ONDM 2017 Tutorial

ACINO: From Application Blending to Application Centricity

Marco Savi[1], Ciril Rozic[2], Pontus Sköldström[3], Dimitrios Klonidis[2], Achim Autenrieth[4], Victor Lopez[5], Domenico Siracusa[1], Ioannis Tomkos[2]

[1]Fondazione Bruno Kessler (FBK), CREATE-NET Research Center, Italy

[2]Athens Information Technology (AIT), Greece

[3]RISE ACREO, Sweden

[4]ADVA Optical Networking, Germany

[5]Telefonica I+D, Spain

This project is funded by the European Union

Outline

1. Why application centricity
2. ACINO principles
3. Implementation: ACINO orchestrator
4. Details on multi-layer provisioning framework
5. Some simulation results
6. Movie!

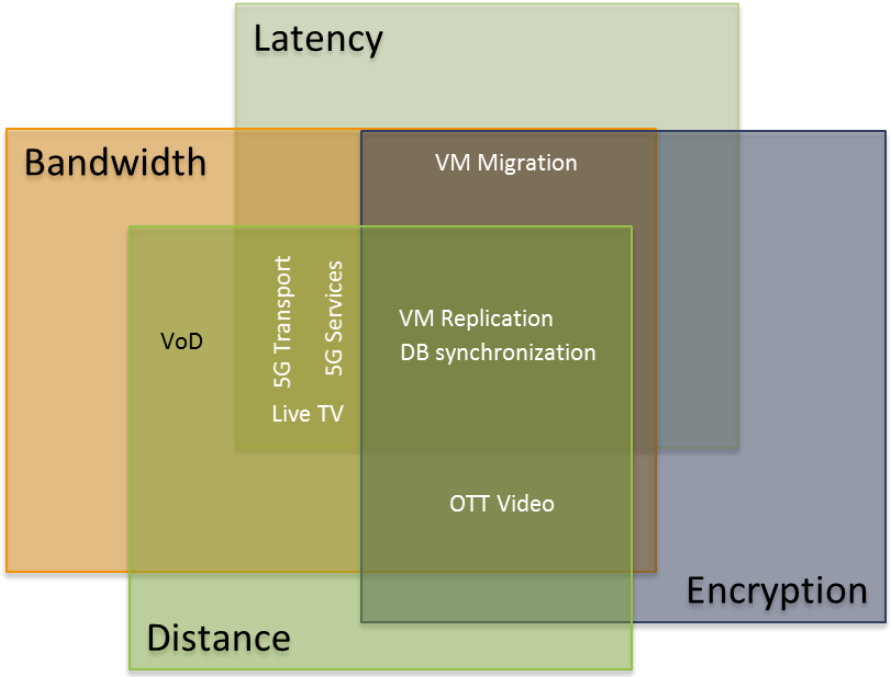
Project ACINO at a glance

- ACINO: **A**pplication-**C**entric **I**P/optical **N**etwork **O**rchestration
- H2020 RIA (Research and Innovation Action) Project
  - ICT6 - Smart optical and wireless network technologies
- Project duration: 36 months (01/02/2015 - 31/01/2018)
- Partners: 6
- Budget: 2.88 M€
- Effort: 290 Person Months

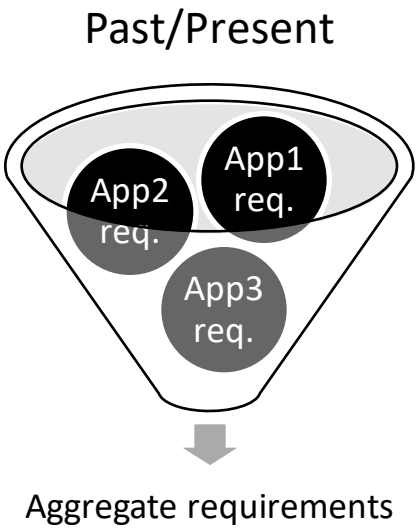
What applications want

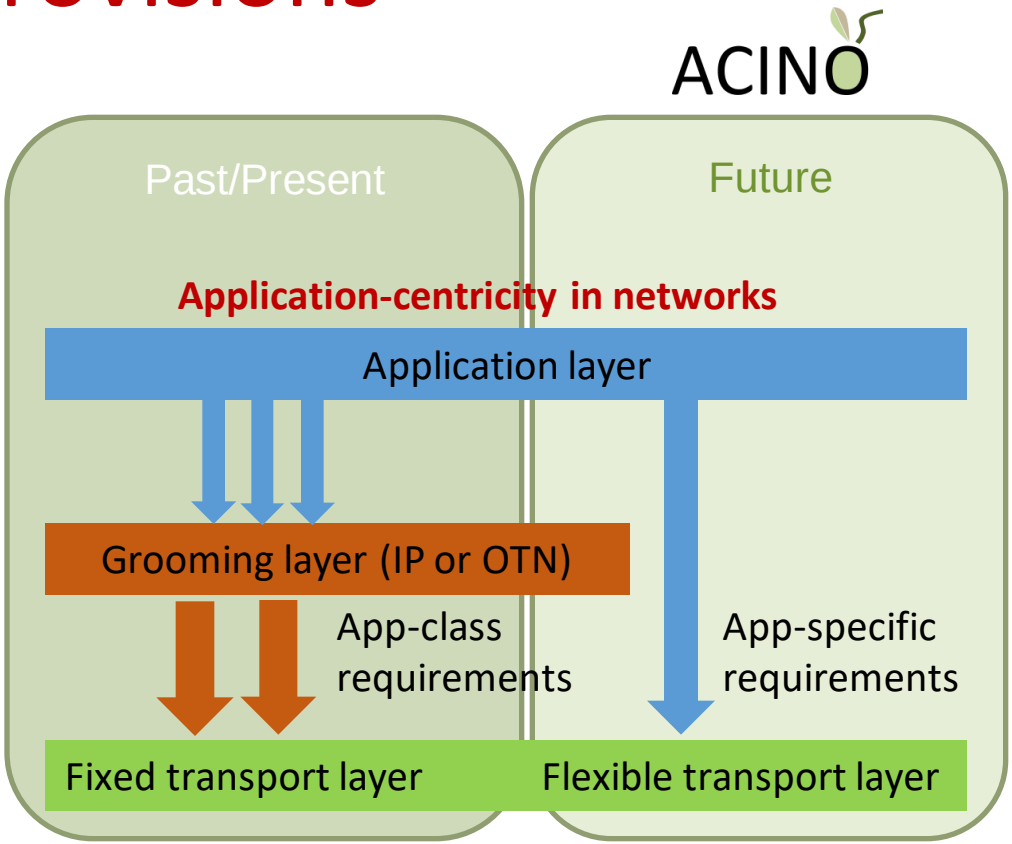
Past	Present/Future
Mostly just bandwidth	Diverse set of requirements

- **Simply provisioning enough bandwidth to applications is not enough anymore**
- Applications are diversifying
  - Need to proper classify them
- Even if applications are classified
  - Many possible requirements may result in a lot of application classes



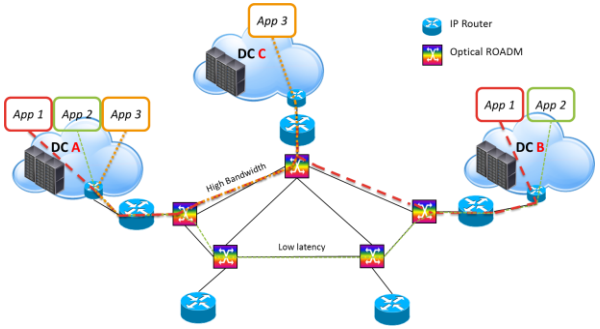
How network provisions



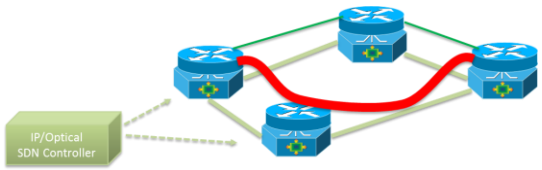


- Differentiate application flows down to the optical layer
  - Multi-layer approach
  - Some aggregation can still be done, but network can now cater to the specific applications' needs

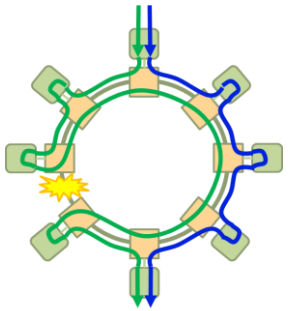
Some ACINO Use Cases



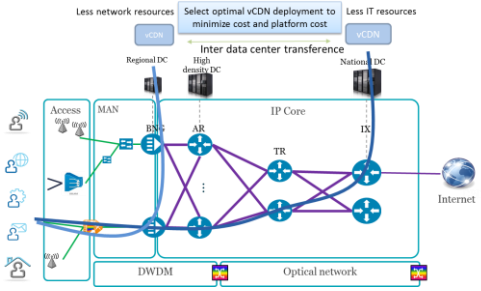
Application-based Data Center Interconnection



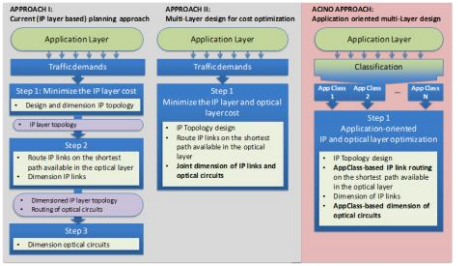
Secure Transmission as a Service



Application-specific resiliency strategies

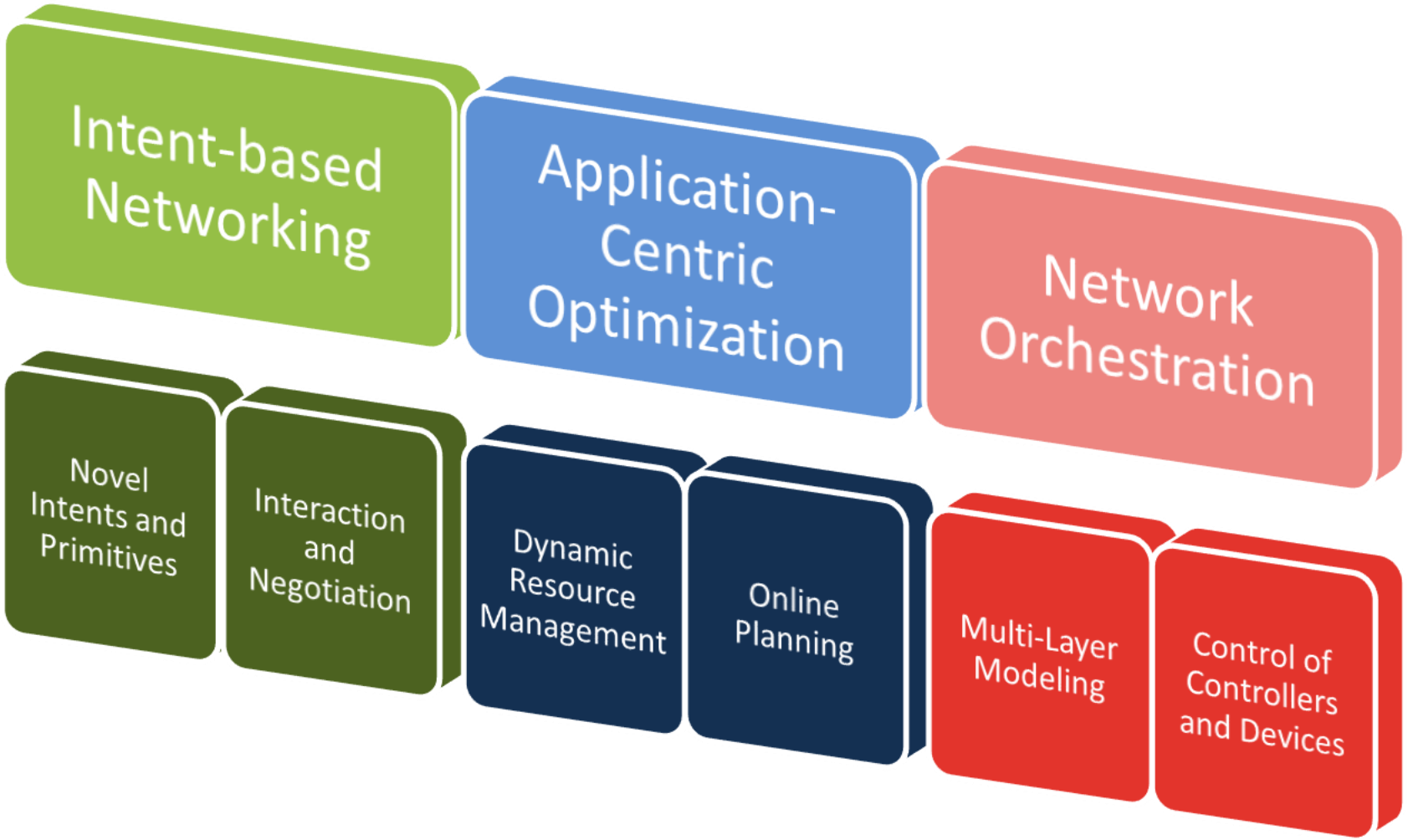


Dynamic Virtual CDN deployment



Application-centric in-operation net planning

ACINO building blocks



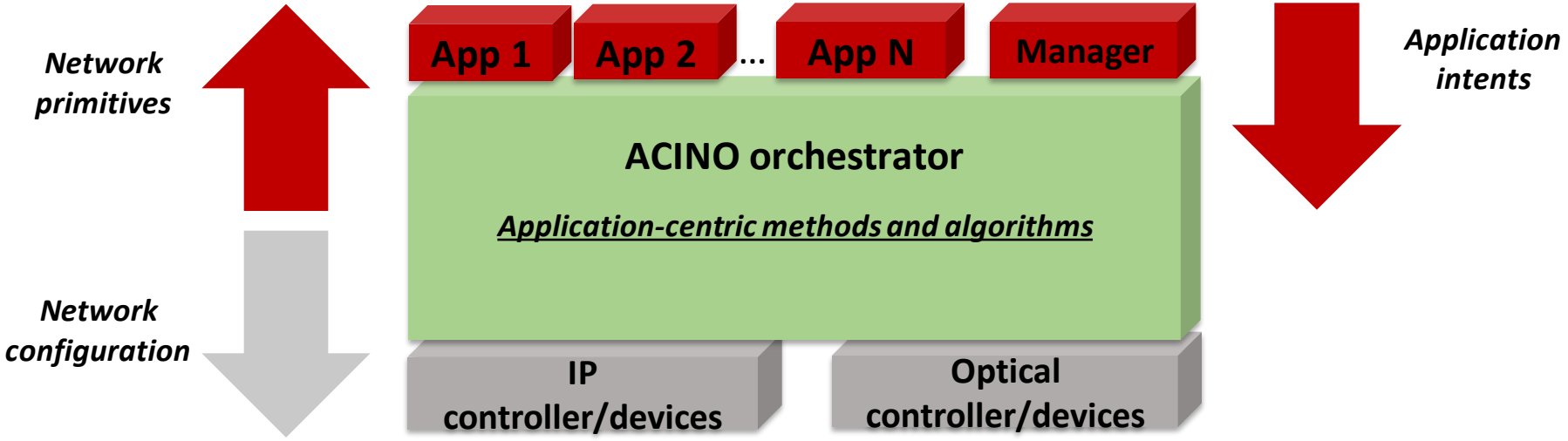
Implementation of ACINO principles: the ACINO orchestrator

- ACINO needs an entity that
  - Has a complete view of the network
    - optical nodes
    - optical links
    - IP nodes
    - IP links
    - running services
    - …
  - Must be able to add and remove connections in either layer
  - Must be able to communicate with network applications

SOFTWARE-DEFINED NETWORKING (SDN) APPROACH!

SDN Orchestration

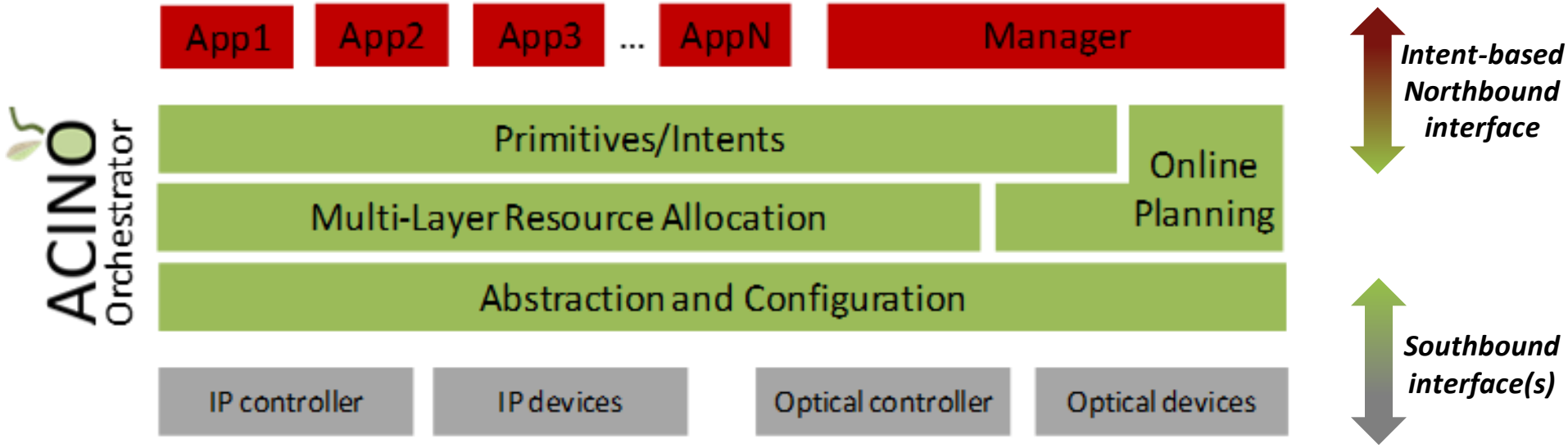
SDN network orchestrator enabling applications to program the IP/Optical transport network



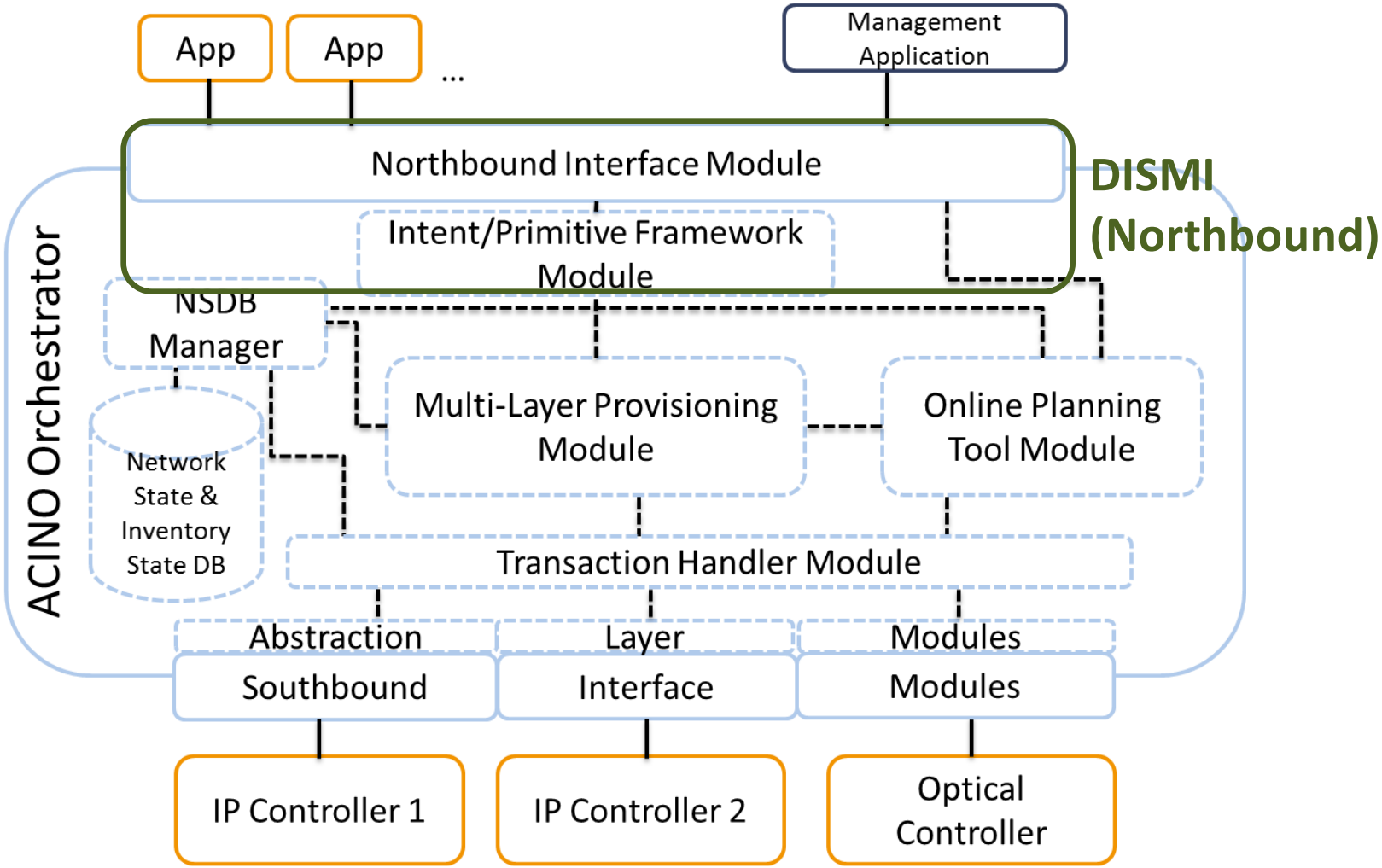
ACINO orchestrator implementation: resource allocation and optimization

ACINO orchestrator high-level architecture

SDN network orchestrator enabling applications to program the IP/Optical transport network

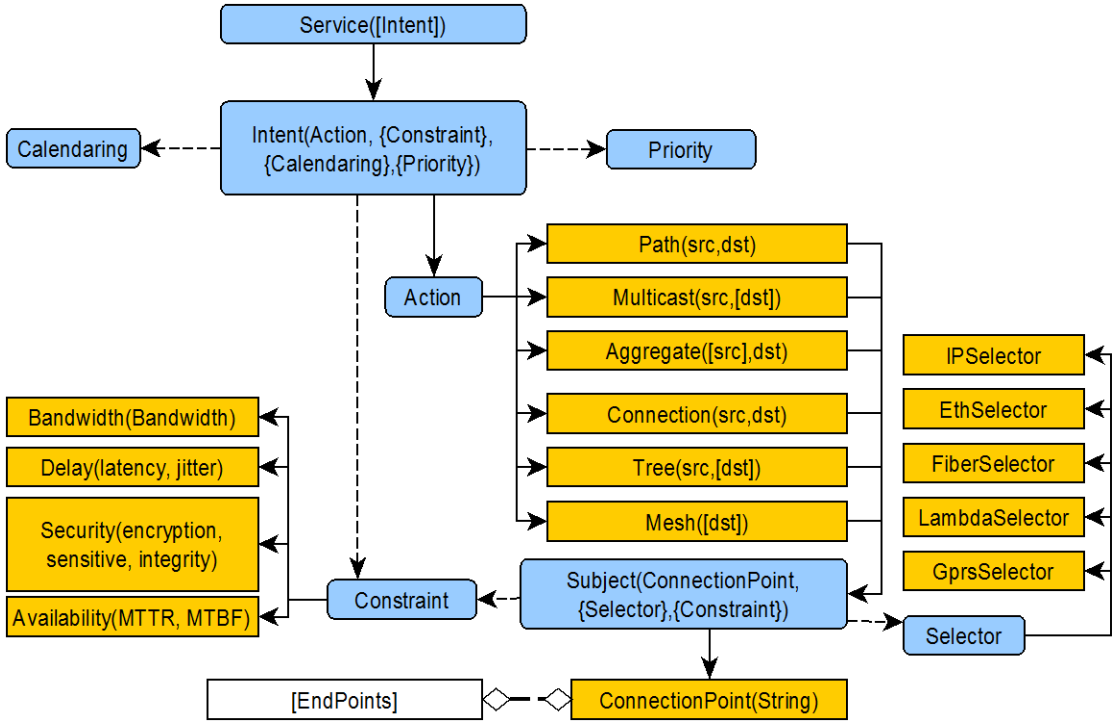


ACINO orchestrator details

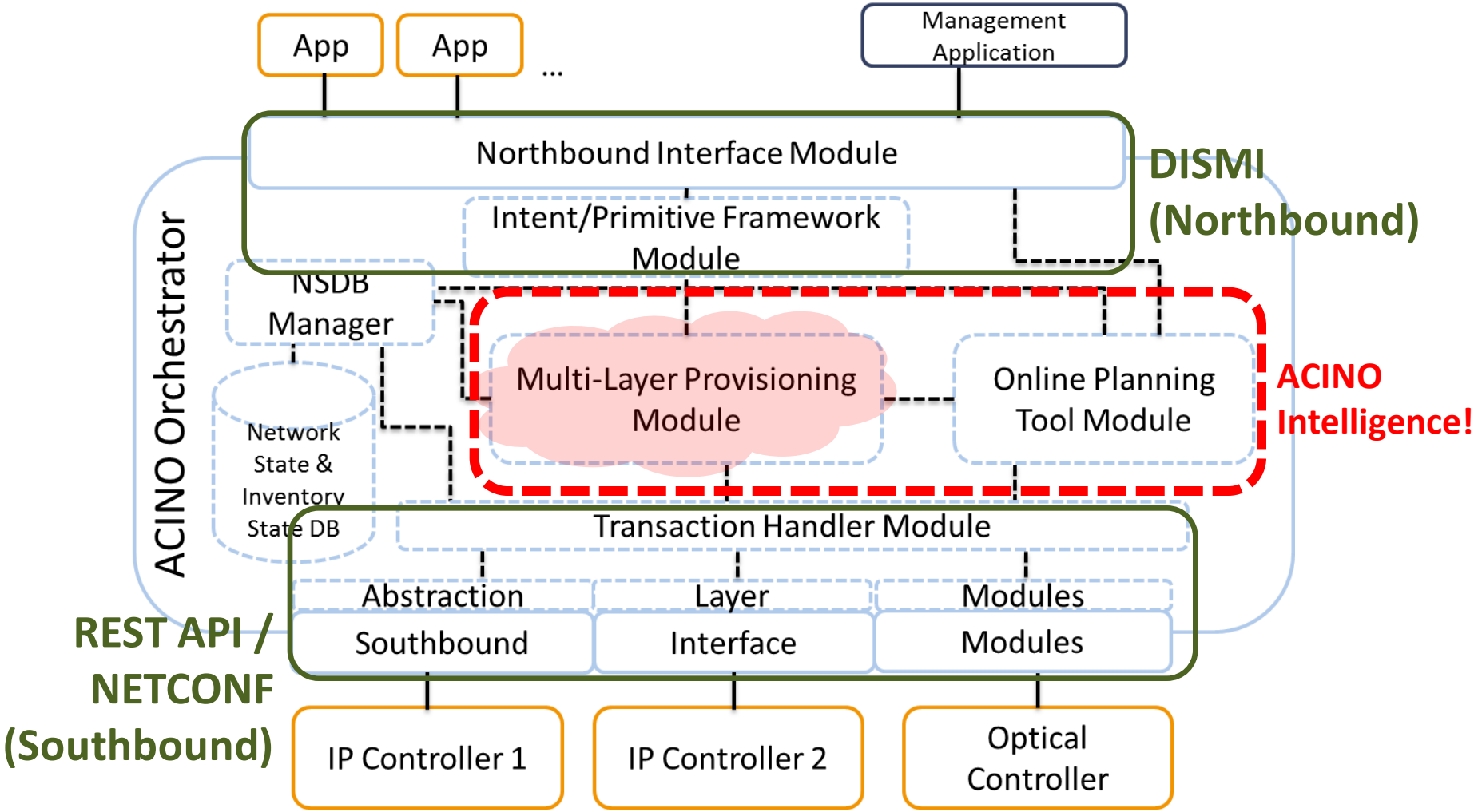


DISMI primitives and grammar

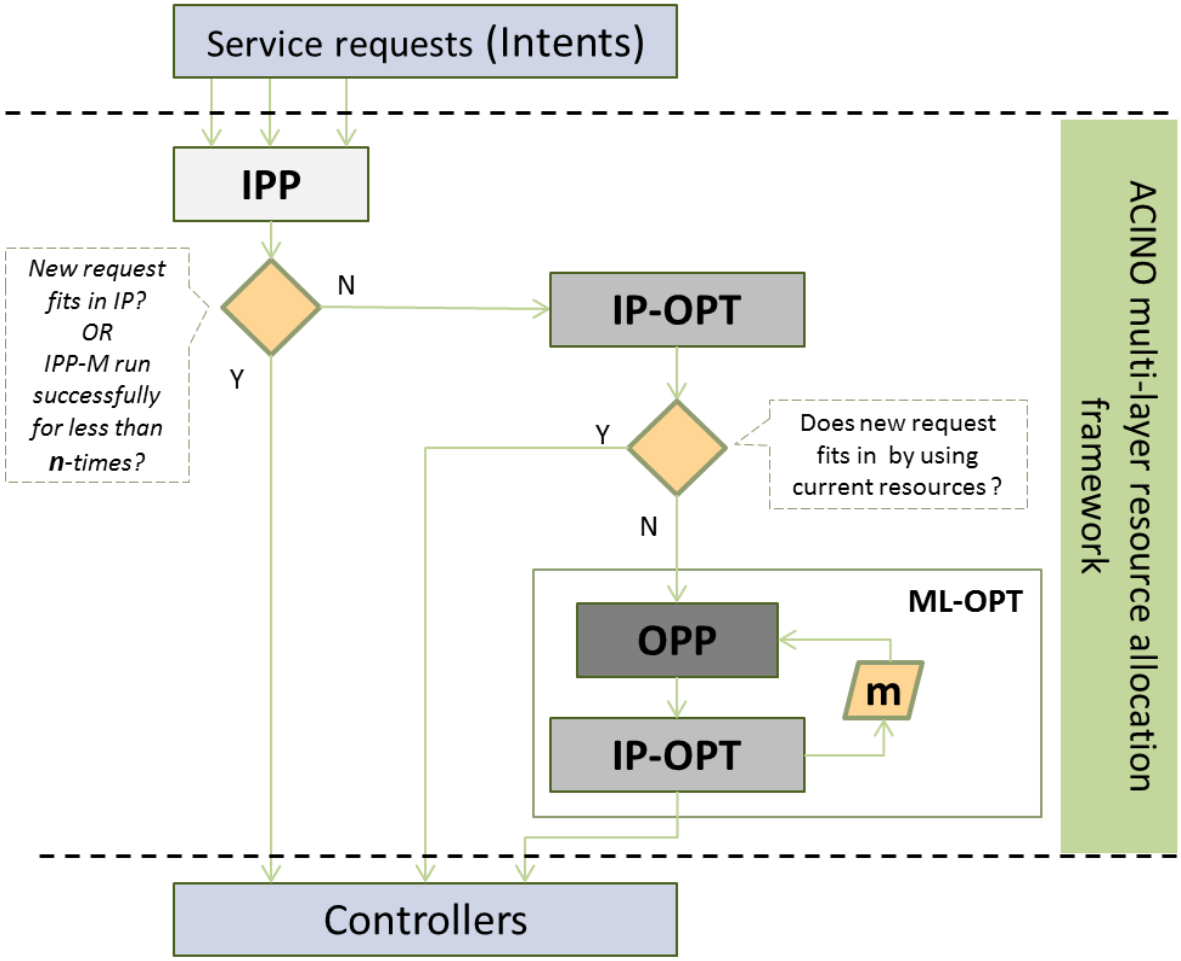
- **Primitives**
  - *Actions* describe the connection requests
  - *Constraints* characterize the requested connection
  - *Selectors* filter the traffic entering the network, enabling the creation of application classes
- How the primitives interact forms the **grammar** of the interface



ACINO orchestrator details



Multi-layer provisioning framework



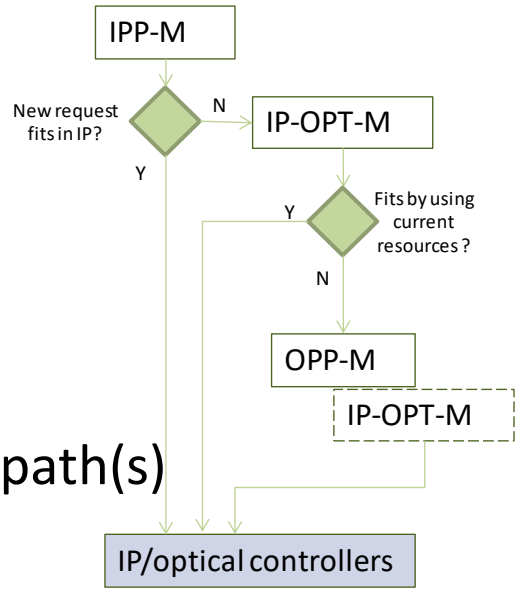
IPP	provisions in IP layer
IP-OPT	optimizes IP layer
OPP	optimizes optical layer

Rationale:

- Optical layer is generally slower than IP
- In a dynamic situation, we want to re-use network resources to save energy and minimize setup times

Module definition

- IPP (IP Provisioning Module)
  - Tries to accommodate the service request at the IP layer
  - Finds an application-aware path
    - Such path meets all the application requirements specified by the service request (e.g. bandwidth, latency, availability, protection…)
    - No new lightpath is established in the network
- IP-OPT (IP Optimizer)
  - Rearranges IP connections in a *hitless* and optimized way
- OPP (Optical Provisioning Module)
  - Similar to IPP, but adds new IP links (lightpaths)
  - Finds the optimal lightpath(s) to add
  - Finds an application-aware path using such lightpath(s)

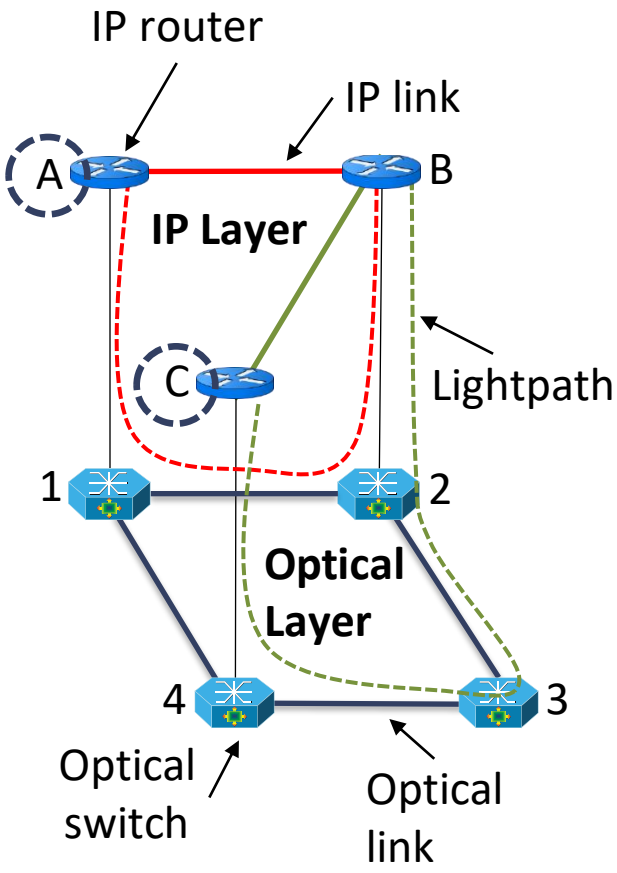


Service request definition

- Tuple $SR(s, d, b, l, a, p)$
  - s source IP router
  - d destination IP router
  - b requested bandwidth
  - l maximum path latency
  - a minimum path availability
  - p protection true/false

Application requirements

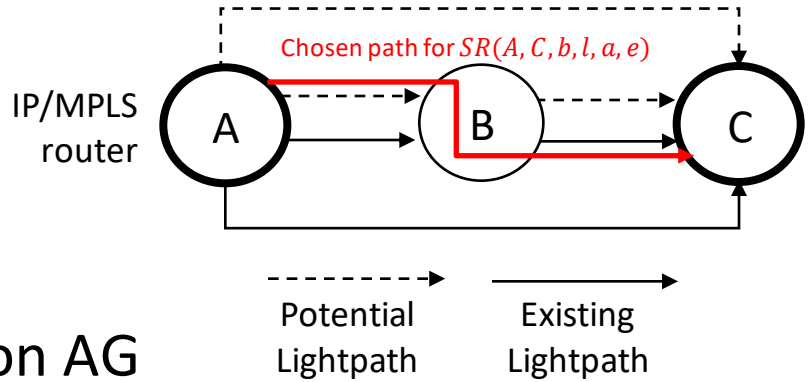
- Example: SR(A, C, 10Gbps, 50ms, 99.999%, true)



Auxiliary graph model

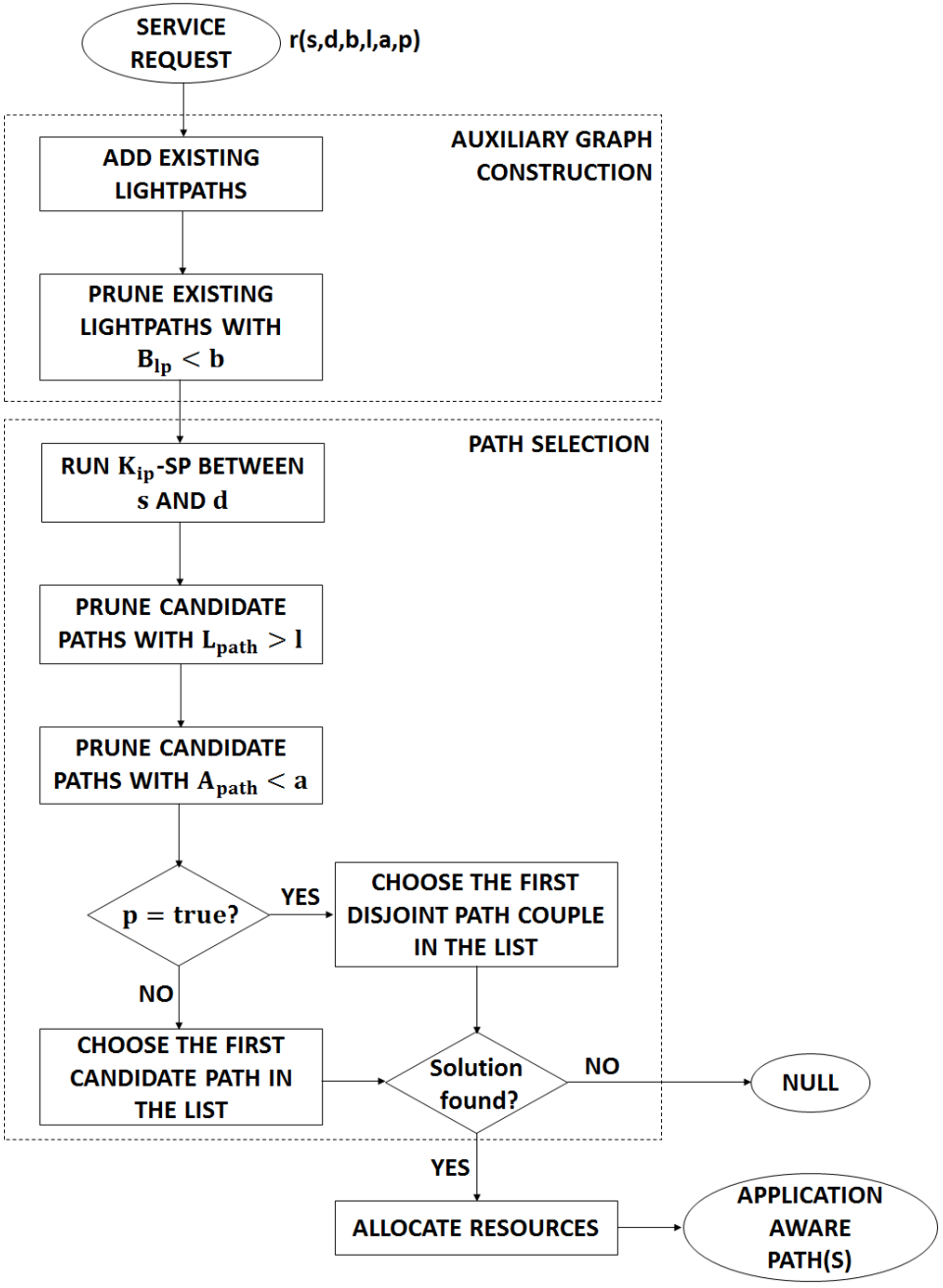
- IPP and OPP are *auxiliary-graph-based* modules

- Auxiliary Graph (AG)
  - Single layer graph
  - Constructed everytime IPP or OPP is called
  - «Augmented» IP network graph
- AG nodes
  - IP routers
- AG links
  - Existing lightpaths
  - Potential lightpaths
- Application-aware paths searched on AG



IPP details

- **Auxilary Graph construction**
  - Add all existing lightpaths
  - Prune those not meeting b
- **Path selection**
  - Find the K shortest paths on AG between s and d (*candidate paths*)
    - Weight on AG links: different possible polices
    - We choose physical length of the lightpaths (distance)
  - Prune those not meeting l and a
  - Select
    - The first path in the list if p=*false*
    - The first disjoint path couple in the list if p=*true*
- **Resource allocation**
  - Allocate resources on the chosen existing lightpaths (i.e., at IP layer)



IP-OPT details

- Finds the best list of IP connection *moves*
- Running services must not be affected
- Implementation of moves
  - Hitless (make-before-break)
  - List can be abandoned at any point, e.g. in case of a failure

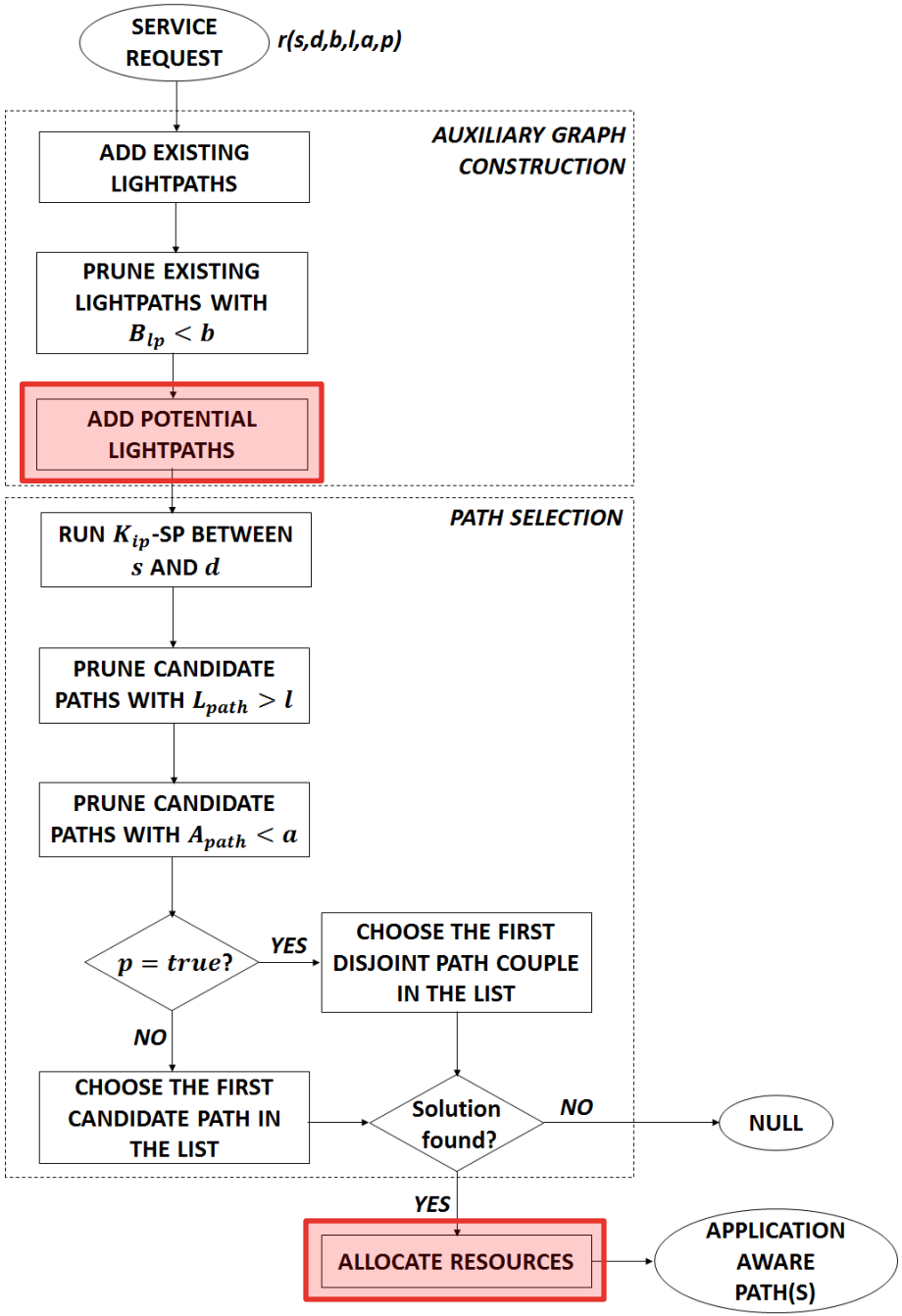
IP-OPT details

- Search for the moves
  1. Sort the connections (initial or random or by BW size)
  2. Each connection
     1. Move to one of K shortest IP paths
     2. Choose the move with the best “cost”
- “Cost” minimization objectives in order

priority	quantity	motivation
1.	# of blocked requests	accommodate applications
2.	# of IP interfaces	save energy and cost
3.	IP link load balancing	prepare for future requests

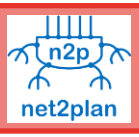
OPP details

- **Auxiliary graph construction**
  - Similar to IPP, but augments the auxiliary graph with *potential lightpaths*
    - Lightpaths that can be established if needed
    - *k-SP FirstFit* Routing and Spectrum Assignment at optical layer to find potential lightpath
- **Path selection**
  - Same as IPP
- **Resource allocation**
  - Allocate resources on the chosen existing lightpaths (IP layer)
  - Establish the chosen potential lightpaths (optical layer)
    - Allocate resources on the newly-established lightpaths (IP layer)



Simulation tool: many requirements

- An engine to accommodate incoming service requests
  - Dynamically
  - Multi-layer (IP/optical)
  - Simulation code must be used later to run on a testbed
  - Open source
  - Able to incorporate ACINO code

NET2PLAN! 

P. Pavon-Marino *et al.*, "Net2Plan: an open source network planning tool for bridging the gap between academia and industry," IEEE Network, 2015

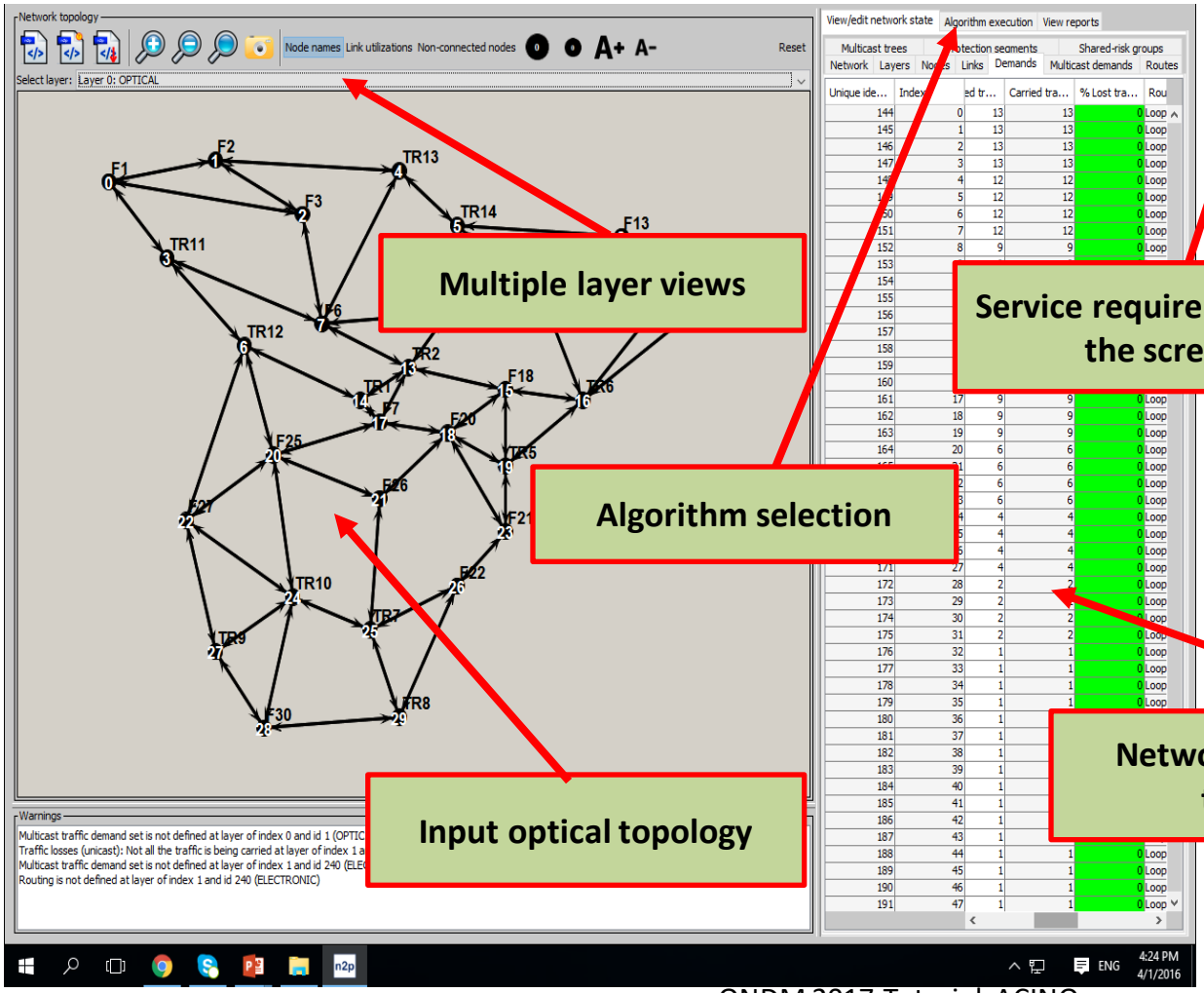
Net2Plan in a nutshell

- Java-based network planning tool
- Can easily manage multi-layer networks
- Provides a very nice GUI! ☺
- Two tools
  - Offline network design (planning)
  - Online simulation (discrete event simulator)
- Implementation of the framework using the **Online Simulation Tool**
- Simulation of a dynamic scenario with service requests arriving/leaving over time
  - Two Java classes needed
    - Event generator
    - Event processor

Net2Plan as an ACINO simulation platform

Inputs: fiber & IP topology, traffic demands, service requirements…

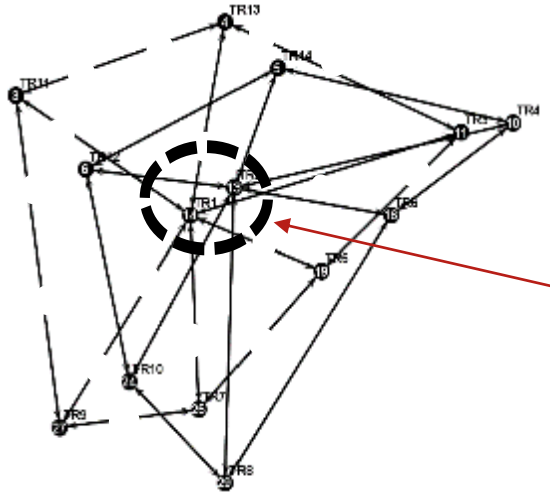
Outputs: lightpath & IP/MPLS routes, link capacities, blocked demands…



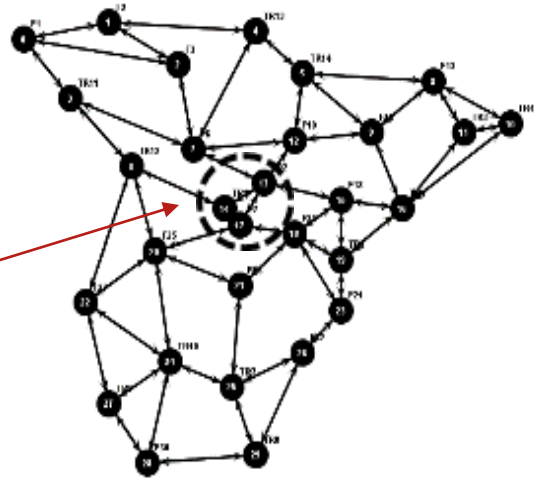
ACINO

Simulation Results

Scenario setup: Reference network



Madrid



example IP topology

fiber topology

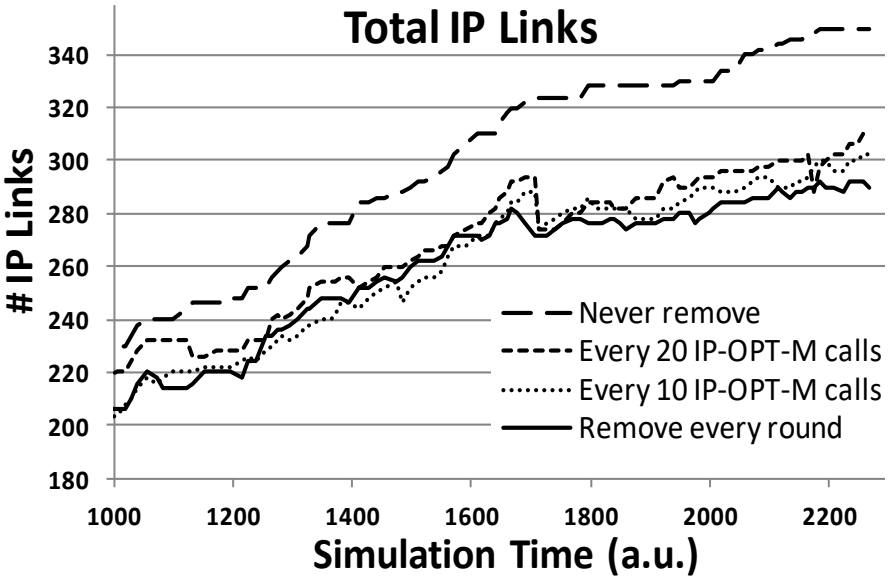
- Fiber lengths are known
  - Propagation delay = latency
- We use only one fiber per fiber-link
- Transceiver bitrates: 10 Gb/s and 100 Gb/s

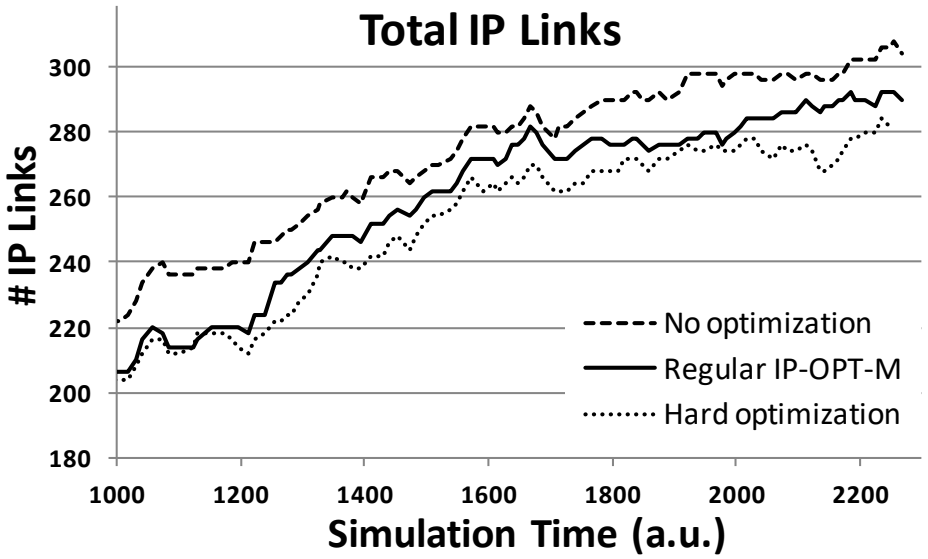
Scenario setup: Input traffic (1)

Traffic class	Percentage	Service requirements	Resource allocation policy
High priority	10%	Max latency: 4.5ms BW: 1-10 Gb/s	Route separately in optical
Latency-sensitive	15%	Max latency: 4.5ms BW: 10-100 Gb/s	Aggregate in IP
Best effort	75%	BW: 10-100 Gb/s	
	100%		

- End-to-end traffic matrix known
- Poisson arrival/departure with average rates from the traffic matrix

IP-OPT evaluated (1)

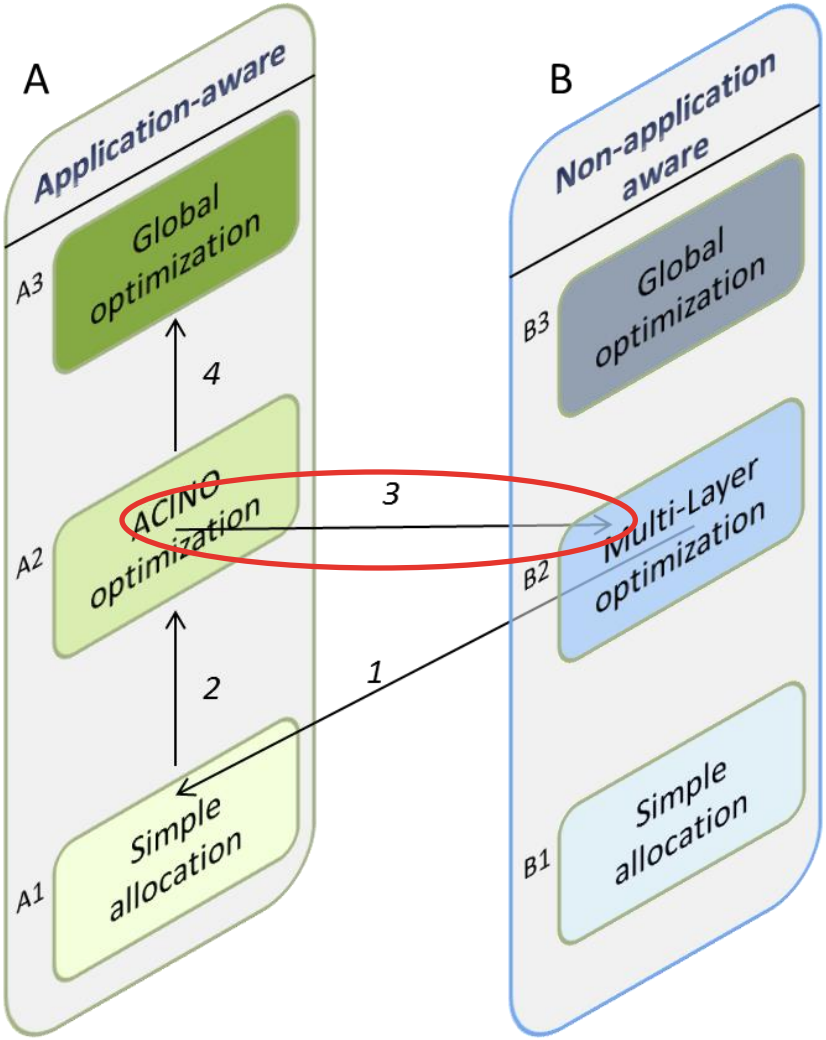




Infrequent removal is good enough

Hitless IP optimization is between none and hard/hitful

ACINO vs. Benchmark (2)



- **Application-unaware**
  - Considers only the *bandwidth* requirement in the provisioning of service requests
- **Application-aware**
  - Considers all the application requirements (*bandwidth*, *latency*, *availability*, *protection*)
- ACINO optimization and multi-layer application-unaware optimization comparison
  - Benefits of ACINO in terms of *application-awareness*

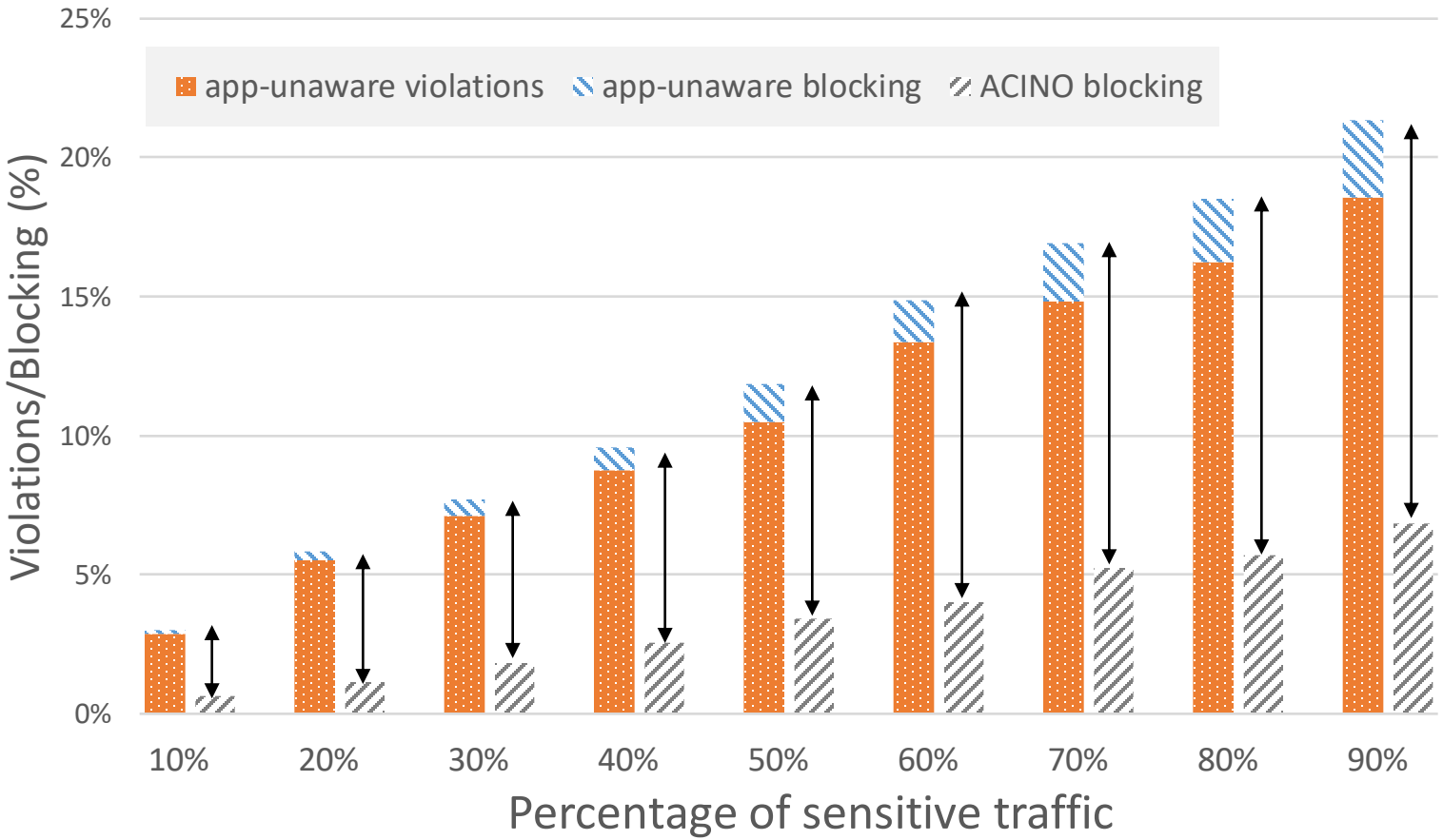
Scenario setup: Input traffic (2)

Traffic class	Percentage	Service requirements	Resource allocation policy
Latency-sensitive	Varies	Max latency: 6ms BW: 10-100 Gb/s	Aggregate in IP
Latency-sensitive + Availability + Protection	Varies	Max latency: 6ms Availability: 99.5% BW: 10-100 Gb/s Protection at IP	
Best effort	Varies	BW: 10-100 Gb/s	
	100%		

Evaluation metrics (2)

- **Service Request Violation**
  - The ratio of the connections that do not meet any of their service requirements, to the total number of requested connections
- **Connection Establishment Blocking**
  - The ratio of the connections blocked due to lack of resources able to meet the service requirements, to the total number of requested connections

ACINO vs. Benchmark (2)



Blocking of ACINO always much lower than (Blocking+Violations) of Application-Unaware

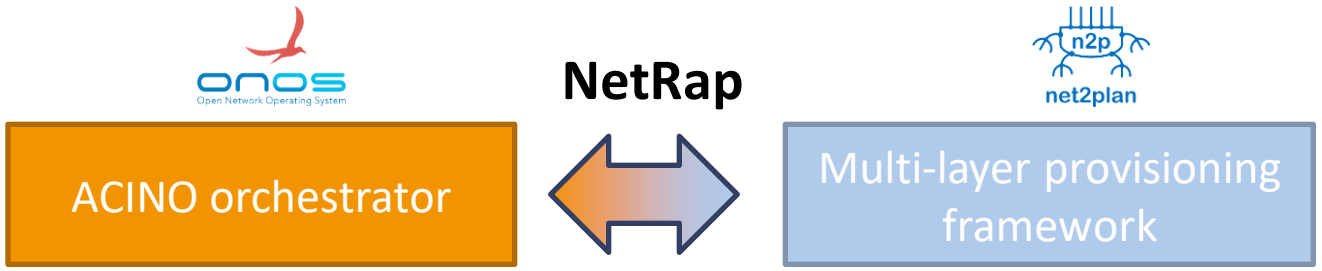
Let's summarize

- In ACINO we deal with
  - Dynamic/online allocation/optimization
  - Multi-layer
  - Application-centricity/awareness
- Introduced an allocation/optimization framework
  - Application-centric in IP ***and*** optical multi-layer networks
  - Reuses efficiently resources
- IP optimization can save resources
- ACINO blocks less than the benchmark blocks and violates

Three more slides, then the movie!

How the framework works on a testbed

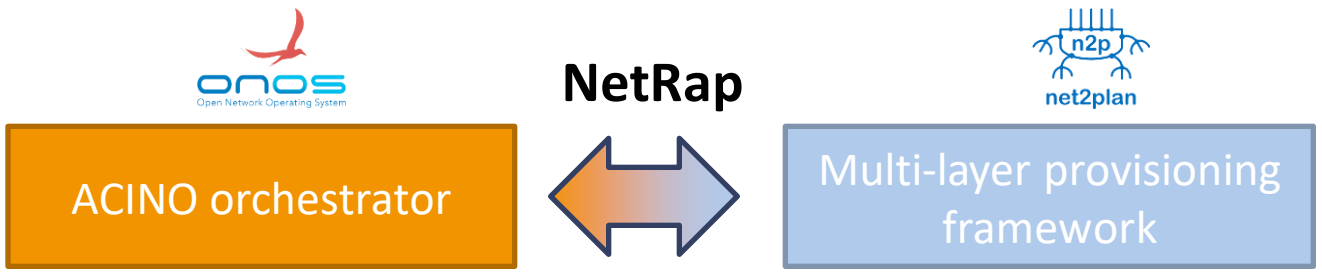
- The **ACINO orchestrator** has been implemented in ONOS http://onosproject.org/
- The **multi-layer provisioning framework** has been implemented in Net2Plan
- Need to exchange information between ONOS and Net2Plan
  - Design of an interface called **NetRap**



How the framework works on a testbed

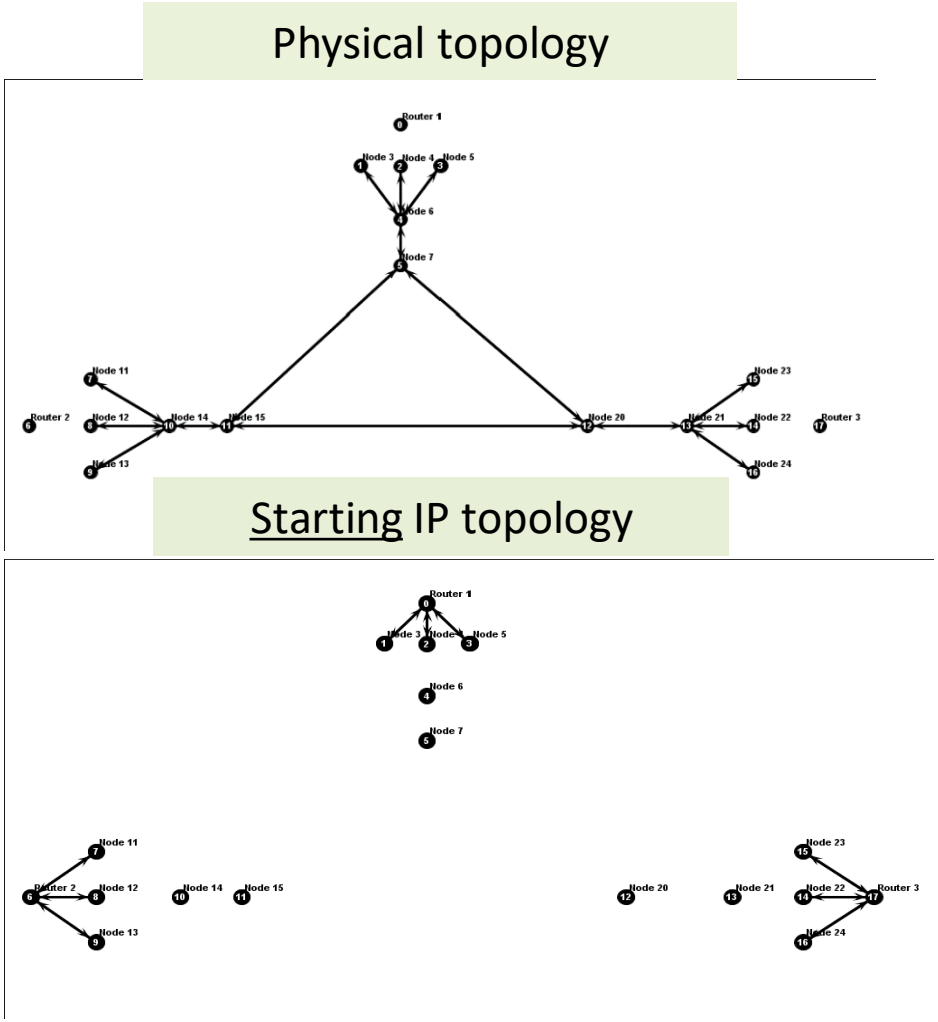
ONOS-Net2Plan interaction

1. Net2Plan receives from ONOS: network topology and other network information
2. Net2Plan computes a path (or says “I can’t”)
3. Net2Plan sends the path information back to ONOS
4. ONOS requests a path from IP and optical controllers



ACINO on a testbed

Three IP+optical nodes



How it all comes together: video demo

- https://youtu.be/5_u_2gULNW0
- By Pontus Sköldström, ACREO, Sweden
- 10 minutes

For more info, see publications

- Ć. Rožić, C. Matrakidis, D. Klonidis, I. Tomkos, ***Network Primitives Based on Latency and Recovery Time in Orchestrated Multi-layer Networks***, International Conference on Transparent Optical Networks (ICTON) 2017
- P. Sköldström, Ć. Rožić, J. J. Pedreño Manresa, ***Making Powerful Friends: Introducing ONOS and Net2Plan to Each Other***, International Conference on Transparent Optical Networks (ICTON) 2017
- M. Savi, C. Rozic, C. Matrakidis, D. Klonidis, D. Siracusa, I. Tomkos, ***Benefits of Multi-Layer Application-Aware Resource Allocation and Optimization***, European Conference on Networks and Communications (EuCNC) 2017
- V. Lopez, D. Konidis, D. Siracusa, C. Rozic, I. Tomkos, J. P. Fernandez-Palacios, ***On the benefits of multi-layer optimization and application awareness*** *(invited)*, Journal of Lightwave Technology 2017
- Ć. Rožić, M. Savi, C. Matrakidis, D. Siracusa, D. Klonidis, I. Tomkos, ***A Framework for Dynamic Multi-layer Resource Allocation and Optimization in Application-Centric Networking***, Optical Networking and Communication Conference (OFC) 2017
- M. Savi, F. Pederzolli, D. Siracusa, ***An Application-Aware Multi-Layer Service Provisioning Algorithm based on Auxiliary Graphs***, Optical Networking and Communication Conference (OFC) 2017
- F. Pederzolli, D. Siracusa, P. Sköldström, S. Junique, Ć. Rožić, D. Klonidis, T. Szyrkowiec, M. Chamania, V. Uceda, V. Lopez, Y. Shikhmanter, O. Gerstel, ***SDN application-centric orchestration for multi-layer transport networks*** *(invited)*, International Conference on Transparent Optical Networks (ICTON) 2016
- Ć. Rožić, D. Klonidis, I. Tomkos, ***Latency-aware Multi-layer Network Optimization in IP-over-WDM Core Networks***, 42nd European Conference on Optical Communication (ECOC) 2016
- Ć. Rožić, D. Klonidis, I. Tomkos, ***A Survey of Multi-layer Network Optimization*** *(invited)*, 20th International Conference on Optical Network Design and Modeling (ONDM) 2016

Thank you for your attention

And thanks to all partners in EU H2020 project ACINO

www.acino.eu

ACINO
Application-Centric IP/optical
Network Orchestration

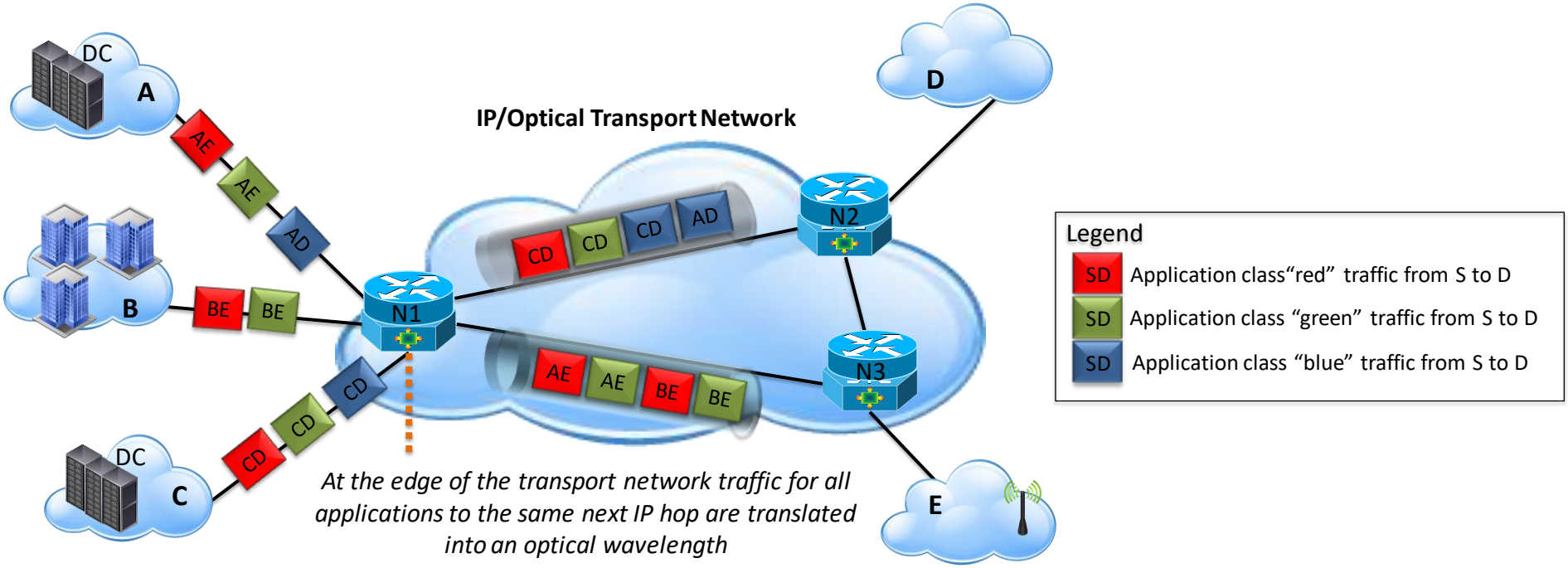
Backup slides

This project is funded
by the European Union

Approach:
From application-unaware baseline…

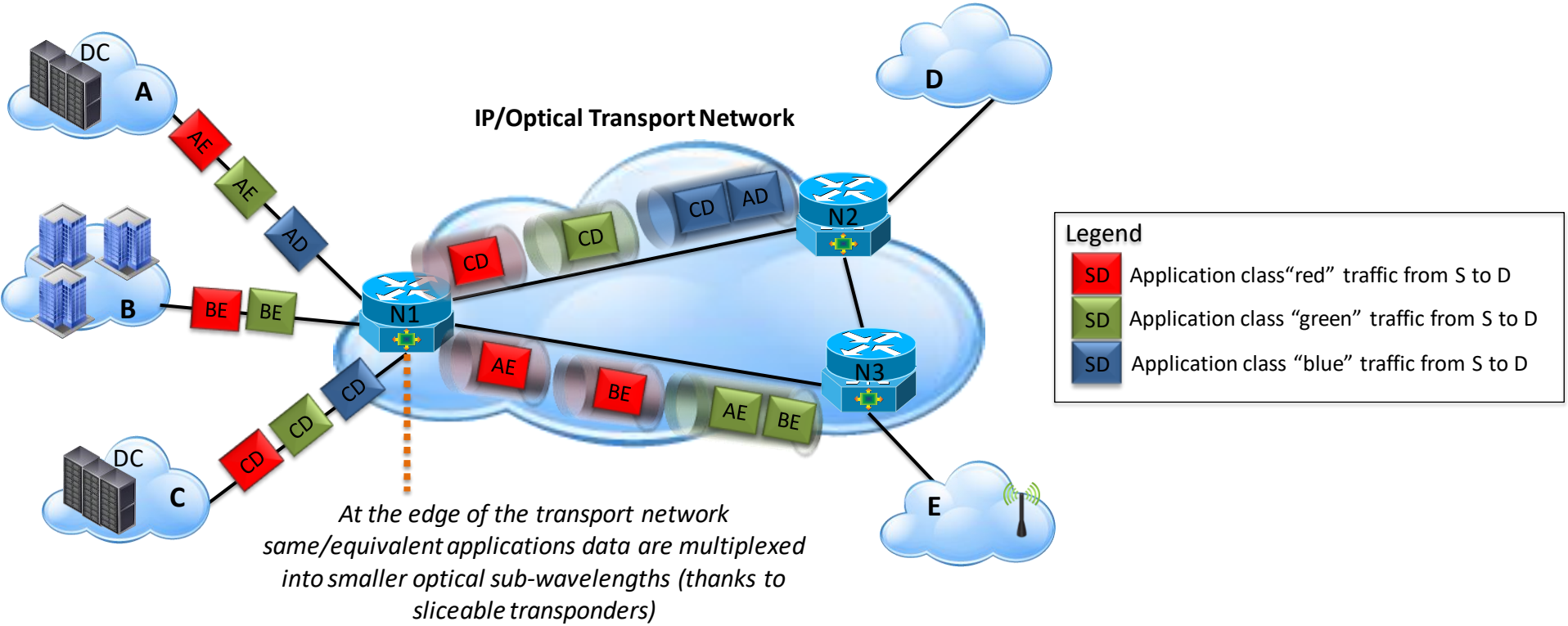
- All traffic served by an IP interface mapped into an optical connection sent towards the destination IP interface
- Application classes could be treated differently at the IP layer by its built-in QoS mechanisms

  → not currently a feature of the optical layer…

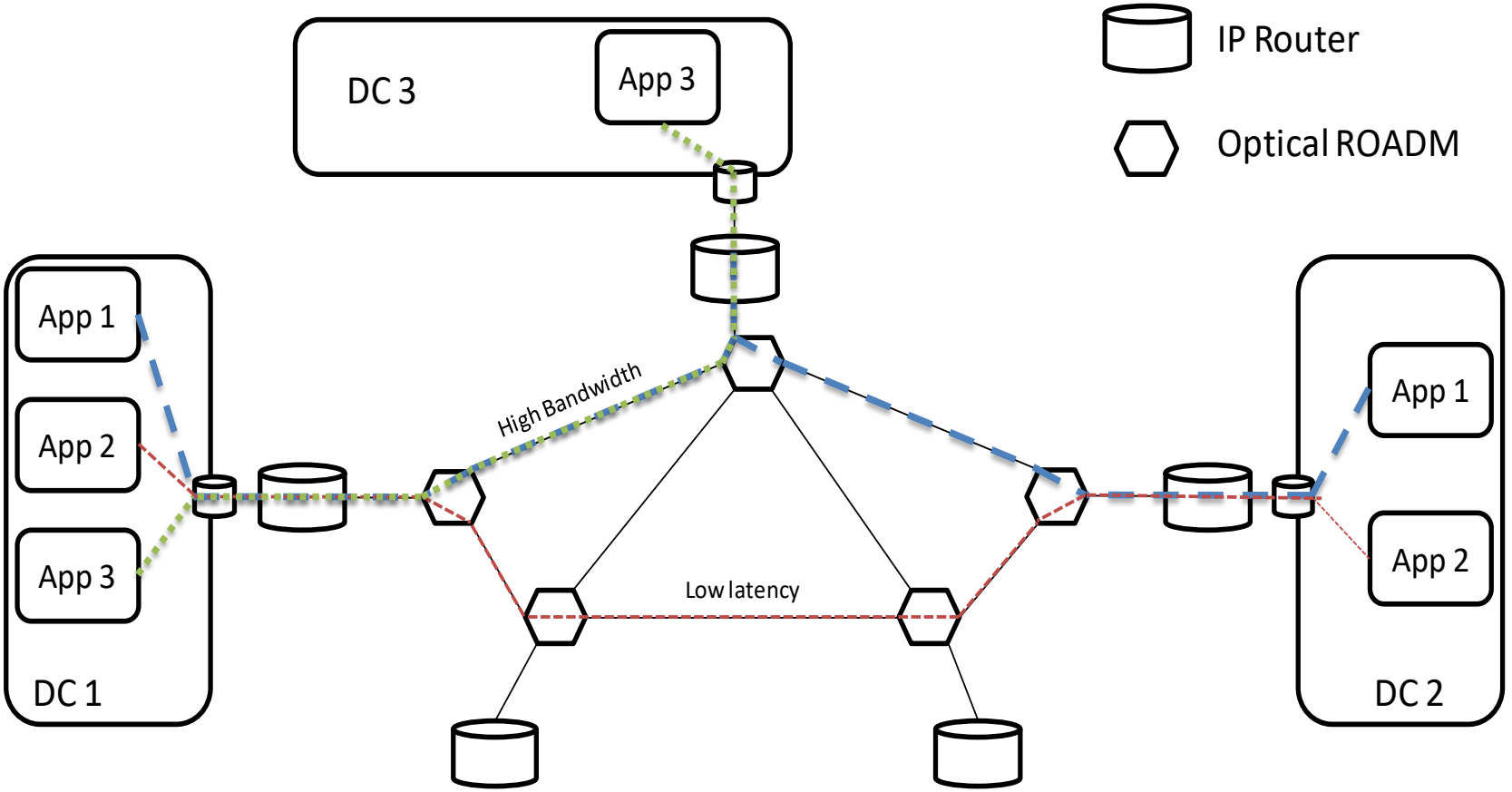


Approach:
… to application-centricity!

- *keep the different application classes separate down to the optical layer*
  - Different service (latency, survivability, security, …) for different apps



Latency awareness example

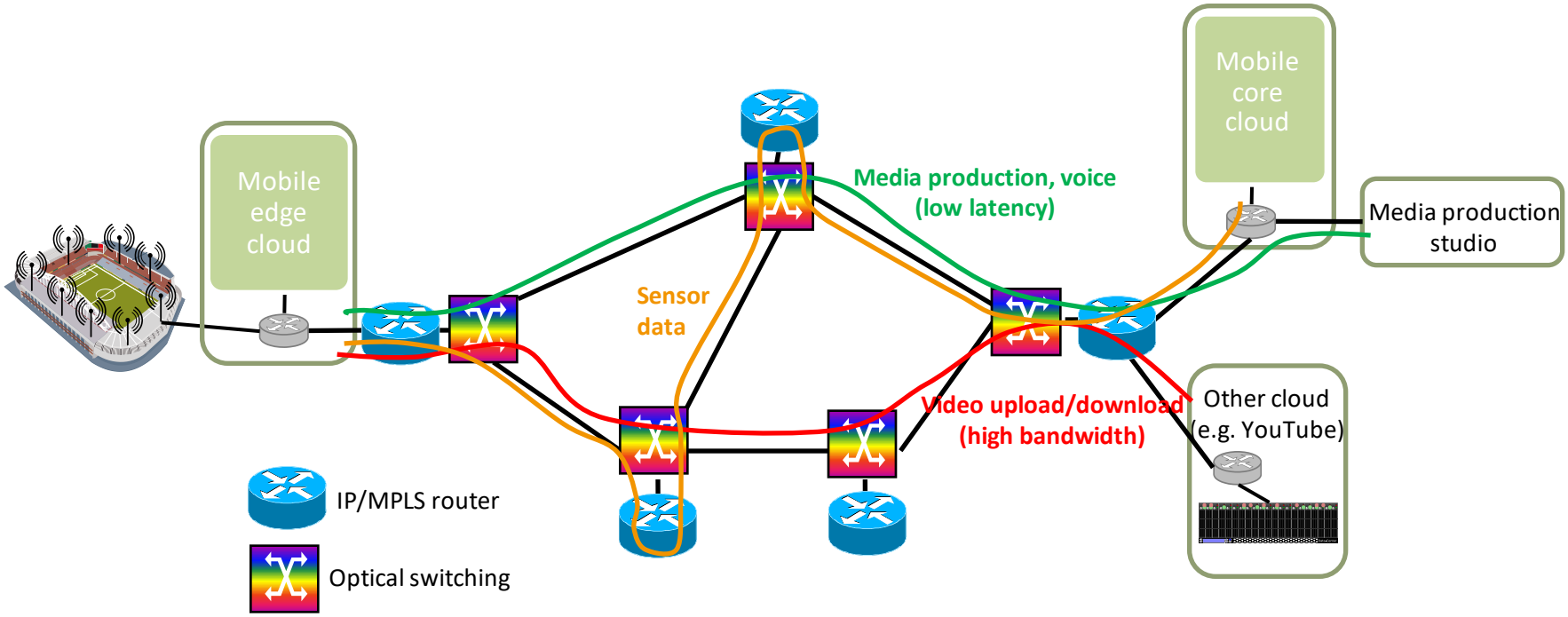


ACINO on a testbed

- Issues:
  1. Computed paths need to have *port* information, and Net2Plan does not have it
  2. Optical box has fixed filters built in, so only some optical paths are valid
  3. IP routers have grey interfaces to optical – these IP links must not not go up/down dynamically

- Solutions:
  1. Model ports as nodes
  2. Each “port” has a *color* attribute; only same color ports can connect
  3. Grey interface IP links have a *do not delete me* tag; they are never removed, and no new ones are added

5G network scenario

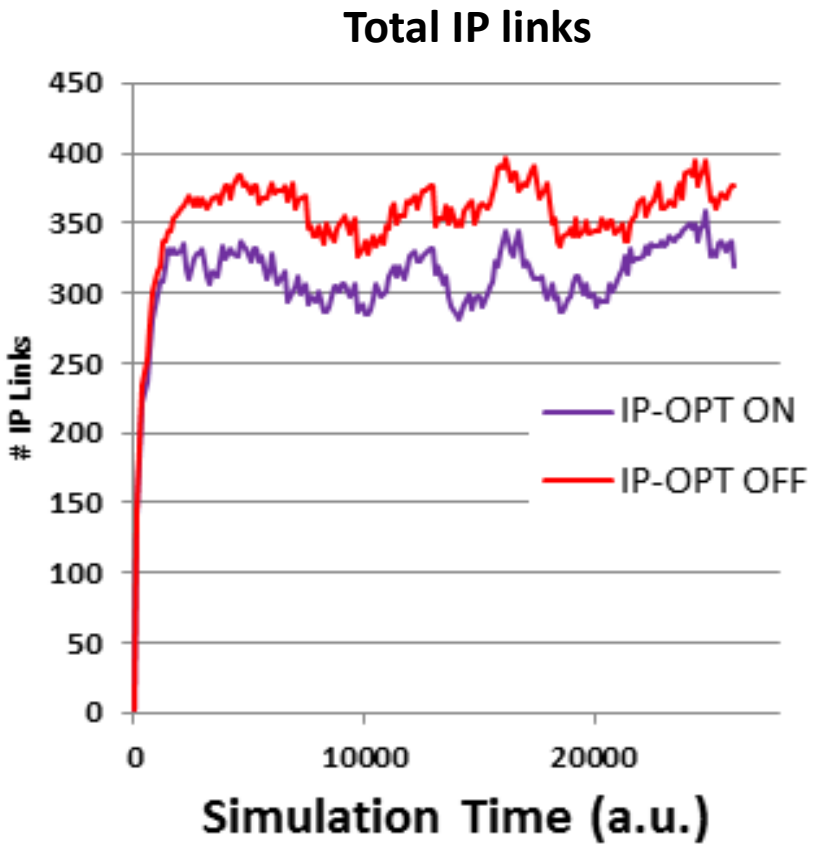


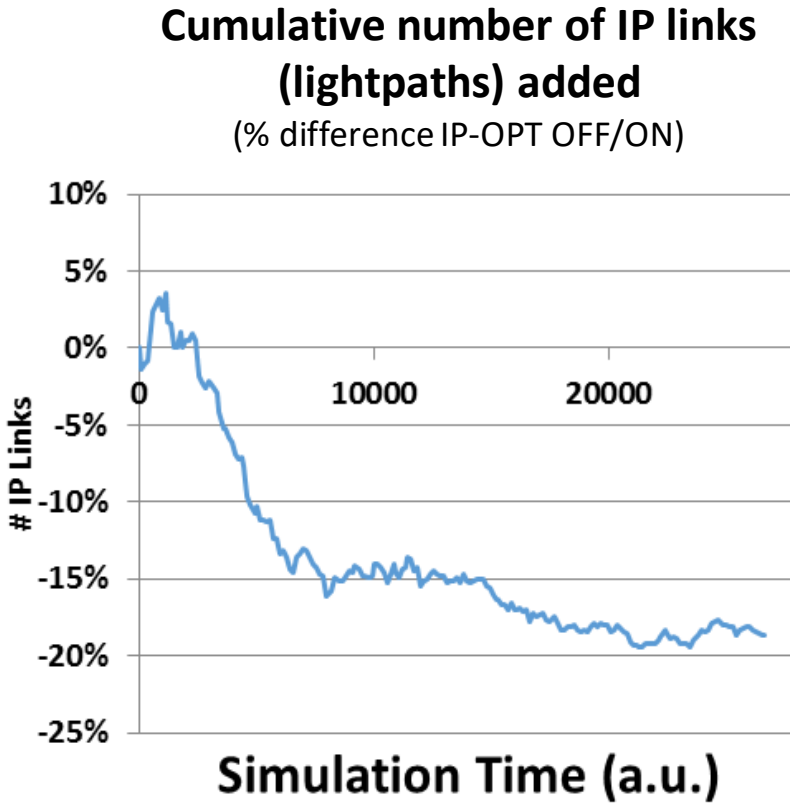
- The three applications take different paths through the network
  - Aligned with the emerging 5G network slicing concept

Scenario setup: Input traffic (2)

Traffic class	Percentage	Service requirements	Resource allocation policy
High priority	10%	Max latency: 6ms BW: 1-10 Gb/s	Route separately in optical
Latency-sensitive + Availability	40%	Max latency: 6ms Availability: 99.5% BW: 10-100 Gb/s	Aggregate in IP
Best effort	50%	BW: 10-100 Gb/s	
	100%		

IP-OPT evaluated (2)





IP-OPT saves resources but OPP is called more often