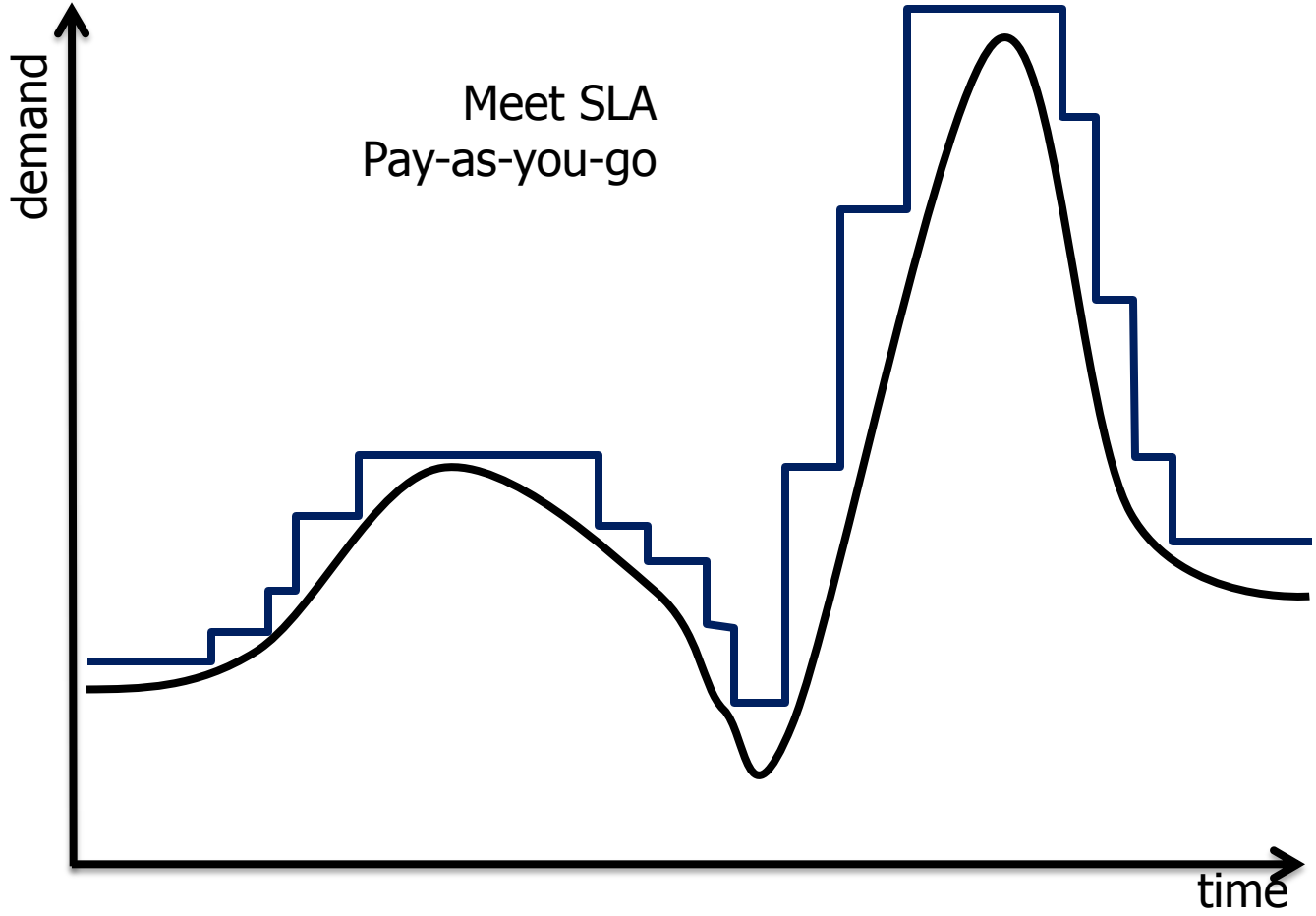


This part of the talk

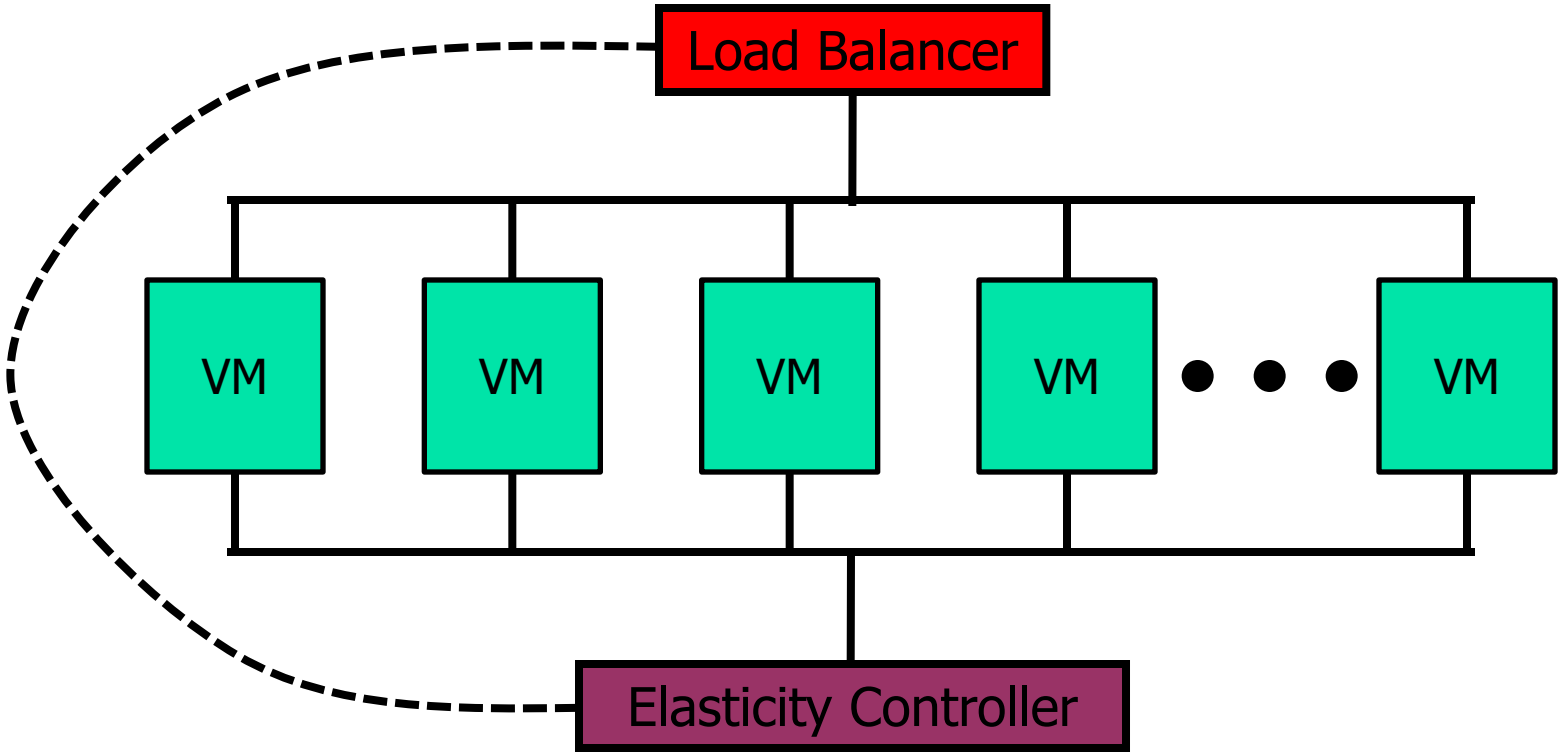
Can the amount of resources allocated for a large scale service adapt to the load in ***absence of centralized control?***

- *Scheme*
- *Analysis*
- *Evaluation*

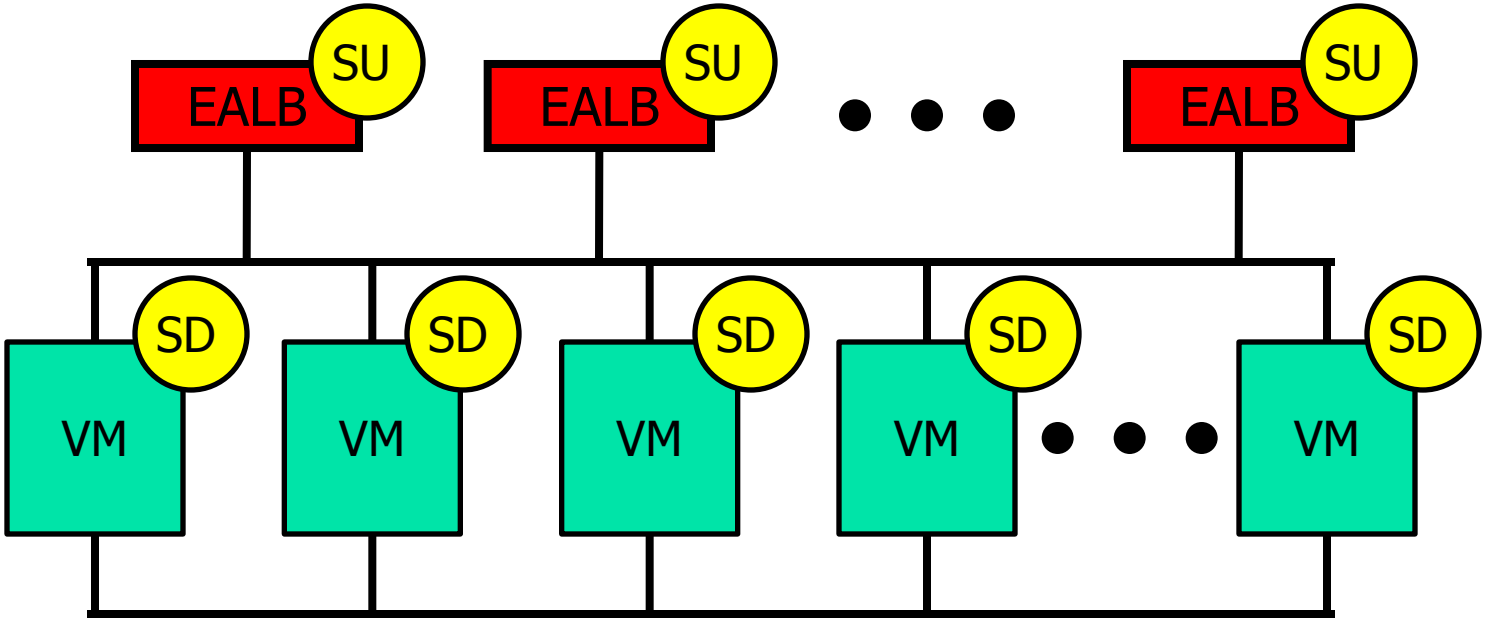
Elasticity



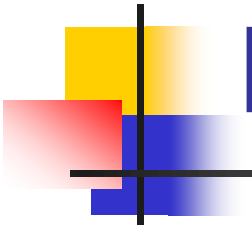
Standard Elasticity Architecture



Proposed Architecture



EALB – Elasticity-Aware Load Balancer
SU - Scale Up
SD - Scale Down

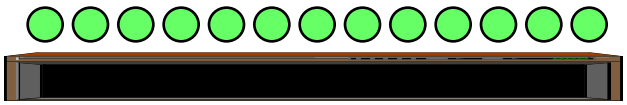


Key Aspects

- Load balancing
  - Fully distributed
- Scale-Up
  - Must take into account the time required to instantiate a new VM
- Scale-Down

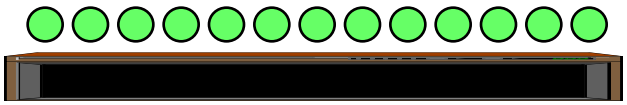
The power of choice

Balanced Allocations (1994) Yossi Azar, Andrei Z. Broder, Anna R. Karlin, Eli Upfal
SIAM Journal on Computing **(2000)**



- n balls to n buckets
- Random choice:
  - loaded bucket has (WHP) **O(*ln n / ln ln n*)** balls
  - probability[empty] $\rightarrow$ **1/e**

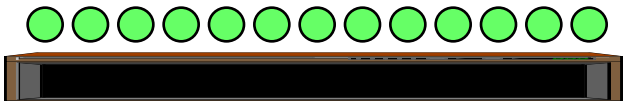
The power of choice (2)



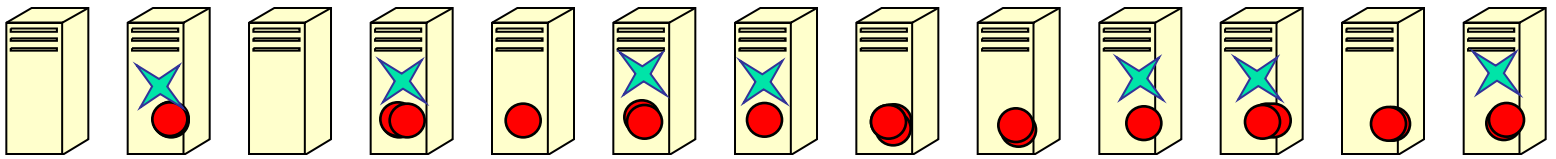
- ***n*** balls to ***n*** buckets
- Choose best out of 2:
  - loaded bucket has (WHP) **O(*log log n*)** balls
- Looking at more buckets improves linearly

The supermarket model

The Power of Two Choices in Randomized Load Balancing (1998) M. Mitzenmacher
IEEE Transactions on Parallel and Distributed Systems (2001)



- Replace buckets by M/M/1 queues
- Jobs arrive in a Poisson stream at rate λn
- Choose **best** out of d
  - $d=1$ is a random assignment
  - **best** means shortest queue
- Different from M/M/k

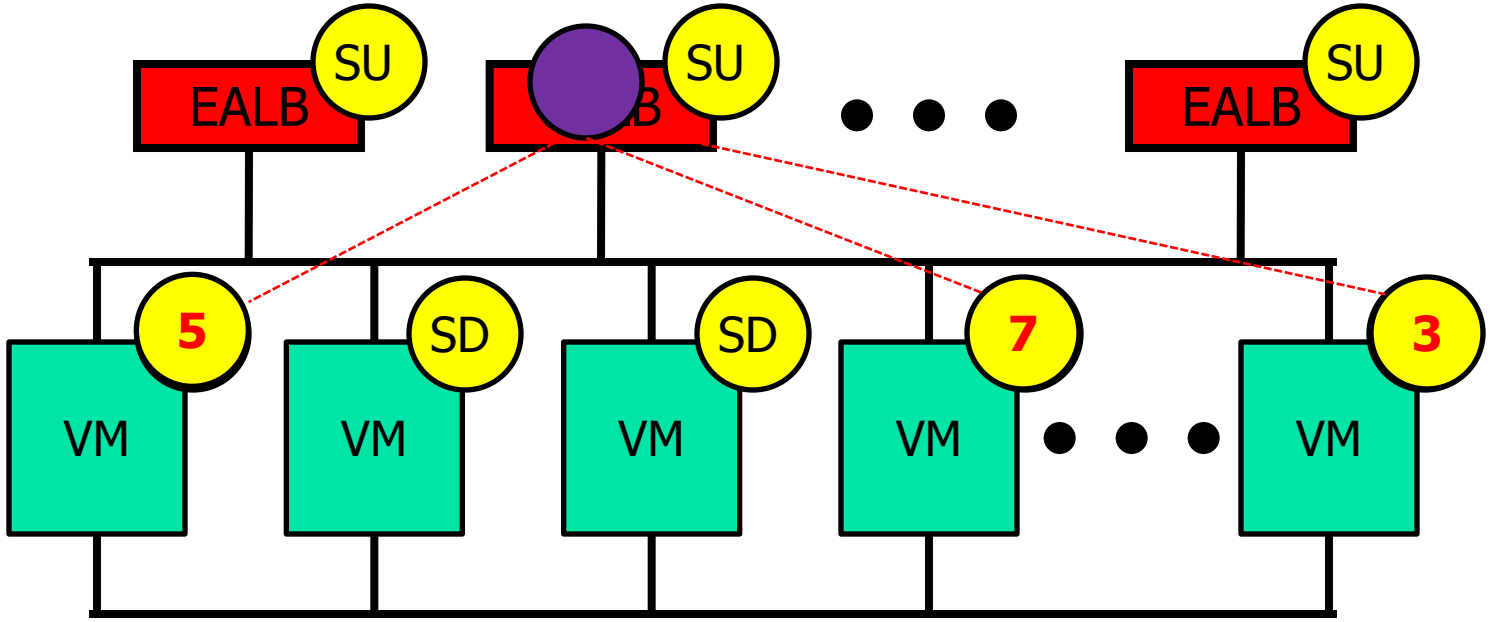


(Un)load Balancing - Distributed Packing

- We define T^P as the effective operating threshold for a VM
  - Counted in number of jobs in server (VM)
  - With high probability, a job arriving at a VM with T^P or less jobs, will meet SLA (more details later)

- Upon an arrival of a job to the EALB:
  - Sample *d* random VMs for their load (# of jobs)
  - If one or more VMs have T^P or less jobs, send to the **most** loaded VM in that class
  - Otherwise: send to the **least** loaded VM

Proposed Architecture



$T^P=6$

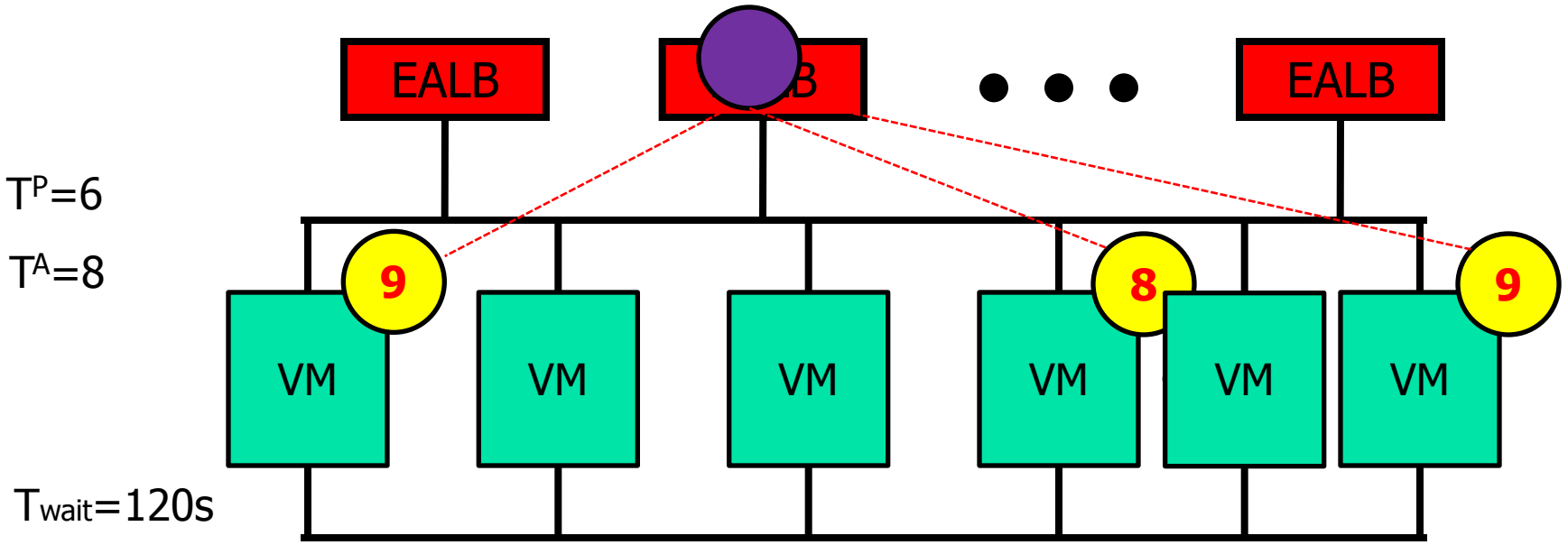
EALB – Elasticity-Aware Load Balancer
SU - Scale Up
SD - Scale Down

Scaling Policies

- Scale-Up:
  - If all *d* sampled VMs hold T^A or more jobs, instantiate a new VM
  - T^A should be greater than T^P
- Scale-Down:
  - If a VM is idle for T_{wait} time units, it self-terminates

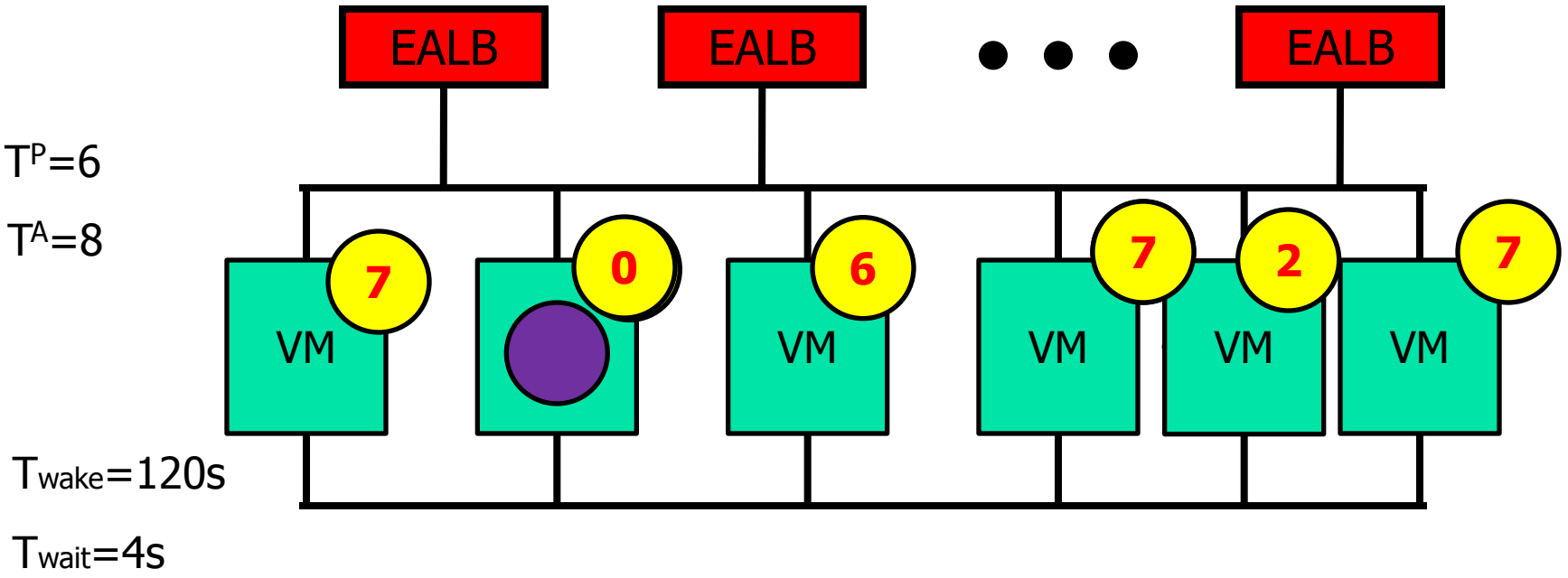
EALB – Elasticity-Aware Load Balancer
SU - Scale Up
SD - Scale Down

Proposed Architecture



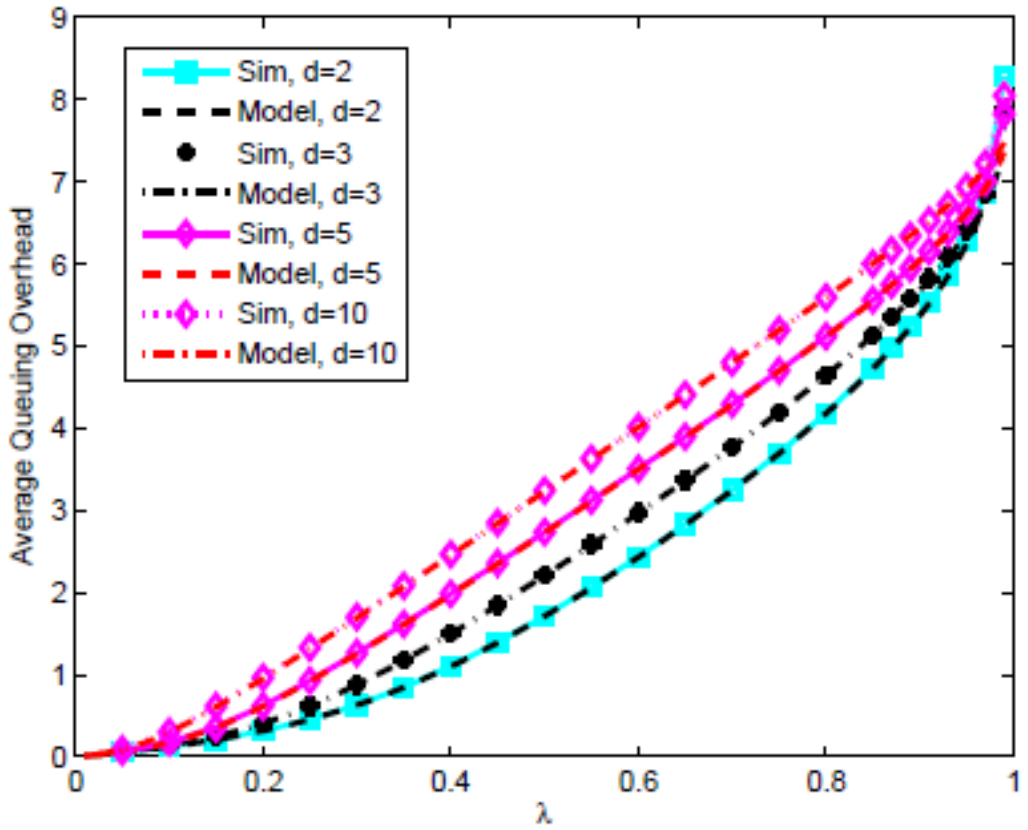
EALB – Elasticity-Aware Load Balancer
SU - Scale Up
SD - Scale Down

Proposed Architecture



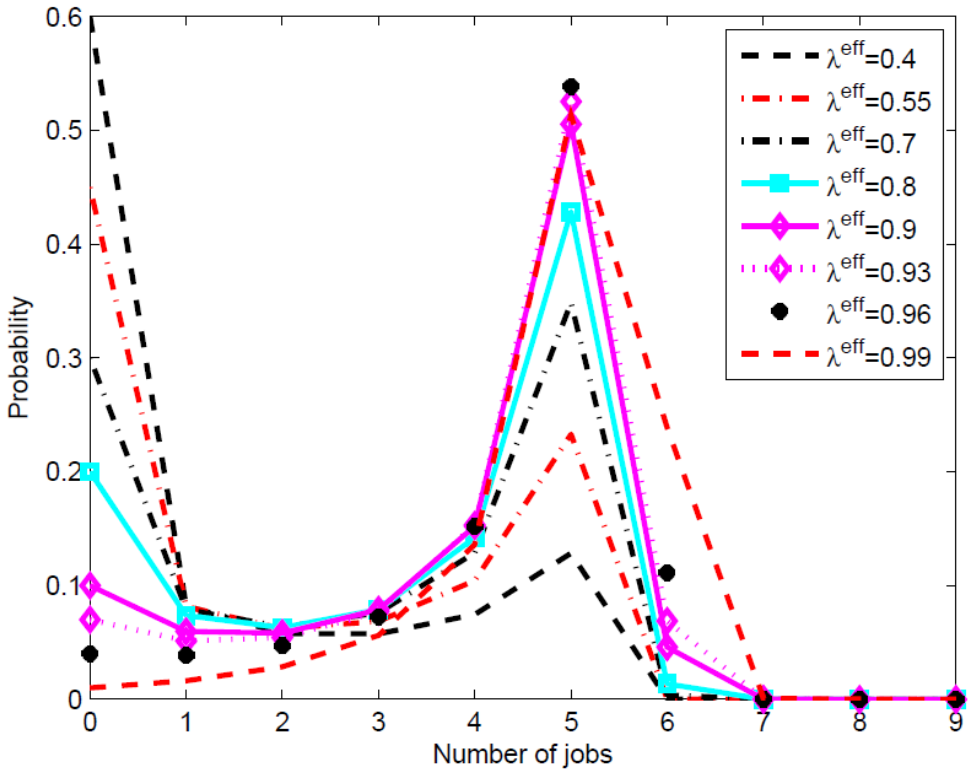
Analysis of a Static System

- *M*
  -
- Jc
  pr
- Vl
  di
  -



isson

Results



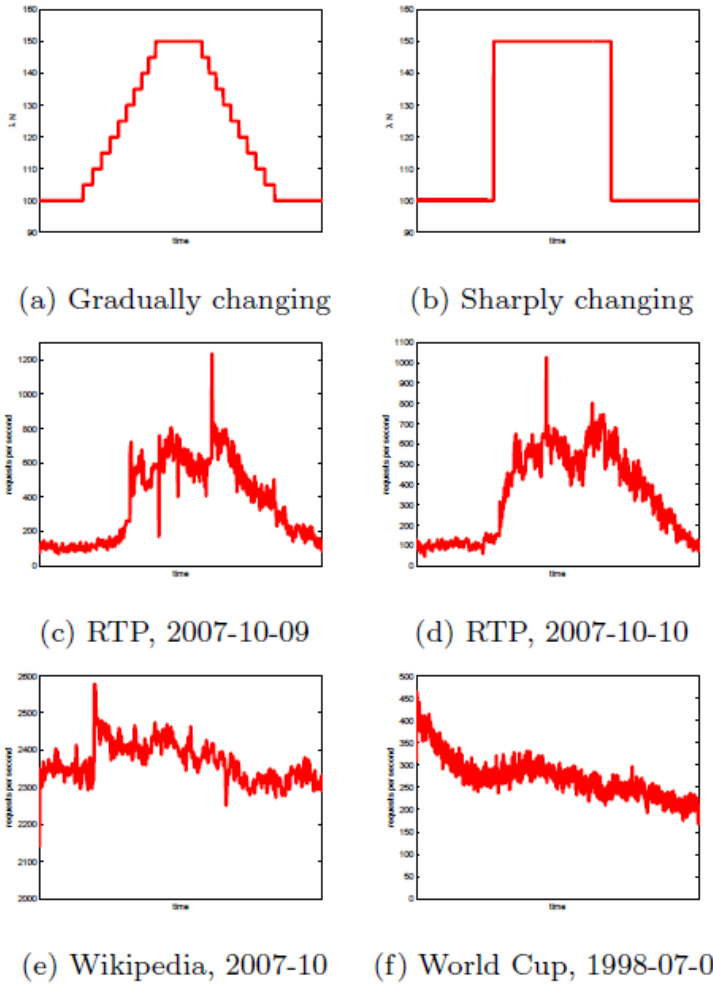
Probability of a VM holding a certain number of jobs, as a function of the number of jobs. M=100, T^p=5, d=5

The Dynamic System - Evaluation

- Obtained through simulations and an Amazon EC2-based implementation
- Tested with a variety of workloads
  - Different load change patterns
    - Synthetic
    - Trace-based
  - Different job processing distribution
    - Exponentially distributed
    - Social-network-like model

The Dynamic System - Evaluation

- Obtained through
  Amazon EC2-base
- Tested with a vari
  - Different load cha
    - Synthetic
    - Trace-based
  - Different job proc
    - Exponentially dist
    - Social-network-lik



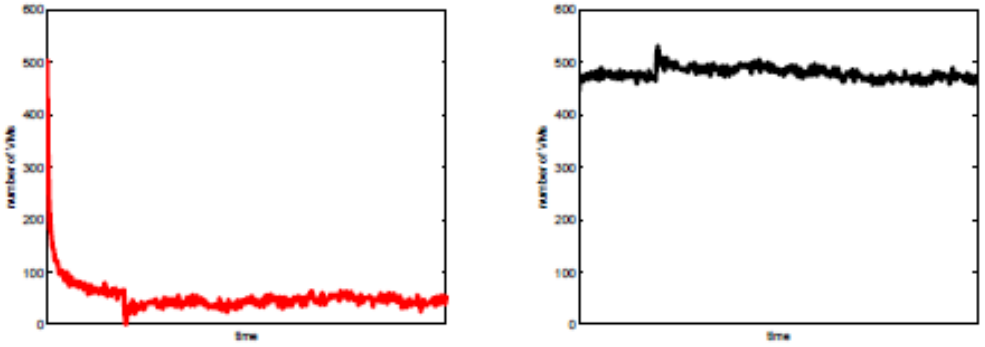
(a) Gradually changing (b) Sharply changing

(c) RTP, 2007-10-09 (d) RTP, 2007-10-10

(e) Wikipedia, 2007-10 (f) World Cup, 1998-07-09

39

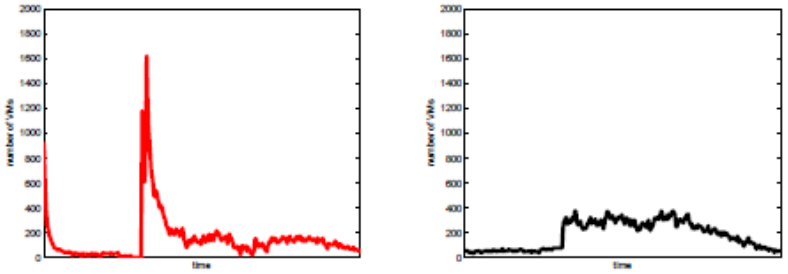
Results



(a) Idle (b) Busy

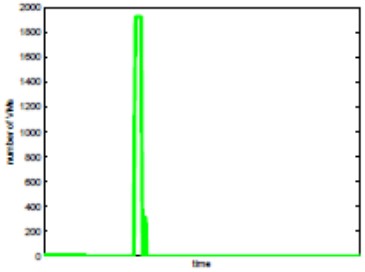
VM states when running with the Wikipedia trace

Results

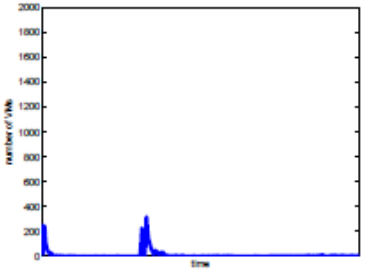


(a) Idle

(b) Busy



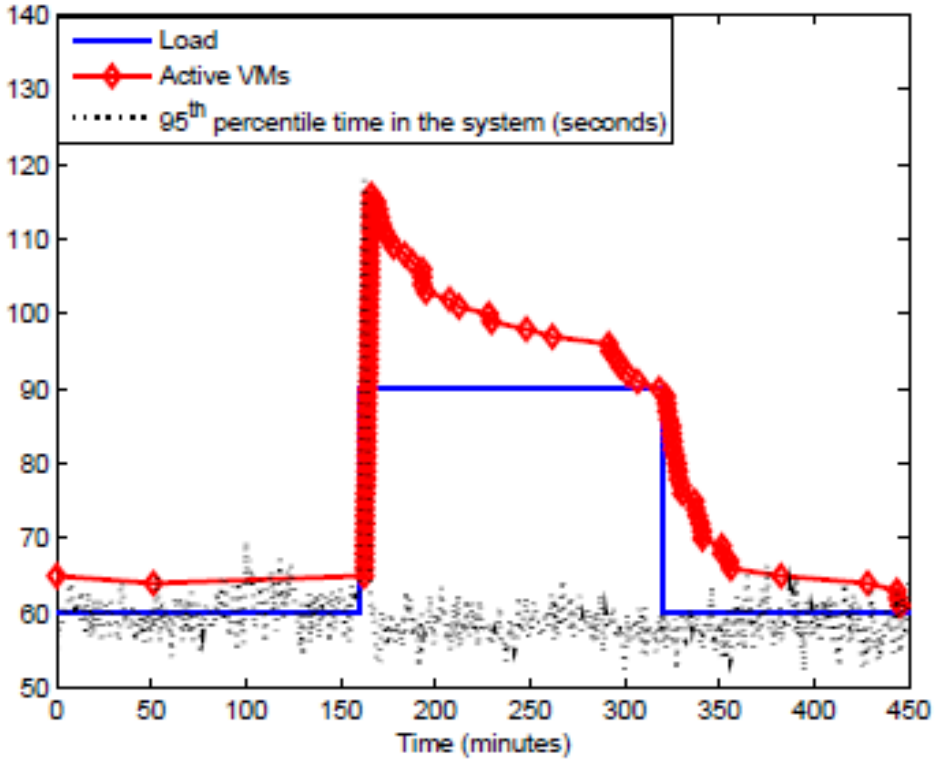
(c) Wakeup



(d) Shutdown

VM states when running with the RTP 2007-10-10 trace

Results



T^P=5, T^A=8, d=5, T_{wait}=160 sec, SLA target = 80 sec

Applicability to Power

- Data centers account for ~2% of world electricity usage

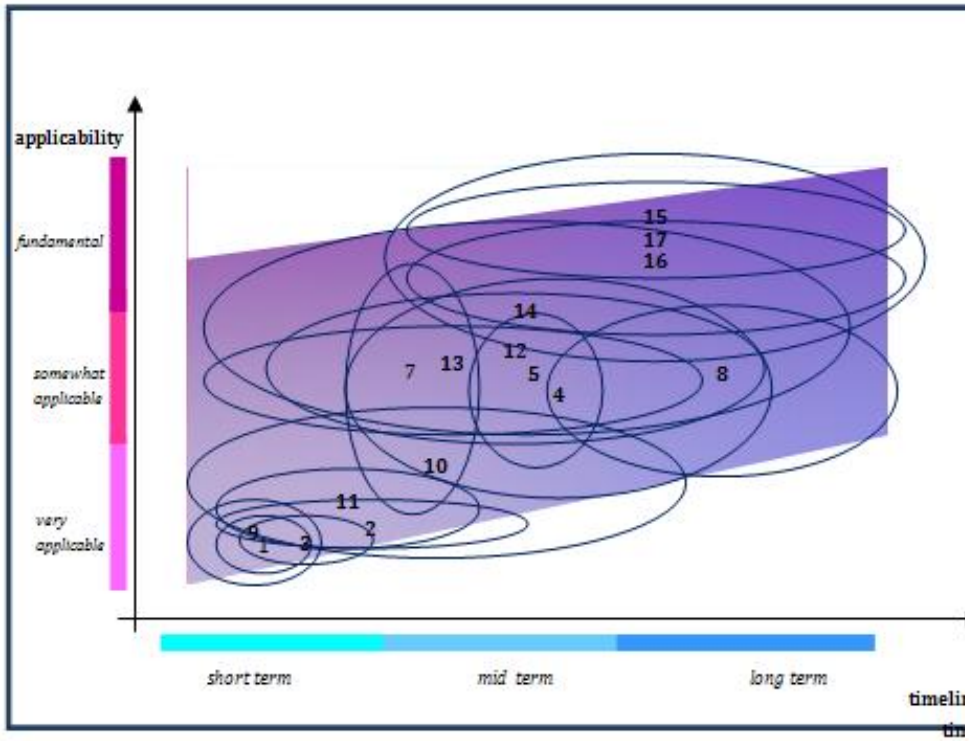
- The same methods can be used to:
  - Pack VMs on less physical hosts
  - While preserving the needed QoS
  - Can be done fully distributed

Summary and Future Work

- A fully scalable scheme for managing elasticity in the cloud
  - Simulated and tested in under real conditions
- Future work:
  - Analysis of the dynamic system
  - Heterogeneous VMs
  - Distributed growth-limit policies

Research challenges

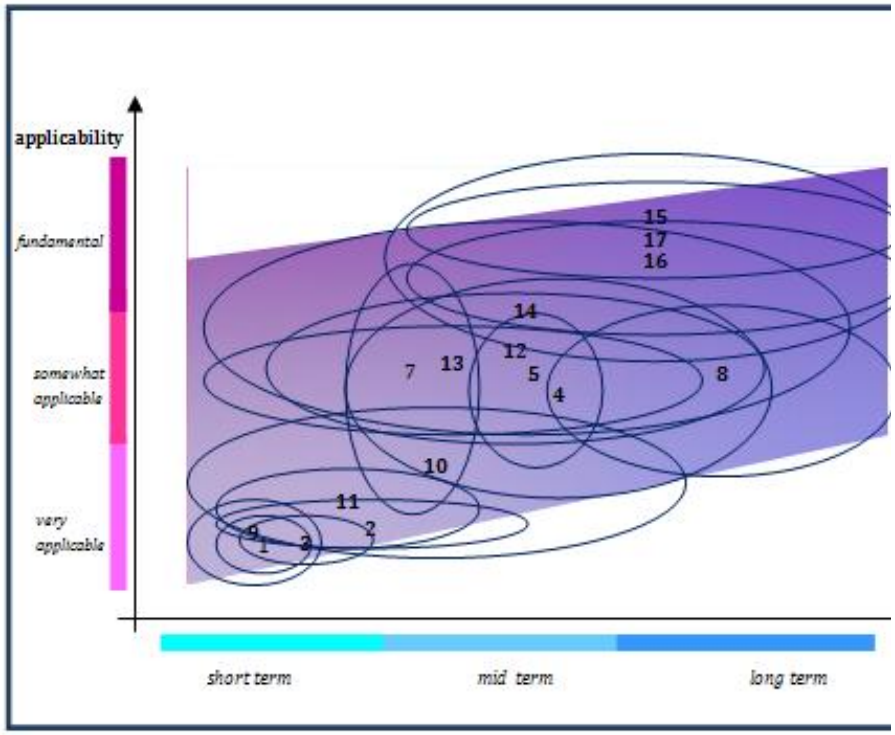
- Can be classified according to timeline and applicability



1. migration of network services into NFV
2. the way SDN is used for the control
3. VM pining to guarantee the required service level
4. distributed resource allocation
5. service automated adaptation
7. reliability and safety of network functions
8. applications that can benefit from the ability to distribute the cloud closer to the users and have guaranteed network performance.

Research challenges

- Can be classified according t applicability



9. the ability of OpenFlow to deliver the needed policy based control to the data plane
10. the amount of feedback and monitoring needed
11. guaranteeing performance of VM based applications
12. languages to describe services
13. development of environment aware software
14. resilient and security
15. a proper modeling for in-network computations that takes both the network as well as the computing aspects into consideration
16. the study of control mechanism and the interaction between several control mechanism that impact different aspects of the same system
17. studying the economic aspects of this new network paradigm

Main challenges (1)

- **Distributed resource allocation**
  - One of the key promises of the Distributed Cloud Networking paradigm is the ability to provide agile cost effective network services.
  - To do that, the operator is require to dynamically allocate the needed resources in an efficient way that will allow providing the needed services quality (globally) using the available resources that are distributed across the network.
  - This task becomes complex since it should cover both compute and networking resources, heterogynous resource types and different level of resource aggregation (from NUMA nodes to mini data center clusters).

Main challenges (1.2)

- **Distributed resource allocation**
  - This problem becomes more difficult due to the dynamic nature of the workload and the need for elasticity and agility which raises interesting research problems related to distributed load balancing, scaling, and the local state of the service component (the VM that executes the service).

Main challenges (2)

- **Efficient distributed monitoring**
  - The ability to provide an adequate resource allocation mechanism (as described above) is strongly based on acquiring the needed information both from the services (applications, network functions) and the infrastructure (compute and networking resources). This is a complex task that generates non-negligible overhead since monitoring information should be collected across the network to generate a global view. Efficient collection of the right amount of information and presenting it in an appropriate way both for the human and automatic control mechanisms is another interesting and important problem we plan to focus on.

Main challenges (2.1)

- **Efficient distributed monitoring**
  - Possible directions
    - Monitoring infrastructure and Applications
    - tools for the application owner
    - Environment aware applications
      - Renegotiate resource allocation

Main challenges (3)

- **Nested control mechanisms**
  - Orchestration of all the functions in the Distributed Cloud Networking paradigm is a very challenging task due to the large scale, geographical spread, the versatility of network functions and the their interdependencies, the dynamic nature of the workload, and the strive to get a cost effective solution.
  - The centralized management approach in which all information is to be analyzed by a single element that will make the optimal global decision may too complex and impractical.

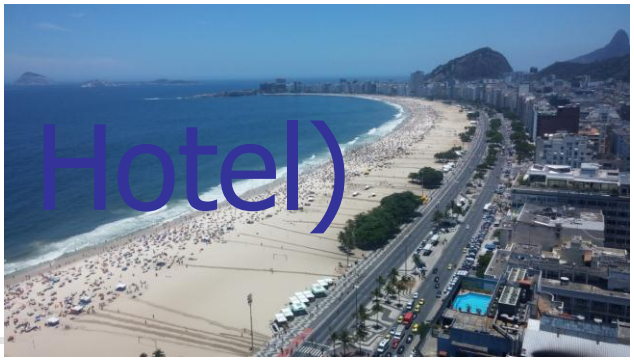
Main challenges (3.1)

- **Nested control mechanisms**
  - Thus, local decision engine may take local limited decisions and the overall global behavior is determined by the interaction among these control mechanisms.
  - Understanding the fundamental issues related to such a complex system and designing appropriate mechanisms for it is yet another area we plan to focus on.

Main challenges (4)

- **Cyber security**
  - Security is a major concern both for operators and network users. In the context of Distributed Cloud Networking, one can consider two different aspects of security.
    - The first one has to do with the vulnerability of the network and the ability to attack it. The shift from closed boxes to virtual machine running on commodity servers allows attackers an easier access to critical components and requires better security and safety mechanisms.
    - The second aspect has to do with the ability to use the new network architecture in order to allow novel techniques to deal with cyber security in general. For example one can use corralled monitoring and DPI in various locations to identify and fight threats.

Take home (to the Hotel)



- Distributed Cloud Networking is a revolution in networking (and Telco)
  - it is happening now
  - can dramatically change the network as we know it
- Many interesting and important research challenges
  - solutions can have an impact on real networks and their users (all of us)


Thank you