On Resource Negotiation in Application-aware Networks

Marco Savi, Antonio Marsico, Domenico Siracusa, **Elio Salvadori**

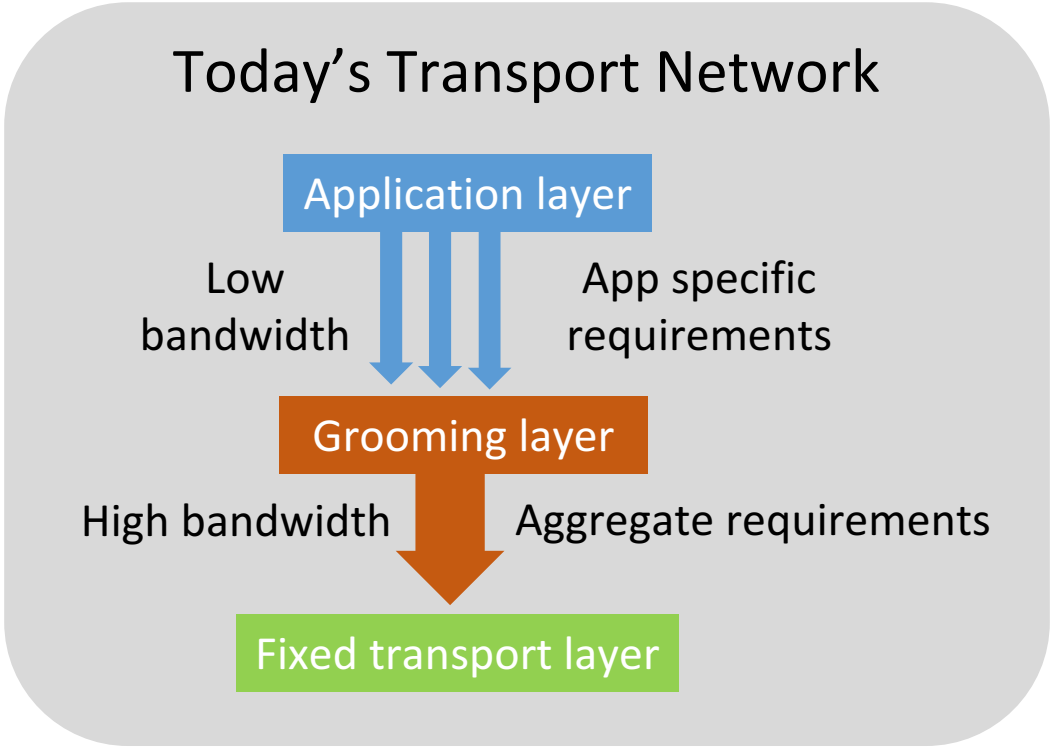
Fondazione Bruno Kessler

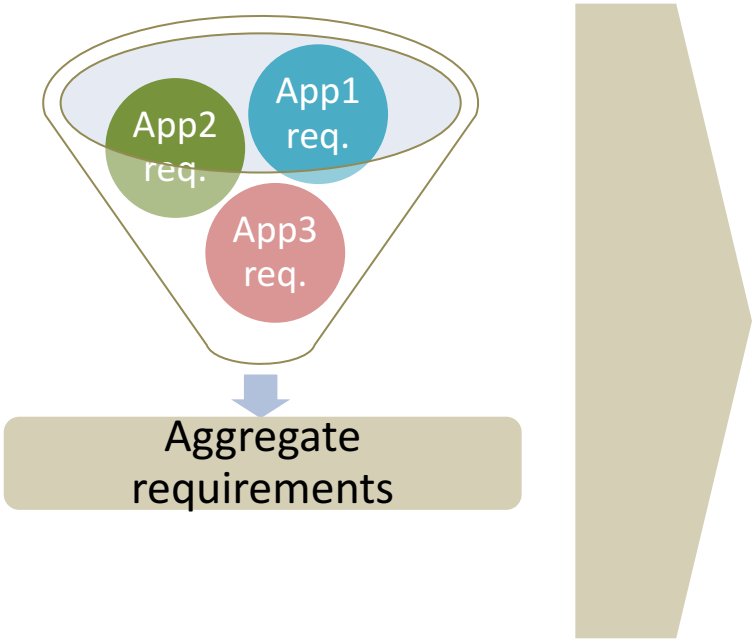
CREATE-NET Research Center

Trento – Italy

ONDM 2017

How today transport networks work

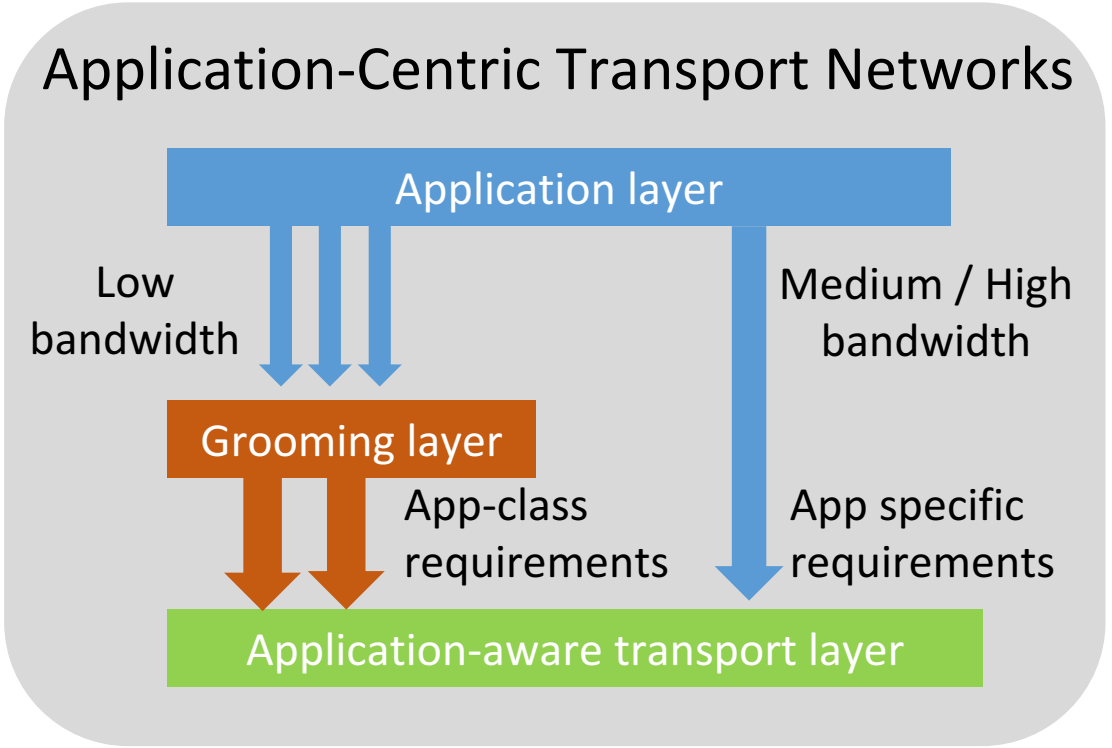




Problem: Transport networks should cater to the *heterogeneous requirements of modern applications* (bandwidth, latency, availability, etc.)

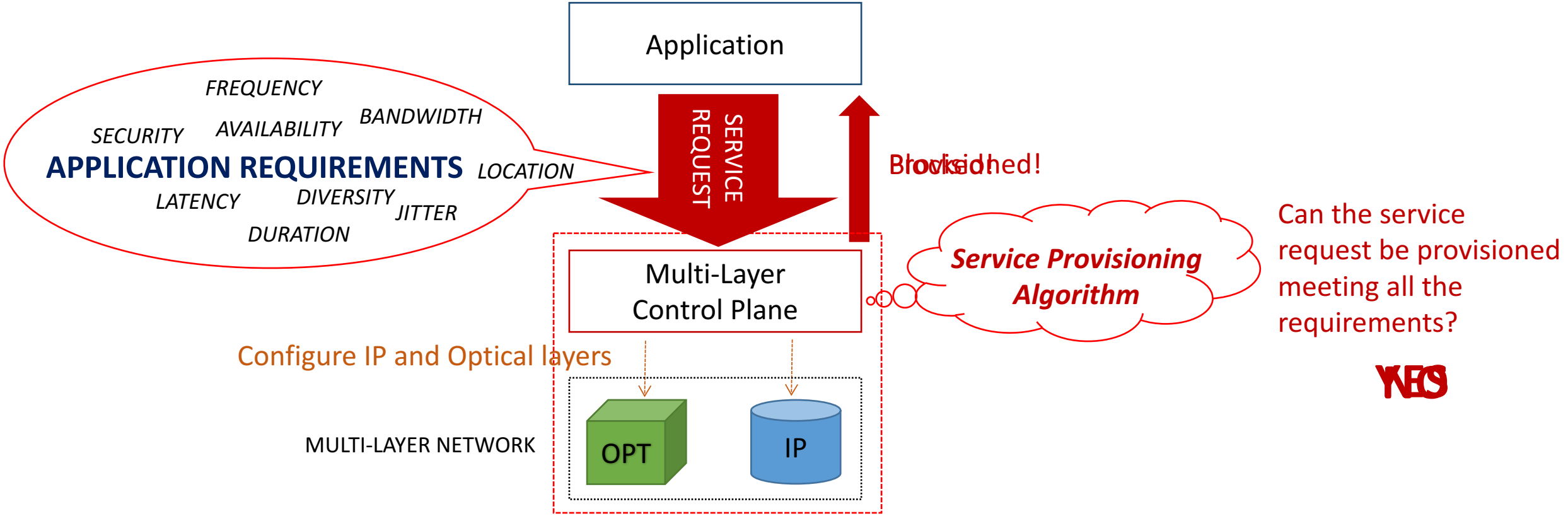
A step further: Application-Centric Networking

ACINO
www.acino.eu



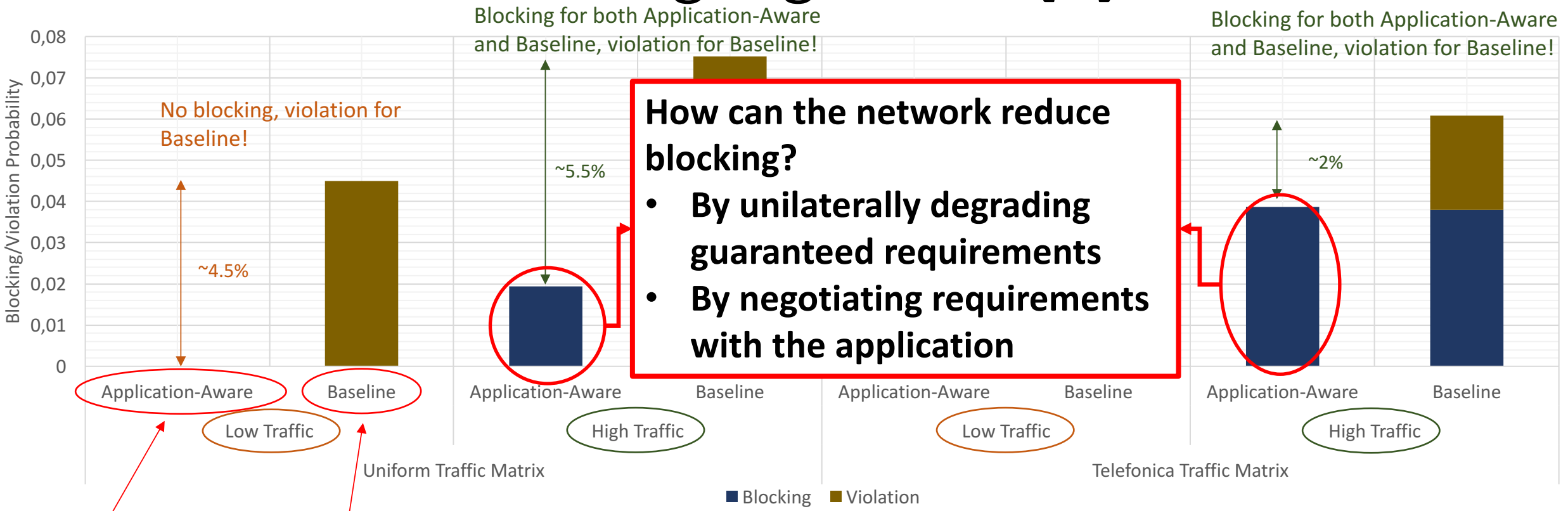
- Applies to multi-layer networks
- Traffic is differentiated according to application requirements

Application-Aware Service Provisioning



- No application requirement **violation** for application traffic accepted in the network
- If impossible to meet the requirements, an application *service request* is a-priori **blocked**
- Service requests are provisioned in an **application-aware** way

Performance of an Application-Aware Service Provisioning Algorithm [1]



How can the network reduce blocking?

- **By unilaterally degrading guaranteed requirements**
- **By negotiating requirements with the application**

Bandwidth, Latency, Availability requirements guaranteed

Only Bandwidth requirement guaranteed

Bandwidth = {1,10,100} Gbps uniform
Latency = {10,∞} ms uniform
Availability = {99.75,0} % uniform

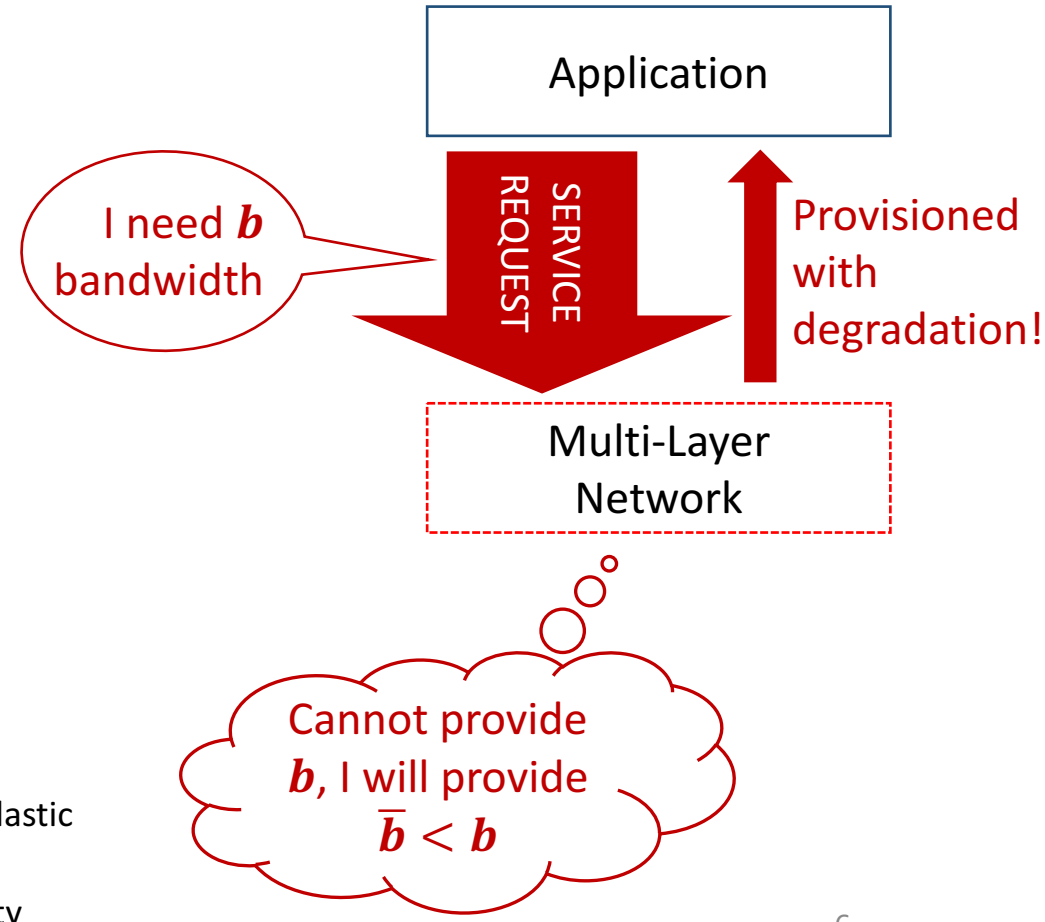
Telefonica reference network: 30 optical nodes, 56 optical links, 14 IP/MPLS nodes

Service requests: M/M/∞ queue

[1] M. Savi et al., "An Application-Aware Multi-Layer Service Provisioning Algorithm based on Auxiliary Graphs," Proc. of OFC, 2017

Unilateral Degradation of Requirements

- Instead of blocking a service request, the network may unilaterally decide to degrade the requirements
- In the state-of-the-art [2][3], only the ***bandwidth*** requirement is degraded
  - [2] proposes a priority-based service provisioning in multi-layer networks
  - [3] proposes a multi-path routing algorithm for the provisions of bandwidth-degraded reliable services
- The application is provisioned an uncontrolled degraded service
- It can be extended to multiple requirements

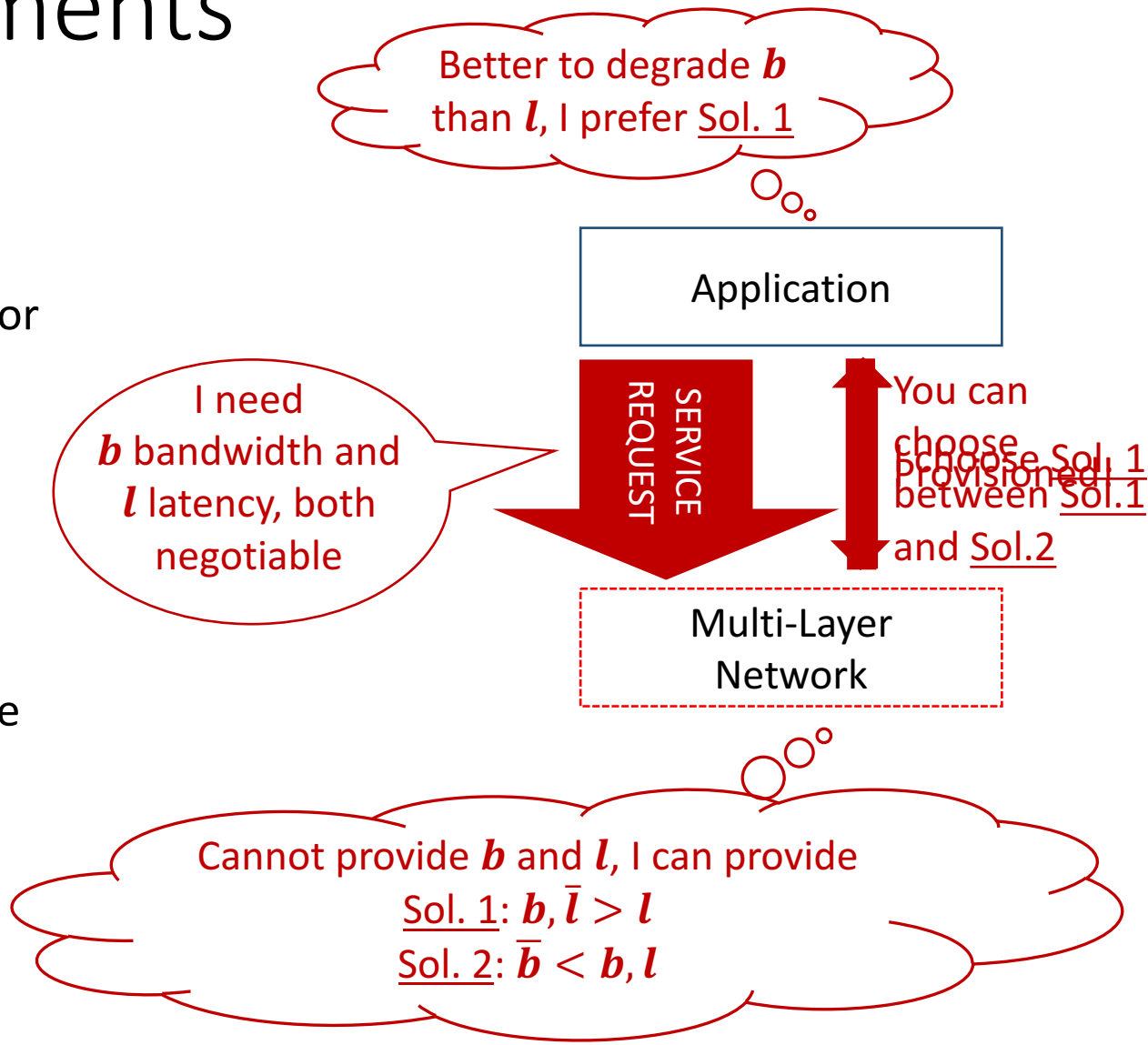


[2] Z. Zhong et al., "On QoS-Assured Degraded Provisioning in Service-Differentiated Multi-Layer Elastic Optical Networks," Proc. of IEEE GLOBECOM, 2016

[3] H. Y. Chang, ''A Multipath Routing Algorithm for Degraded-Bandwidth Services under Availability Constraint in WDM Networks,'' Proc. of IEEE WAINA 2012

Negotiation of Requirements

- If the Service Request is blocked, the application may be *flexible*
  - It can accept some degraded requirements for the Service Request
- Steps for negotiation
  1. The Application informs the network of which requirements can be negotiated
  2. The network offers several alternative solutions with degraded parameters
  3. The Application autonomously chooses the *best* one
- The negotiation offers the possibility to find an agreement between applications and networks on the provisioned service



Negotiation of Requirements

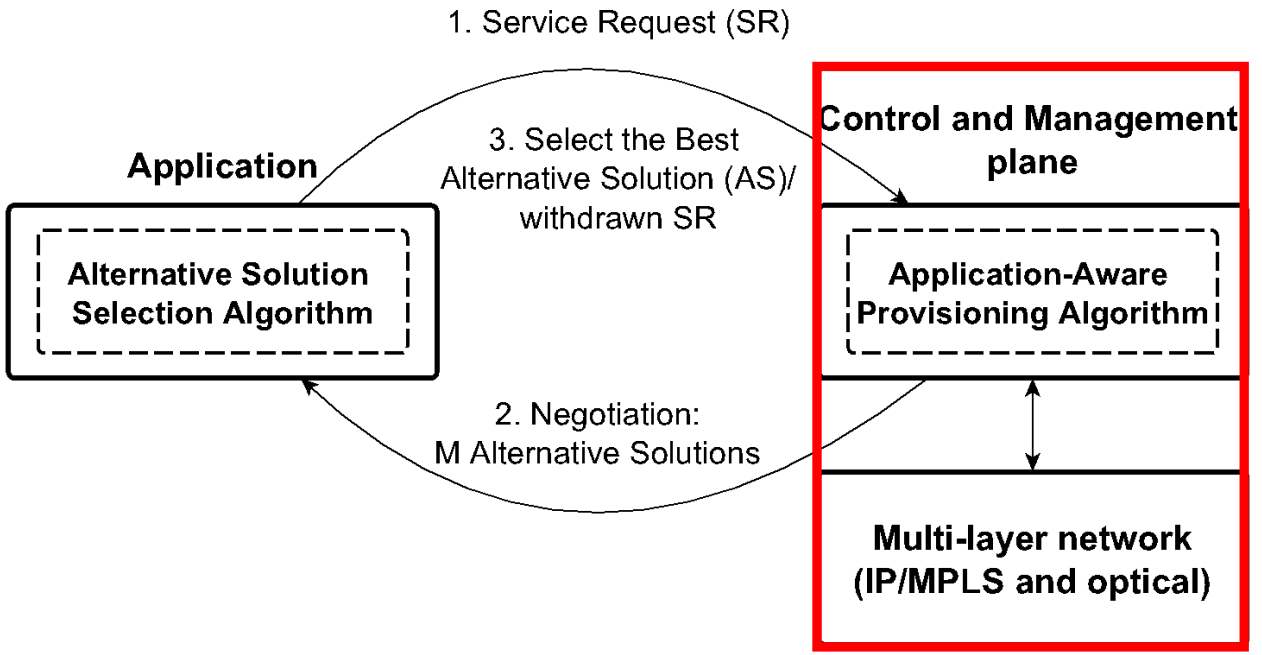
Challenges:

- *Interaction between applications and networks*
  - Definition of northbound interfaces (e.g., intent-based networking)
  - Designing of interaction mechanisms
- *Network-side algorithms*
  - Computation of the alternative solutions based on the network status
- *Application-side algorithms*
  - Applications need to perform decisions (How to chose the *best* alternative solution? What does *best* mean?)
    - *It could eventually be automated*

Negotiation of Requirements Our implementation

Network side

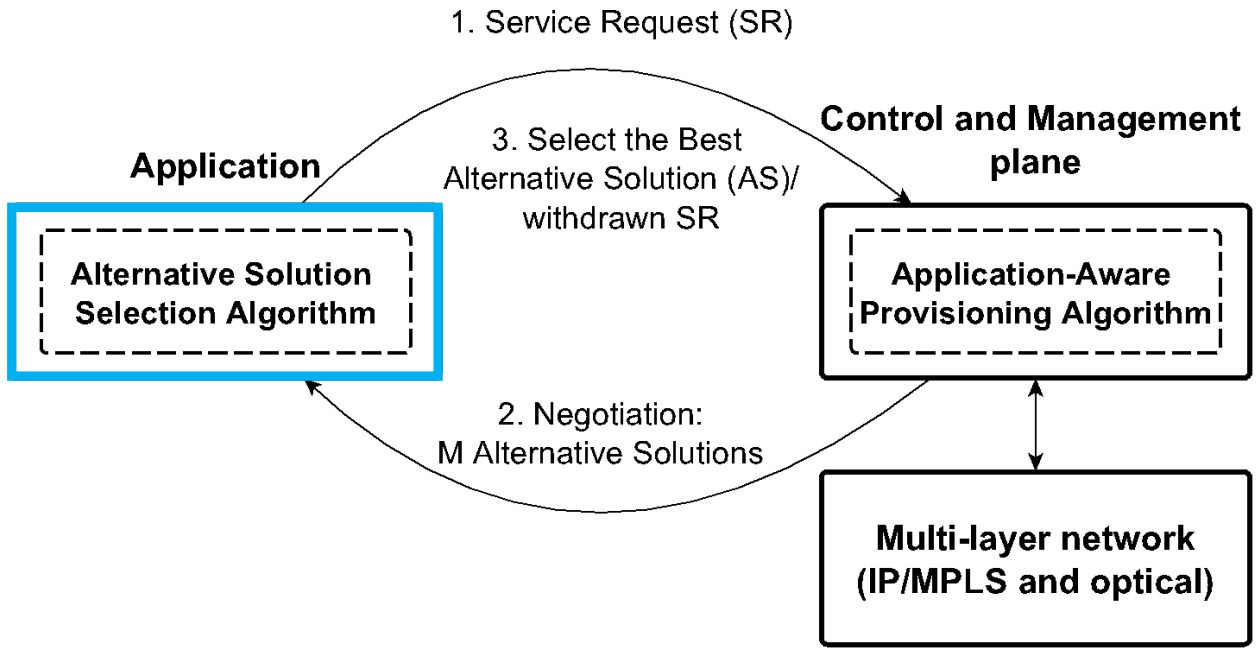
- The *Application-Aware Provisioning Algorithm* finds M alternative solutions by applying a degradation to all the combinations of negotiable requirements
- Example: bandwidth b and latency l
  - Degrade b: solution with $\bar{b} < b$ found
  - Degrade l: solution with $\bar{l} > l$ found
  - Degrade b and l: solution with $\bar{\bar{b}} < b$ and $\bar{\bar{l}} > l$ found



Negotiation of Requirements Our implementation

Application side

- The application stores all the least-acceptable values for the negotiable parameters (e.g. l_{max}, b_{min})
- *Alternative Solution Selection Algorithm*
  1. Discard all the alternative solution not meeting any of the least-acceptable values (e.g. $l > l_{max}$ and/or $b < b_{min}$)
  2. The *best* solution, among the remaining, is the one with minimum *normalized euclidean distance* from the original requirements
  3. In case there is no remaining solution, the service request is *blocked*



Preliminary Performance Evaluation Blocking Probability

1. Higher benefits (in relative terms) for **low traffic load**
2. Blocking probability can be **more than halved**, with Max Allowed Degradation=40%!

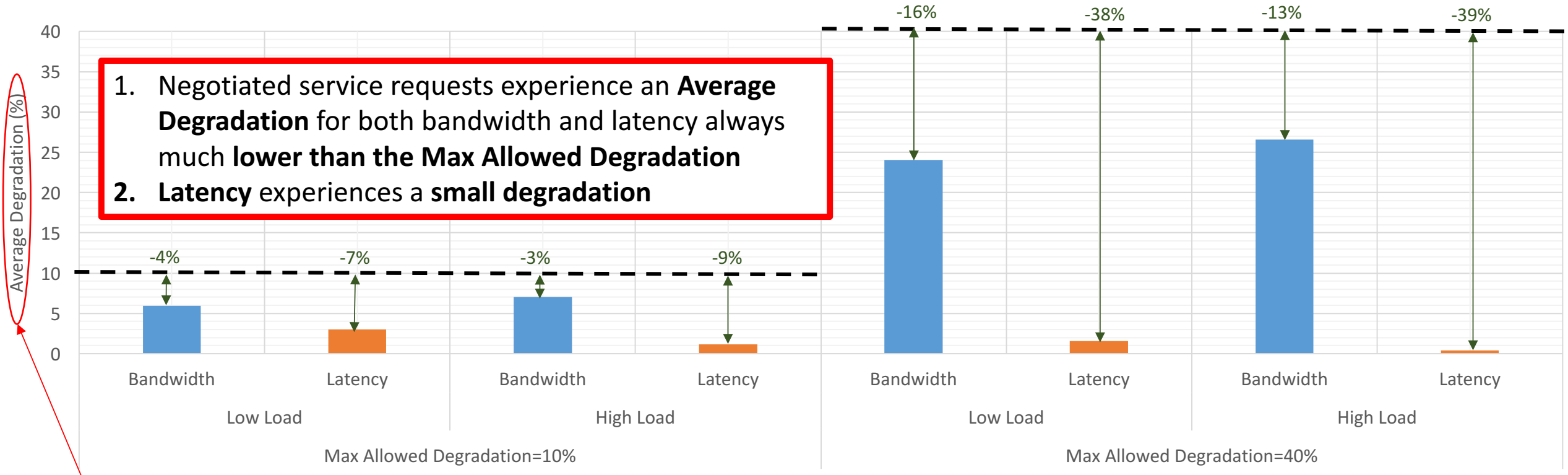
Both Bandwidth and Latency are always negotiable. Least-acceptable value for both is 10% or 40% less than the value specified in the service request.

Bandwidth = {1,2,5,10} Gbps uniform
Latency = 10 ms
Telefonica Traffic Matrix

Telefonica reference network: 30 optical nodes, 56 optical links, 14 IP/MPLS nodes

Service requests: M/M/∞ queue

Preliminary Performance Evaluation Service Degradation (Bandwidth and Latency)



Considers only the service requests that enter the negotiation phase

Bandwidth = {1,2,5,10} Gbps uniform
Latency = 10 ms
Telefonica Traffic Matrix

Telefonica reference network: 30 optical nodes, 56 optical links, 14 IP/MPLS nodes

Service requests: M/M/∞ queue

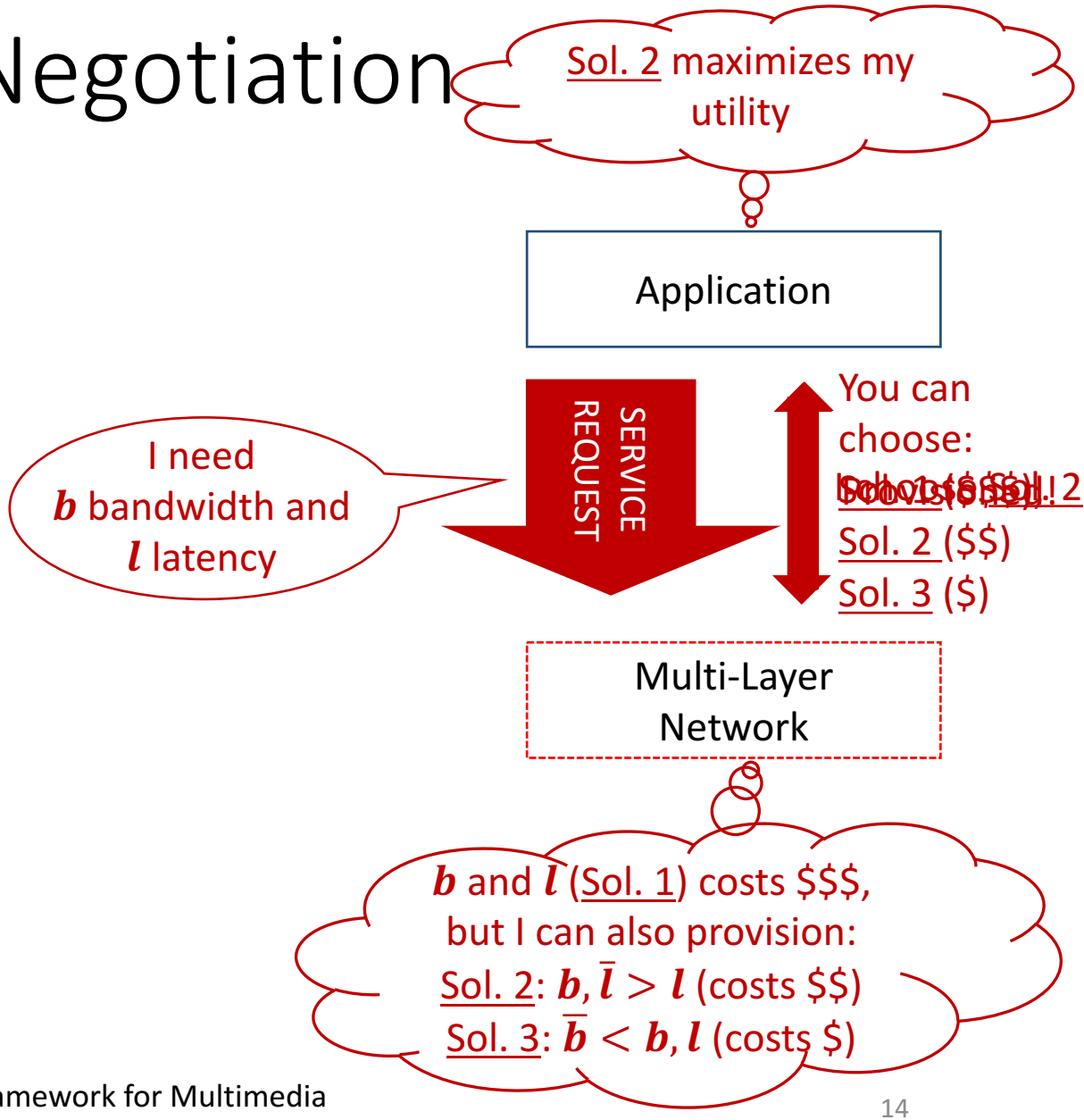
More on Negotiation

- Negotiation is not only performed to
  - Reduce blocking probability
  - Offer a controlled service degradation

  but it can be used to **satisfy both** network **and** application **utility**
- Preliminary scouting: two possible schemes
  - *Price/Service Trade-off Negotiation*
  - *Auction*

Price/Service Trade-Off Negotiation

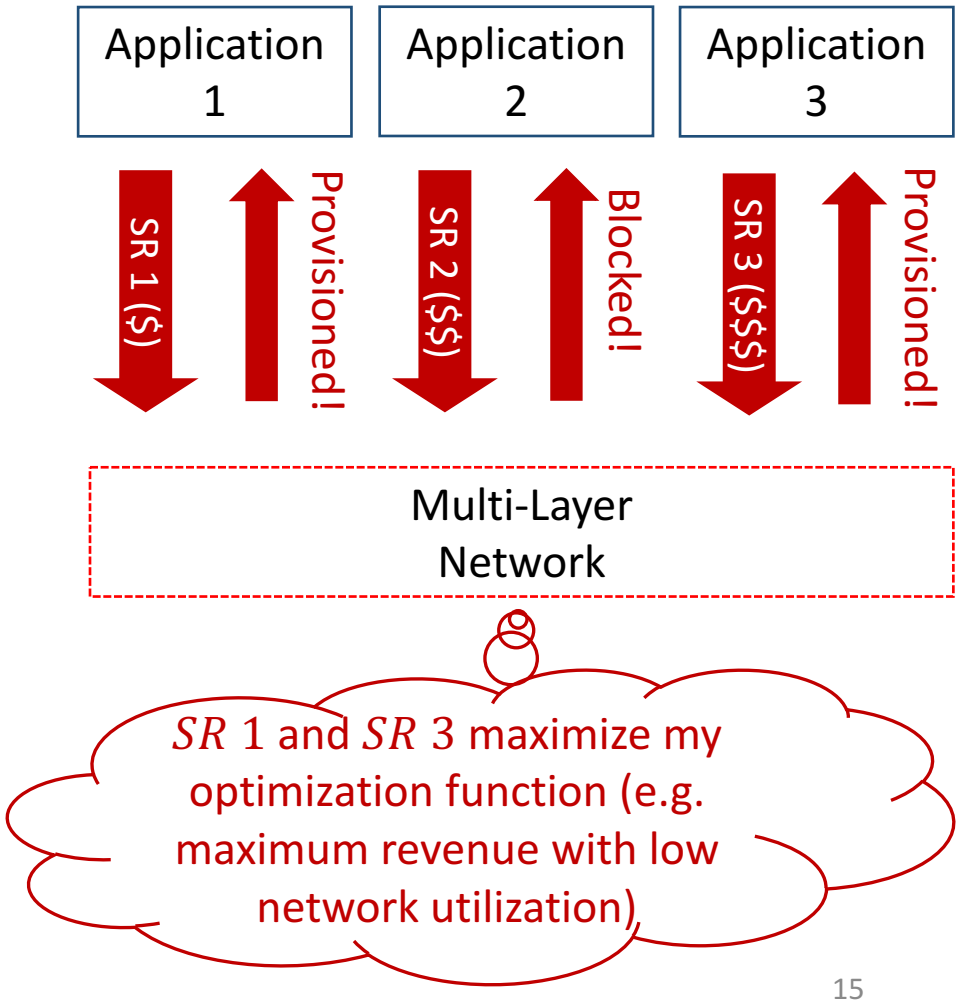
- An applications with specific requirements asks for a service request
- Networks may
  - Propose a *price* for this service request
  - Offer several alternatives with different requirements and prices
    - It can help in saving resources for other applications
- The application chooses the alternative with best price/service trade-off (utility)
- [4] proposes a pricing system for QoS negotiation in IP networks
  - It evaluates a price function by only considering the ***bandwidth*** requirement
- Could be extended to consider multiple application requirements



[4] X. Wang et al., "An Integrated Resource Negotiation, Pricing, and QoS Adaptation Framework for Multimedia Applications," IEEE Journal on Selected Areas in Communication, vol.18 n.12, pp. 2514-2529 2000

Auction

- Applications can bid a price for a service request (SR)
- The Network evaluates all the bids in batch
  - It may allocate the ones maximizing a particular optimization function
- Amazon EC2[5] and Google Cloud[6] offer this service to customers for the instantiation of Virtual Machine (VM) instances
  - VMs are instantiated with discounted prices
  - This type of instances can be disrupted at any time



[5] Amazon EC2 Spot Instances, https://aws.amazon.com/ec2/spot/
[6] Google Preemptible Virtual Machines, https://cloud.google.com/preemptible-vms/

Conclusions

- Applications come with an heterogeneous set of specific requirements to be satisfied during provisioning
- Network may not have resources to allocate some requests
- Negotiation provides an opportunity to find a compromise between apps needs and availability of resources
  - Demonstrated reduced blocking probability against a service degradation smaller than the maximum allowed one
- Negotiation may take place to maximize utility of both customers and service providers also when resources are enough

Thanks!

- Questions?

- Pls refer to: **Marco** msavi@fbk.eu or **Elio** esalvadori@fbk.eu