Virtual Network Embedding with Path-based Latency Guarantees in Elastic Optical Networks

Sepehr Taeb, Nashid Shahriar, Shihabur R. Chowdhury, Massimo Tornatore, Raouf Boutaba

Jeebak Mitra, Mahdi Hemmati

UNIVERSITY OF WATERLOO
FACULTY OF MATHEMATICS
David R. Cheriton School of Computer Science

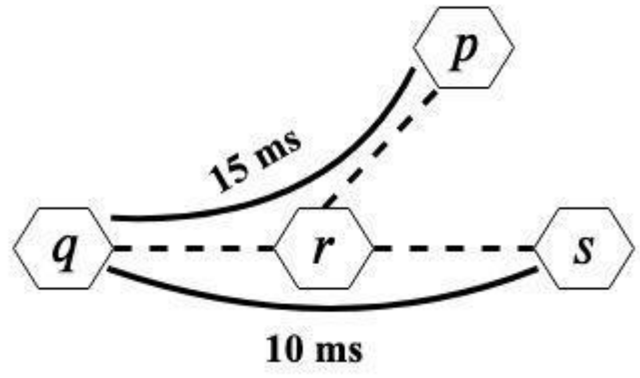
HUAWEI

Outline

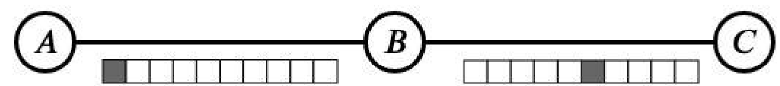
- Introduction
  - Elastic Optical Networks (EONs)
  - Key Contribution
  - Latency Model
- Problem Statement
- Integer Linear Program (ILP) Formulation
  - Constraints
  - Objective
- Heuristic Algorithm
- Evaluation
- Conclusion & Future Work

Introduction

- Many emerging applications have diverse latency requirements
  - Intelligent transportation, Industry automation, Online gaming, High-frequency trading
- An enabling technology to support latency-sensitive applications is **network virtualization**
  - Facilitates deployment of multiple virtual networks (VNs) with varying latency requirements on the same substrate network
  - Virtual network embedding maps VN nodes and links to substrate resources while guaranteeing latency constraints
- We focus on **transport network** as our substrate that connects Point of Presence (PoP) nodes
  - Optical network is the dominant technology due to its high-bandwidth and low-latency
  - Create lightpaths to embed virtual links



Virtual Network



Substrate Network

Elastic Optical Networks (EON)

- Traditional fixed-grid technology allocates spectrum in coarse-grained fashion
  - **Inefficient** - supports only 50 or 100 GHz wavelength grids
  - **Rigid** - allows limited transmission configurations for each data rate
- Elastic Optical Networks (EONs) are emerging to overcome the limitations
  - Enables **finer granularity** (12.5GHz) with arbitrary number of spectrum slices based on customer demand
  - Facilitates **tuning** of transmission configurations as per the need

Transmission configurations

Data Rate (Gbps)	Modulation format	FEC (%)	Spectrum bandwidth (GHz)	Reach (km)
100	QPSK	25	50	2000
200	QPSK	25	100	1000

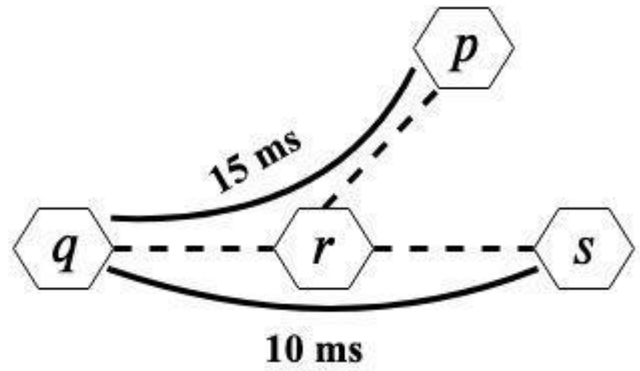
Traditional Optical Network

Data Rate (Gbps)	Modulation format	FEC (%)	Spectrum bandwidth (GHz)	Reach (km)
100	QPSK	25	50	2000
	16QAM	20	25	1250
200	QPSK	25	75	1000
	32QAM	20	37.5	400

Elastic Optical Network

Key contribution

- Existing literature represents latency requirements on **virtual links** (VLinks)
  - Cannot provide end-to-end latency guarantees
- We propose **path-based** latency requirements on virtual networks, called as VPath
  - Latency constraint is enforced along an entire path between PoPs
  - More flexibility in selecting substrate paths and transmission configurations for embedding VLinks
- How to distribute latency budgets to VLinks without violating path-based latency requirements?



VN request with path-based latency requirements

Latency model for a lightpath

- Node processing latency:
  - Transponders: ≈ 30 ns
  - FEC processing: ≈ 10 μs (standard) or ≈ 150 μs (super)

$$L_{node} = 2 \times (L_{transponder} + L_{FEC})$$

- Path latency
  - Fiber propagation: **4.9 μs/km**
  - Amplifiers: 150 ns
  - ROADMs: O(nano seconds)

$$L_{path} = len(p) \times L_{prop} + n_{amp} \times L_{amp} + (|p| + 1) \times L_{roadm}$$

- Zero queueing delay
  - By allocating dedicated resource on source and destination nodes
  - On intermediate nodes, data is optically switched - no queue buildup

Problem statement

Inputs:

- EON substrate Network
    - K-shortest path between each pair of nodes
- A set of transmission configurations
- VN request:
    - VLinks have bandwidth demand in Gbps
    - Path-based latency constraints
    - Given node mapping

Approach:

- Embedding a VLink by splitting its demand into multiple substrate paths
    - One path can be used more than once

Outputs:

- To embed each VLink, select
    - A set of substrate paths and appropriate transmission configurations
    - Spectrum slice allocation

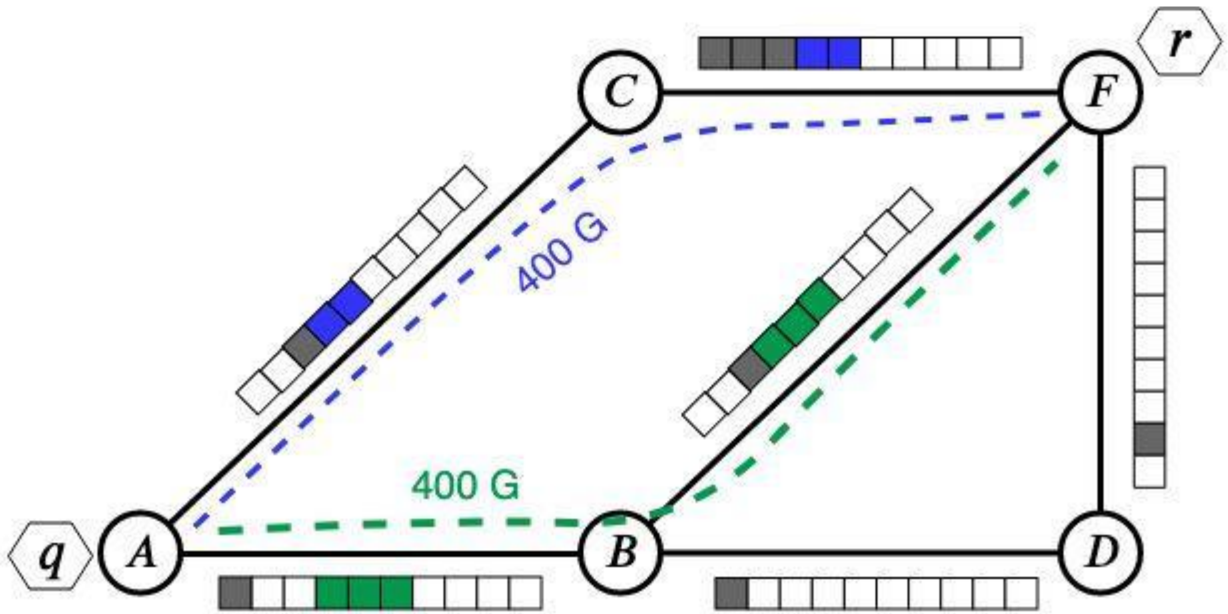
Objective:

- Minimize total spectrum resource allocation for the VN embedding (Primary)
- Minimize the total number of splits, i.e., transponders (Secondary)

Problem Formulation: Constraints

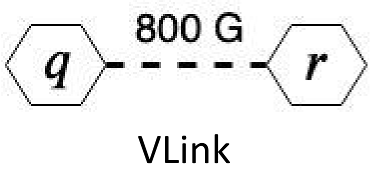
- An spectrum slice on a fiber link can be allocated to at most one split
- Each VLink demand is provisioned using up to a maximum (***q***) splits
    - Each split is realized using a transmission configuration satisfying its optical reach
    - Sum of the data rates carried by the splits is equal to the VLink demand

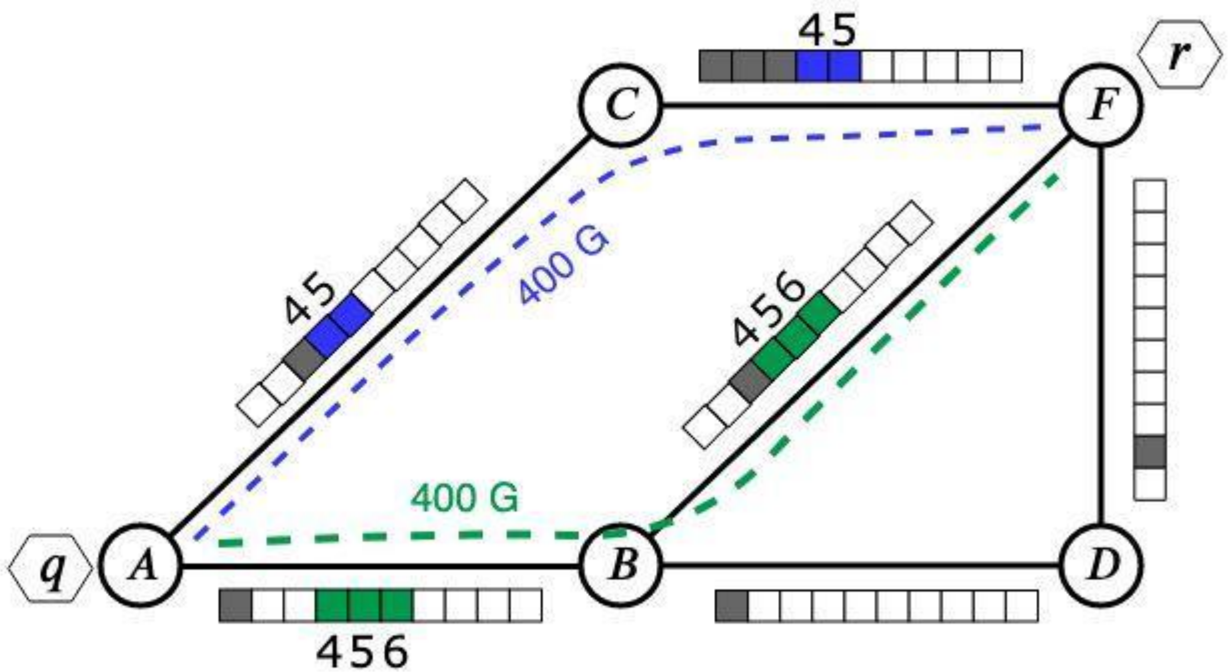
VLink



Constraints (Cont'd)

- Spectral contiguity and continuity:
  - Slices assigned to each split must be adjacent on each link of a substrate path (Contiguity)
  - Same set of slices should be assigned to each split along all links of a substrate path (Continuity)

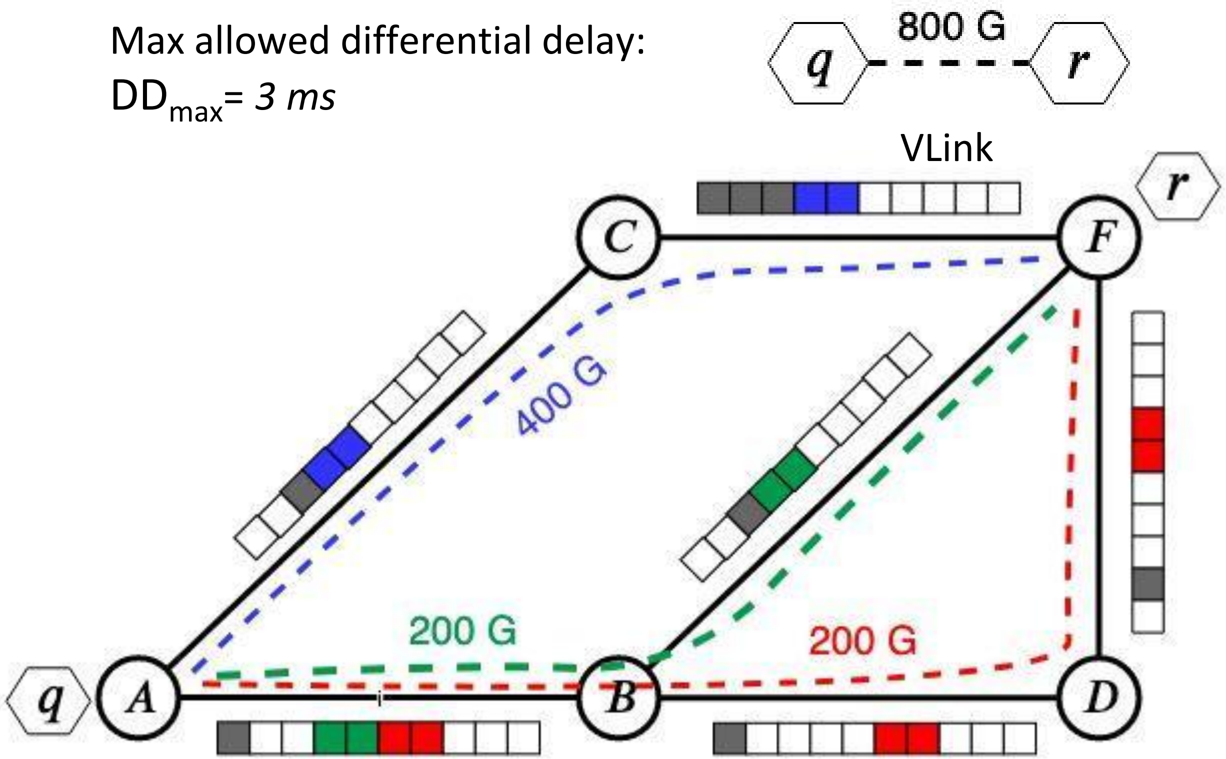




Constraints (Cont’d)

- Differential delay constraints:
  - The difference between the maximum and minimum latency of the splits provisioning a VLink should be less than DD_{max}

Max allowed differential delay:
$DD_{max} = 3\ ms$



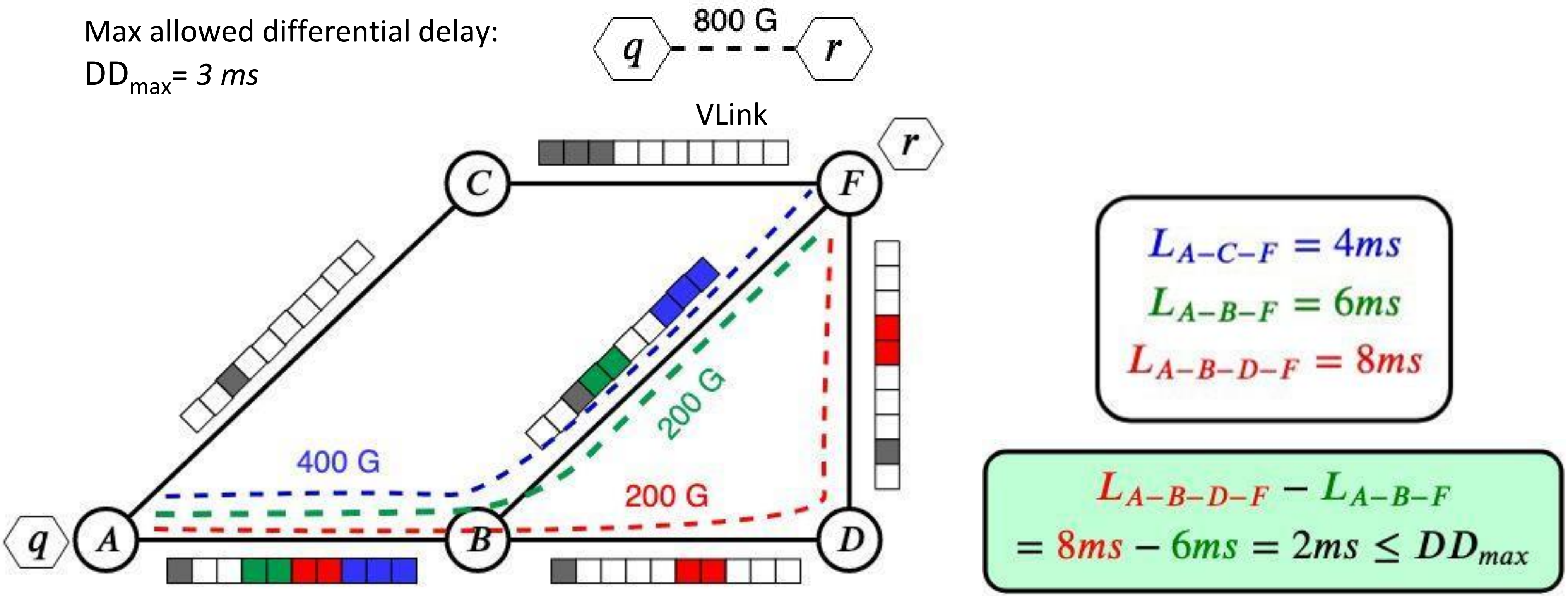
$$L_{A-C-F} = 4ms$$
$$L_{A-B-F} = 6ms$$
$$L_{A-B-D-F} = 8ms$$

$$L_{A-B-D-F} - L_{A-C-F} = 8ms - 4ms = 4ms \geq DD_{max}$$

Constraints (Cont'd)

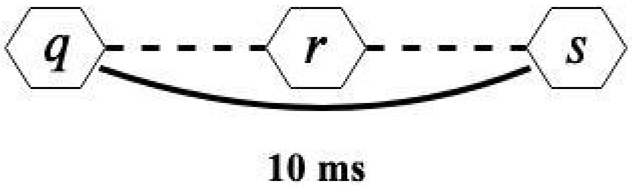
- Differential delay constraints:
  - The difference between the maximum and minimum latency of the splits provisioning a VLink should be less than DD_{max}

Max allowed differential delay:
DD_{max}= *3 ms*

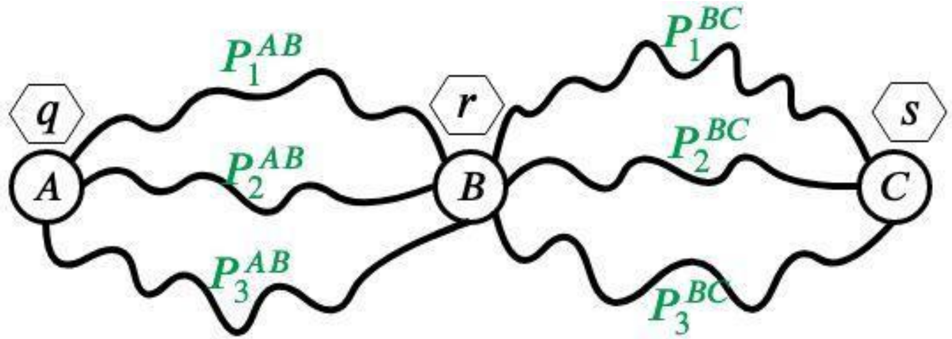


Constraints (Cont’d)

- Latency constraints for VPath:
  - The latency of each VLink embedding is equal to the maximum latency among its splits
  - The sum of the latencies of the VLinks on a VPath should satisfy the latency constraint



VN Request

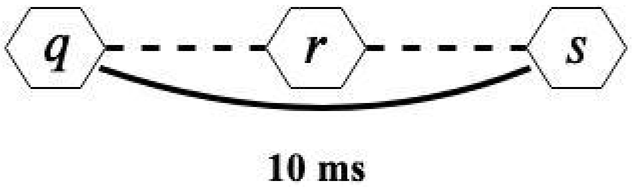


$$L_{P_1^{AB}} = 3ms$$
$$L_{P_2^{AB}} = 4ms$$
$$L_{P_3^{AB}} = 5ms$$

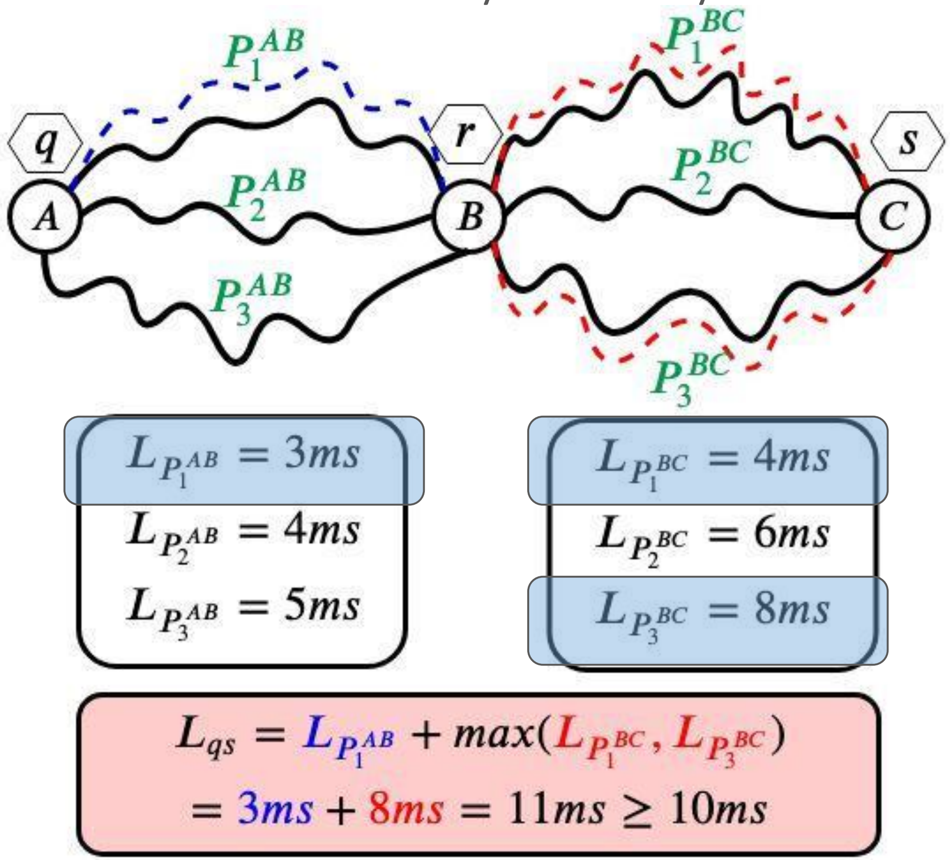
$$L_{P_1^{BC}} = 4ms$$
$$L_{P_2^{BC}} = 6ms$$
$$L_{P_3^{BC}} = 8ms$$

Constraints (Cont'd)

- Latency constraints for VPath:
  - The latency of each VLink embedding is equal to the maximum latency among its splits
  - The sum of the latencies of the VLinks on a VPath should satisfy the latency constraint

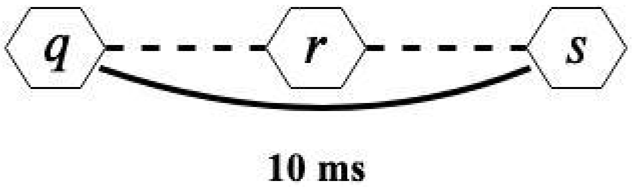


VN Request

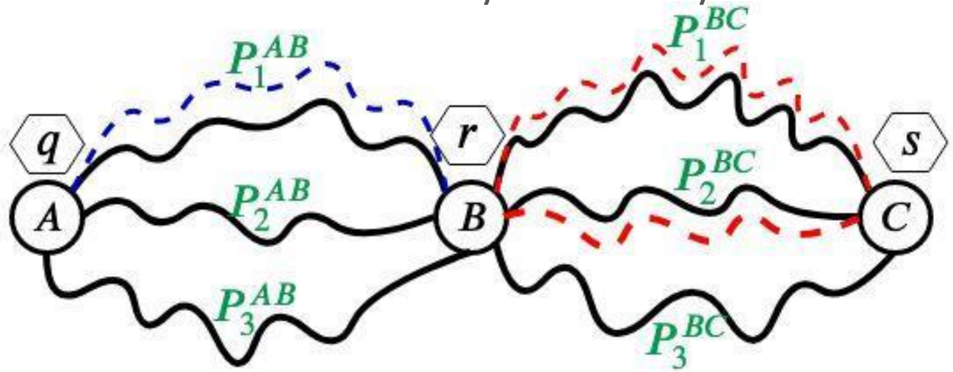


Constraints (Cont’d)

- Latency constraints for VPath:
  - The latency of each VLink embedding is equal to the maximum latency among its splits
  - The sum of the latencies of the VLinks on a VPath should satisfy the latency constraint



VN Request



$L_{P_1^{AB}} = 3ms$

$L_{P_2^{AB}} = 4ms$

$L_{P_3^{AB}} = 5ms$

$L_{P_1^{BC}} = 4ms$

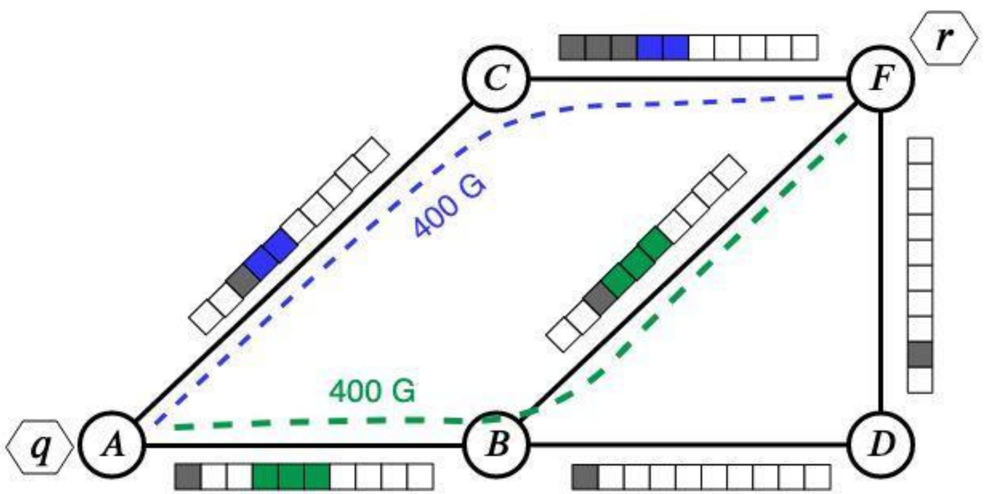
$L_{P_2^{BC}} = 6ms$

$L_{P_3^{BC}} = 8ms$

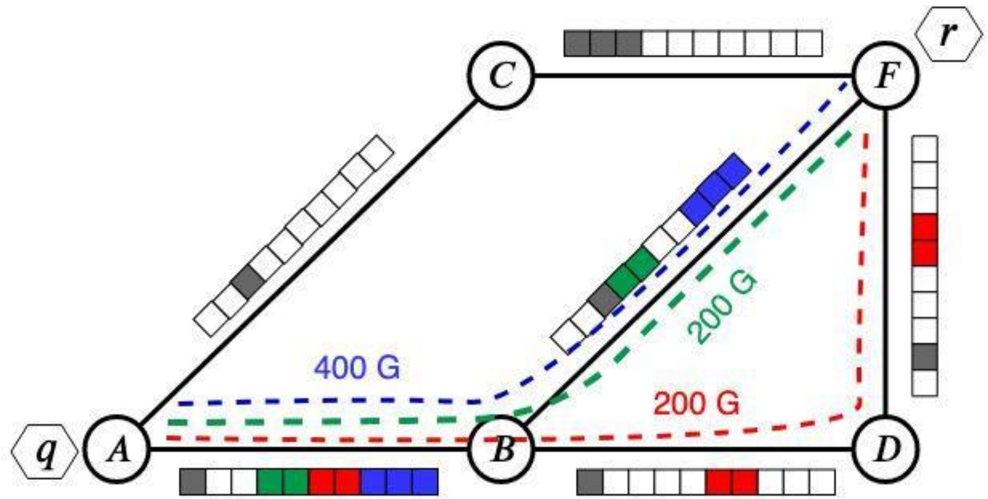
$$L_{qs} = L_{P_2^{AB}} + max(L_{P_1^{BC}}, L_{P_2^{BC}})$$
$$= 3ms + 6ms = 9ms \leq 10ms$$

Objective

- Minimize total spectrum resource allocation for the VN embedding (Primary)
- Minimize the total number of splits (Secondary)



Primary obj: 4 + 6 = 10 slices
Secondary obj = 2 splits



Primary obj: 6 + 4 + 6 = 16 slices
Secondary obj = 3 splits

Heuristic Algorithm

- Composed of 2 main steps
- Step 1: Choosing a VLink to be embedded next and computing an estimation of the latency budget for the VLink in terms of the candidate substrate paths
  - Most constrained VLink in terms of spectrum slice availability and latency
- Step 2: Finding an optimal embedding for the chosen VLink
  - Splits the VLink demand among multiple candidate paths
  - Uses the most spectrally efficient transmission configuration for each of the selected paths
  - Allocates spectrum slices on each link of the path
  - Finds the actual latency of the VLink based on the selected paths to help determine the latency of a VPath in step 1

Step 1: Finding Next VLink Algorithm

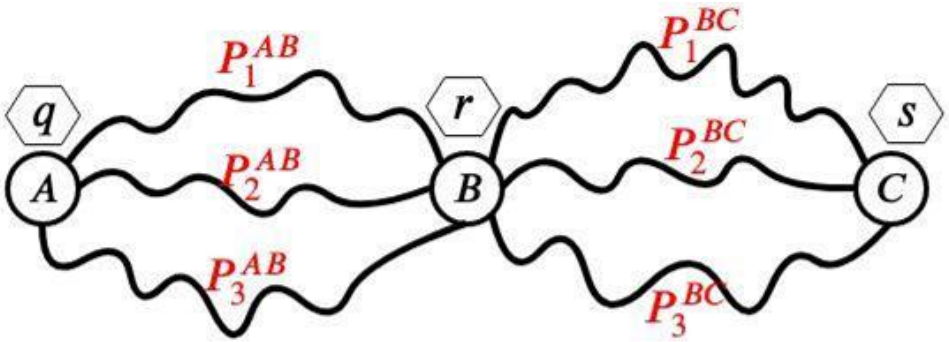
- Estimate latency budgets for all **VLinks** yet to be embedded
  - Assigned budgets do not violate any latency constraint
  - Determines the number of candidate paths to use for the VLinks
- The **number of available slices** on the candidate paths satisfying the assigned latency budget is maximized for the most constrained VLink
  - Spectrum resource availability is the bottleneck
  - Compute using binary search on the number of available spectrum slices
  - Check if the number of slices can be used without violating any latency constraint
- Return the VLink with the **minimum number of available slices** that does not violate the assigned latency budget

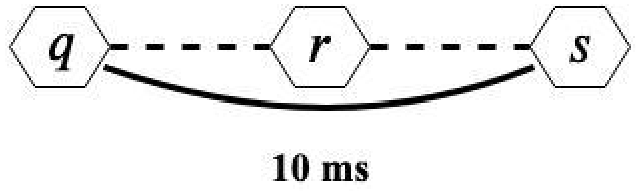
Step 1: Finding Next VLink Algorithm (Cont'd)

- Binary Search on the range of the number of available slices on the candidate paths
  - Finding the number of available slices on the candidate paths for each VLink
  - Get the minimum value of the number of available slices
  - Check if the number of slices can be used without violating any latency constraint
  - Do binary search
- Return the VLink with the **minimum number of available slices** that does not violate the assigned budget

Step 1: Finding Next VLink Algorithm (Cont'd)

Estimating a latency budget for each **VLink**





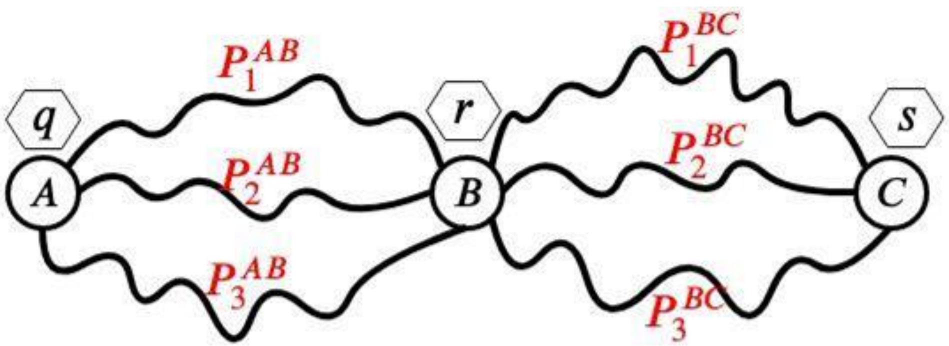
VN Request

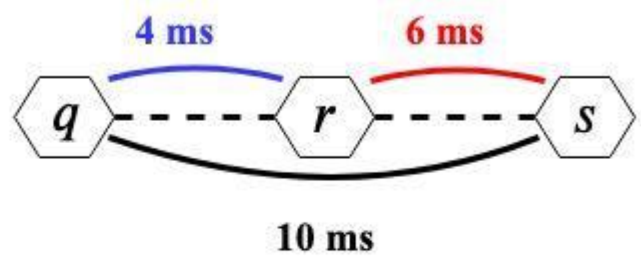
	L	Available slices
P_1^{AB}	$3ms$	4
P_2^{AB}	$4ms$	3
P_3^{AB}	$5ms$	1

	L	Available slices
P_1^{BC}	$4ms$	10
P_2^{BC}	$6ms$	4
P_3^{BC}	$8ms$	6

Step 1: Finding Next VLink Algorithm (Cont'd)

Estimating a latency budget for each **VLink**





VN Request

	L	Available slices
P_1^{AB}	$3ms$	4
P_2^{AB}	$4ms$	3
P_3^{AB}	$5ms$	1

7 Slices

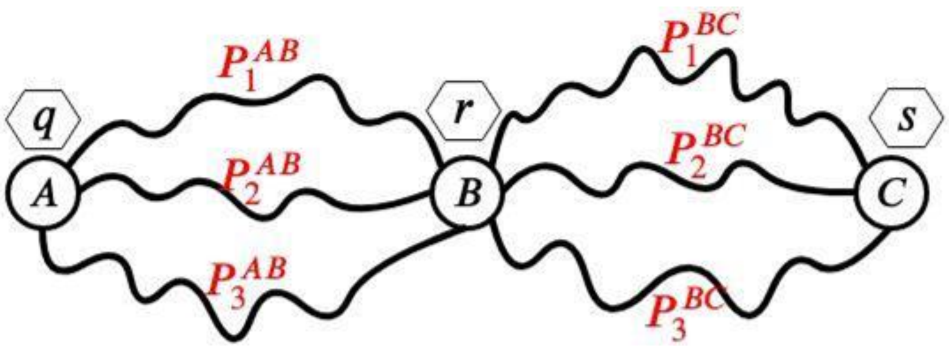
	L	Available slices
P_1^{BC}	$4ms$	10
P_2^{BC}	$6ms$	4
P_3^{BC}	$8ms$	6

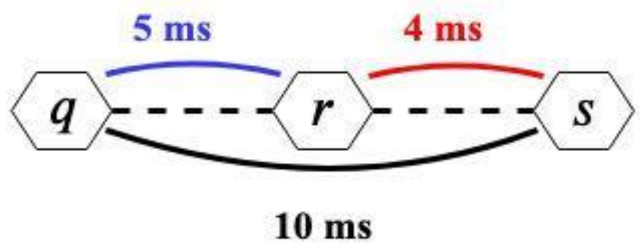
14 Slices

Goal: Maximize the number of slices for the VLink with minimum number of usable slices

Step 1: Finding Next VLink Algorithm (Cont’d)

Estimating a latency budget for each **VLink**





VN Request

	L	Available slices
P_1^{AB}	$3ms$	4
P_2^{AB}	$4ms$	3
P_3^{AB}	$5ms$	1

8 Slices

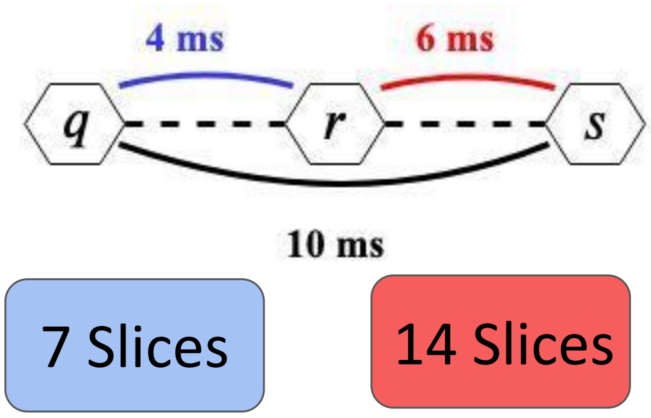
	L	Available slices
P_1^{BC}	$4ms$	10
P_2^{BC}	$6ms$	4
P_3^{BC}	$8ms$	6

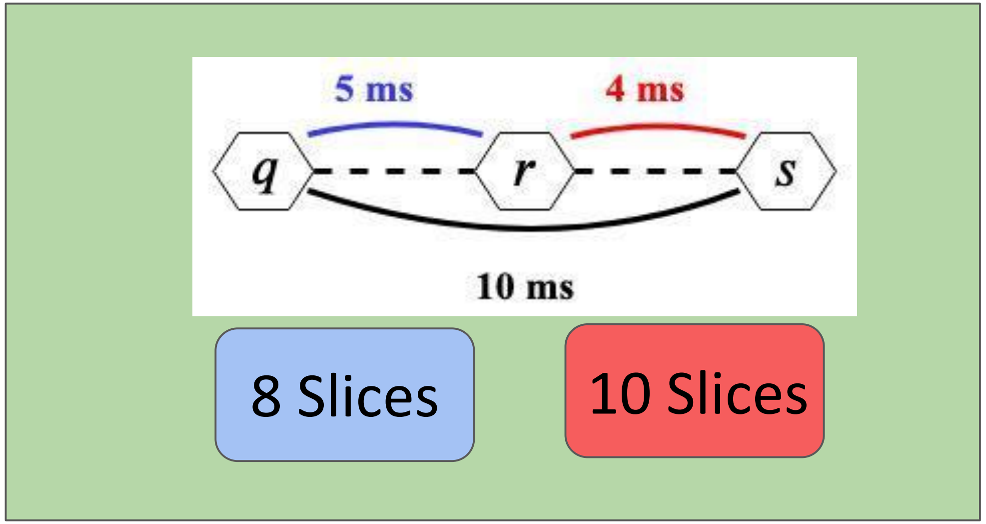
10 Slices

Goal: Maximize the number of slices for the VLink with minimum number of usable slices

Step 1: Finding Next VLink Algorithm (Cont'd)

Estimating a latency budget for each **VLink**





Goal: Maximize the number of slices for the VLink with minimum number of usable slices

Step 2: Optimal embedding for a VLink

- Compute link embedding using an exhaustive search considering all possible
  - Path selection (considering splitting)
    - All **multiset** of candidate paths with size <= q
    - Assigning **data rate** satisfying VLink demand

    } Specifying Splits
  - Transmission configuration selection
    - Choose a configuration supporting the **datarate** along the distance of a path in the multi-set
  - Spectrum slice assignment
    - **First-fit** slice allocation
- Select the combination of <path, transmission configuration, slice assignment> that minimizes the objective
  - Extends an algorithm published in [1]
- Additional pruning
  - Multi-sets of paths that violate differential delay constraint
  - Solutions requiring more slices than a lower bound computed using dynamic programming

1. Shahriar, Nashid et al. "Achieving a Fully-Flexible Virtual Network Embedding in Elastic Optical Networks." *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications* (2019): 1756-1764.

Step 2: Optimal embedding for a single VLink (Cont’d)

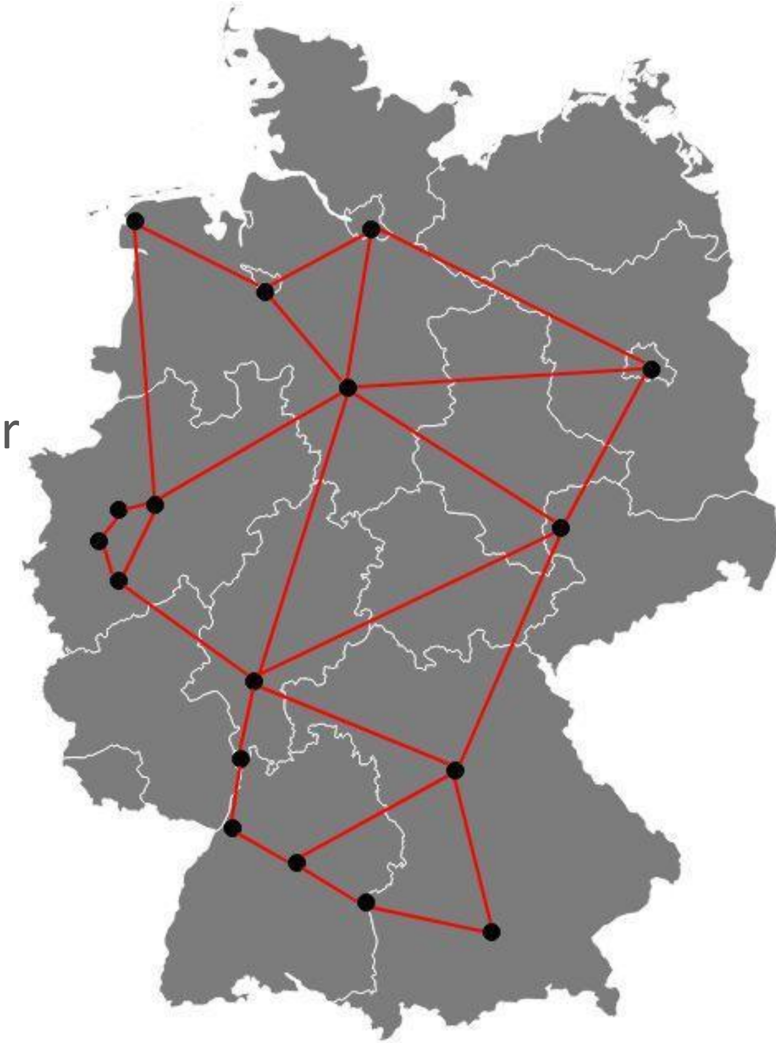
Optimization techniques:

- Pruning path multi-sets that violate differential delay constraint
- For a VLink with demand = D:
  - Solve the embedding for all d <= D and only one split
  - Find estimation for embedding using multiple splits (lower bound)
  - Only consider ones with better lower bound compared to best solution

Evaluation - Small Scale Benchmark

- Nobel Germany[1] EON
  - 17 Nodes and 26 Links
- Number of spectrum slices per link
  - Fixed grid: 12 slices of 50 GHz
  - Flex grid: 48 slices of 12.5 GHz
- Possible configurations provided by industry partner
- Max number of splits (q) is 4
- VNs are generated synthetically
  - Fixed node mapping
  - 8 VNodes
  - Variable LNR: from 1 to 2.5 (8 to 20 VLinks)
  - Latencies: Latency of the shortest path * α (α >= 1)

$$L(\alpha) = L(\text{path with lowest latency}) \times \alpha$$



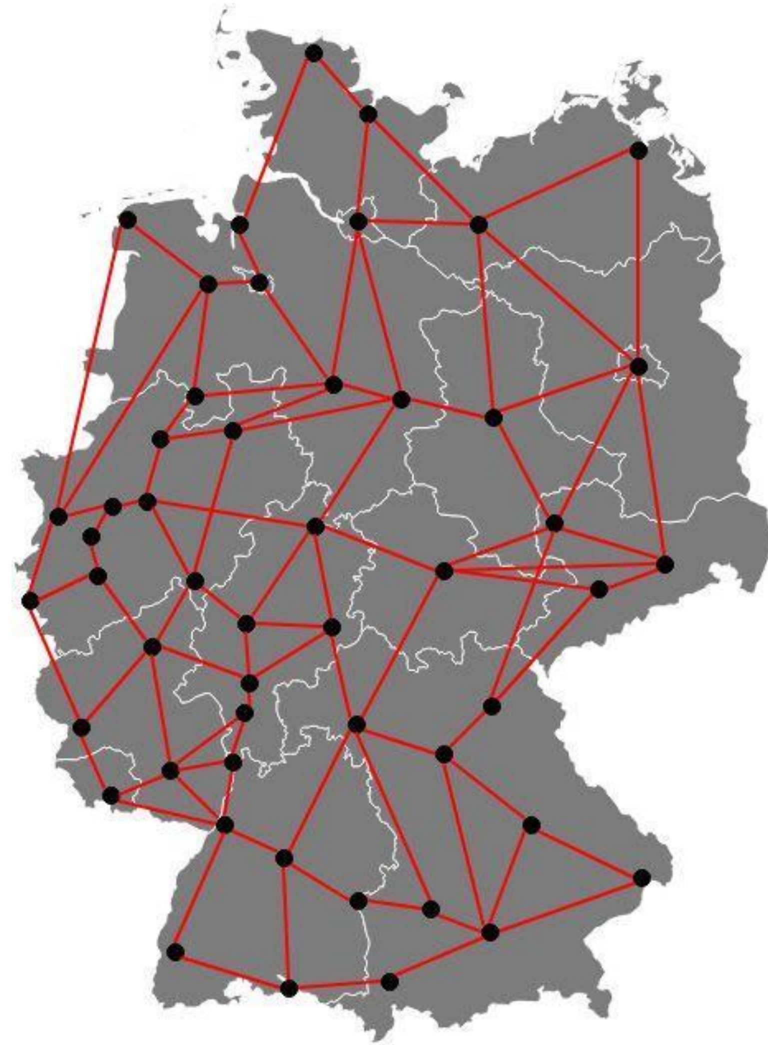
1. http://sndlib.zib.de/

Evaluation - simulation settings

Large Scale:

- EON: Germany50[1]: 50 Nodes, 88 Links
- Number of spectrum slices per link:
  - Fixed grid: 80 slices of 50 GHz
  - Flex grid: 320 slices of 12.5 GHz
- VNs are generated synthetically
  - Fixed node mapping
  - 50 VNodes
  - Variable LNR: from 1 to 3.5 (50 to 175 VLinks)
  - Latencies: Latency of the shortest path * α (α >= 1)

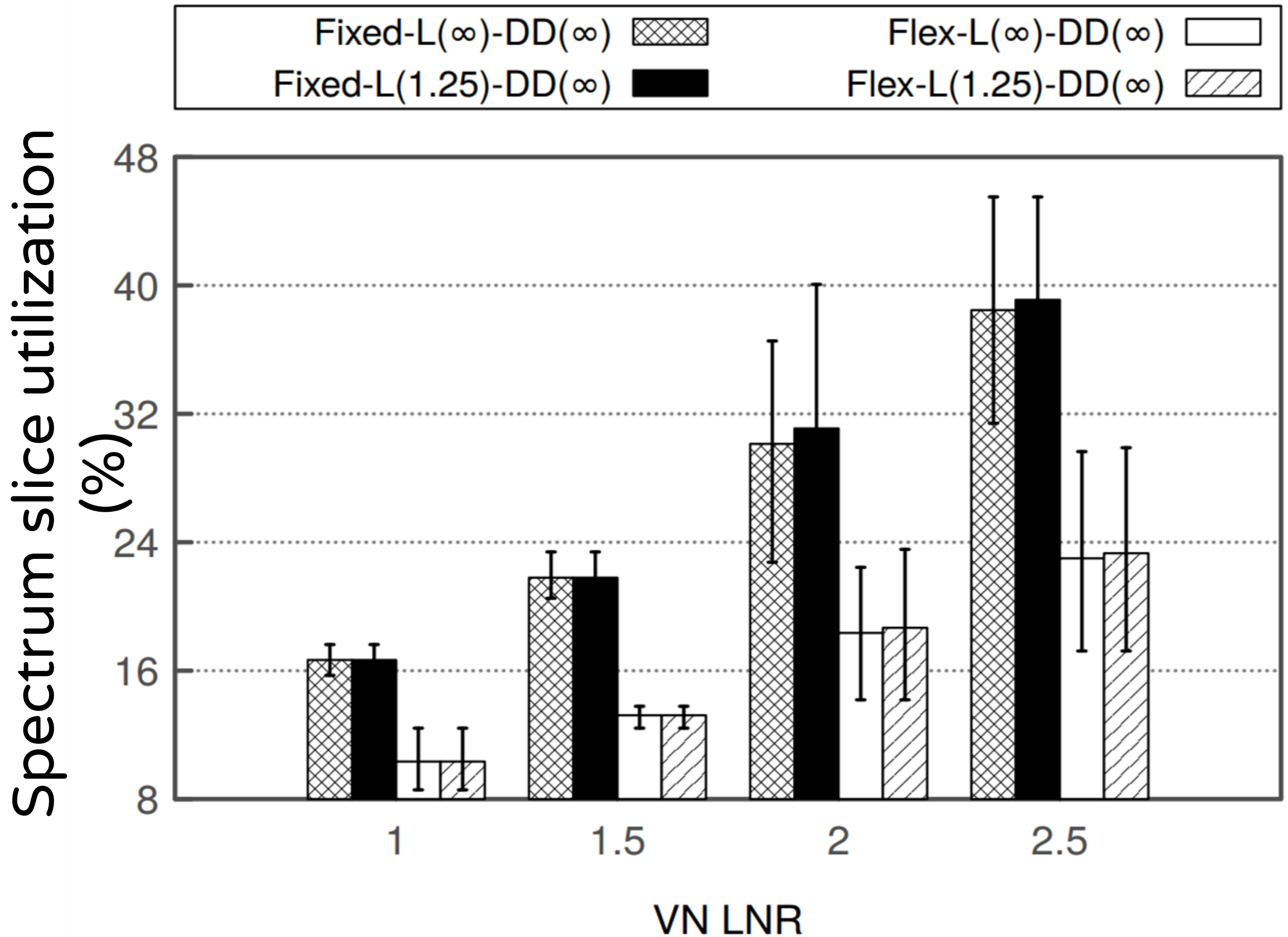
$$L(\alpha) = L(\text{path with lowest latency}) \times \alpha$$



1. http://sndlib.zib.de/

Evaluation - Small Scale Benchmark

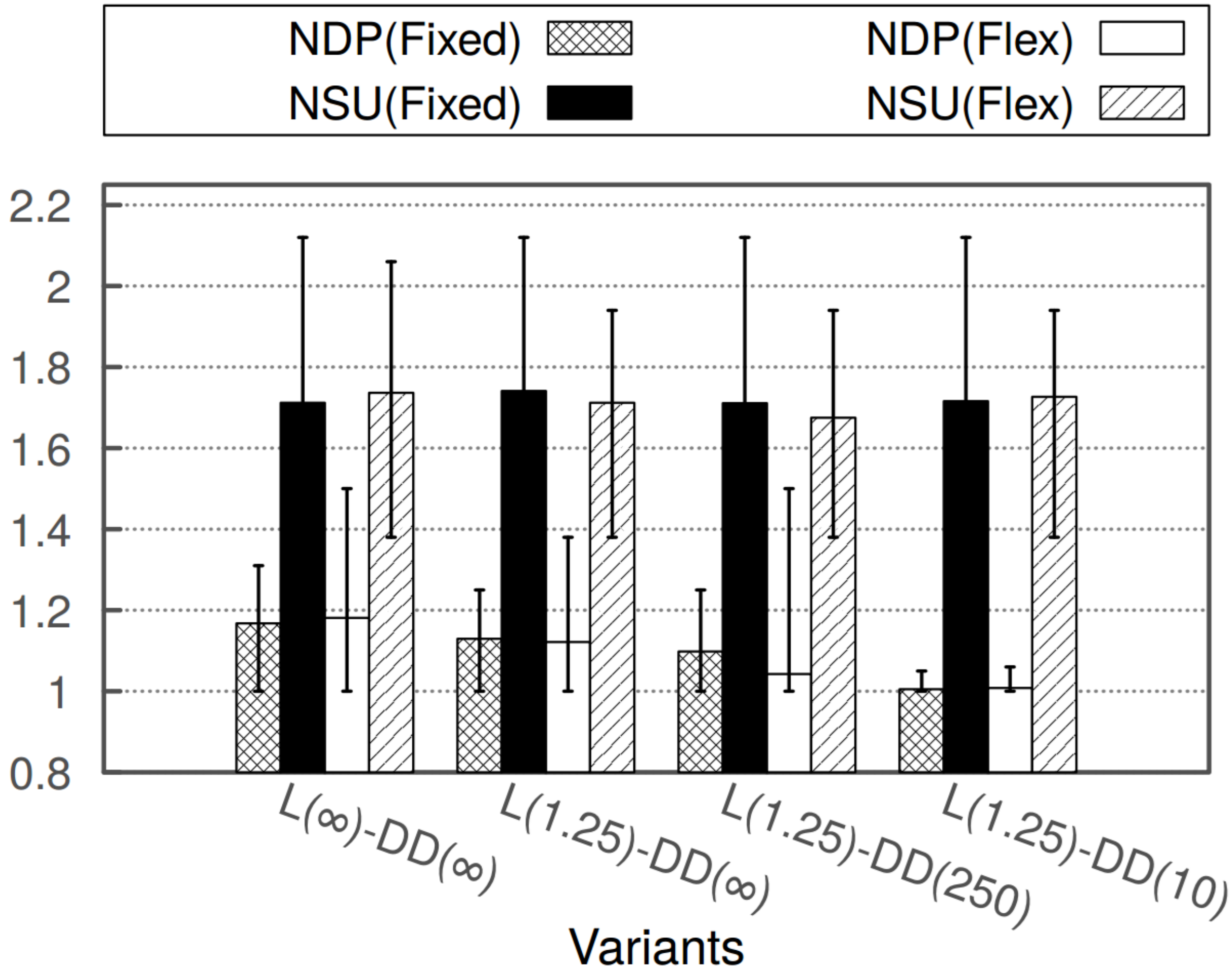
- Impact of the latency constraints on resource utilization
- Compared variants
  - Fixed-L(α)-DD(β):
    - Fixed grid
    - α: latency factor
    - β: max differential delay
  - Flex-(α)-DD(β):
    - Flex grid
    - α: latency factor
    - β: max differential delay



Evaluation - Small Scale Benchmark

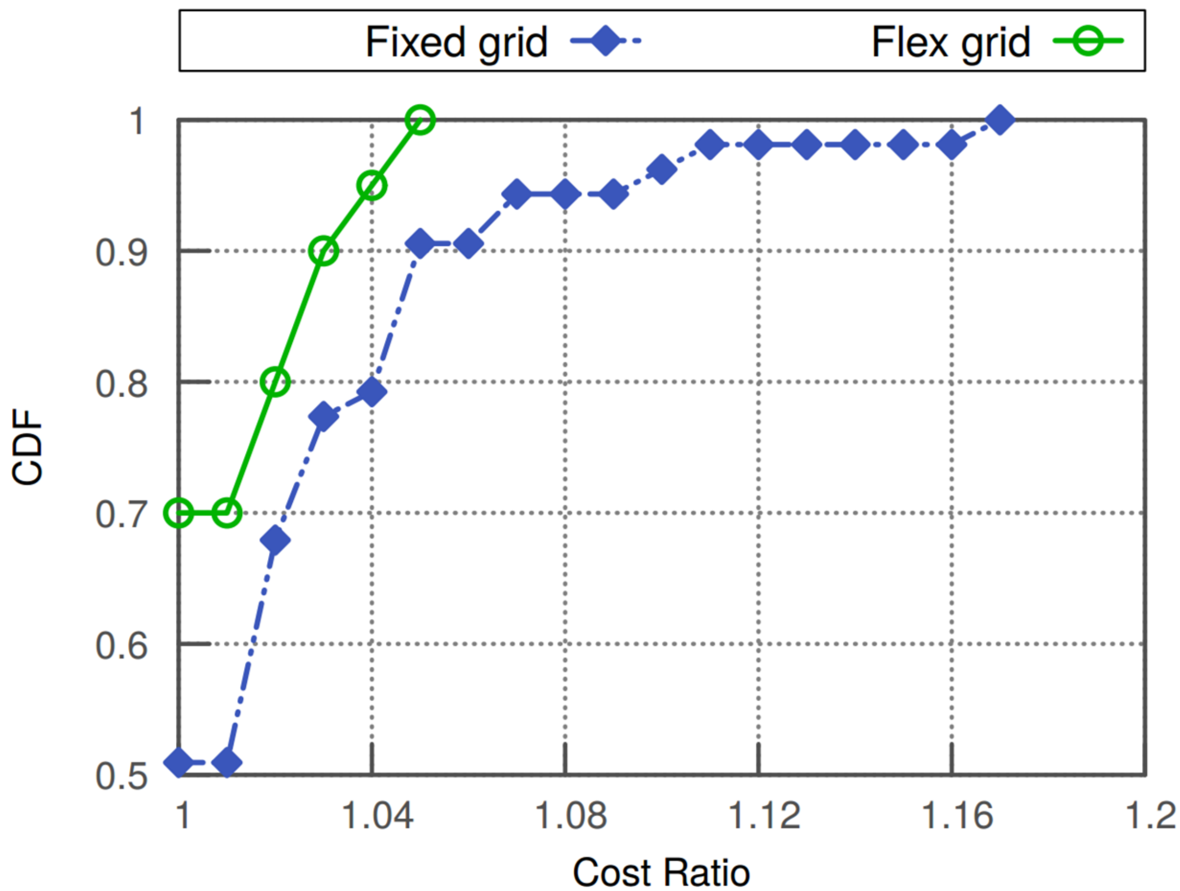
- Impact of differential delay on substrate path selection

- Metrics
  - NDP (Fixed/Flex)
    - Avg. number of **distinct path** used to embed a VLink
  - NSU (Fixed/Flex)
    - Avg. number of **splits** used to embed a VLink

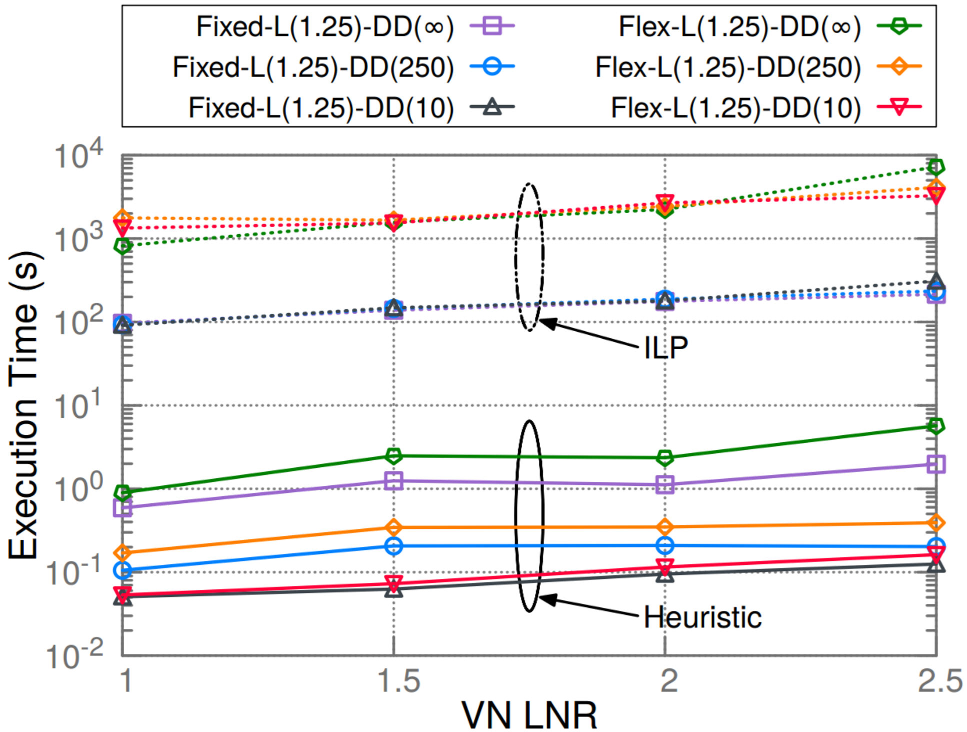


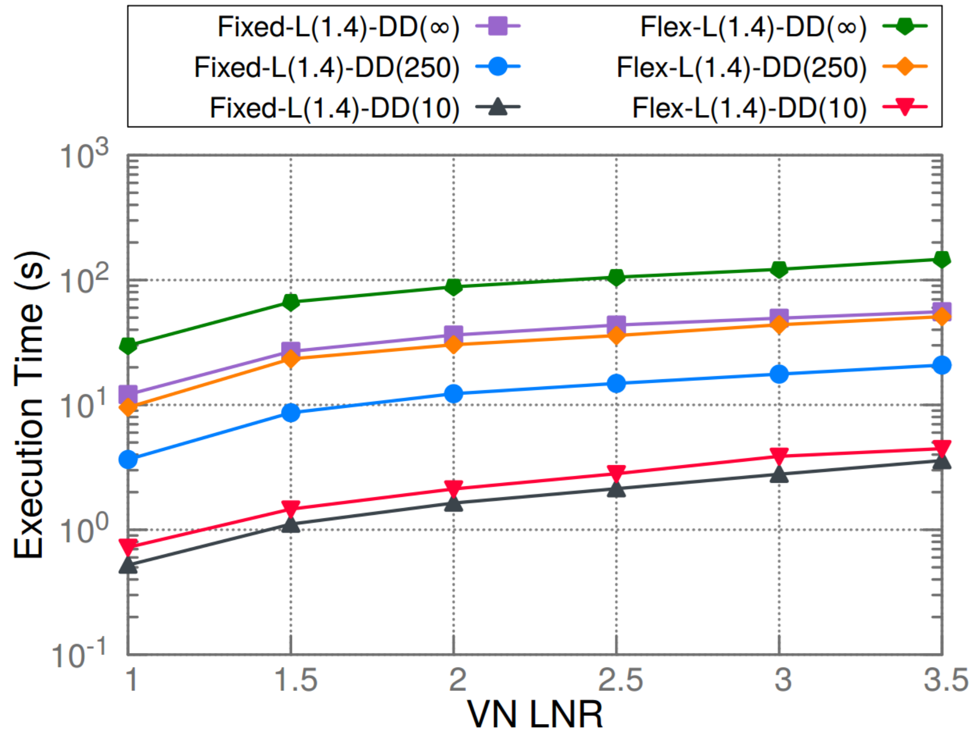
Evaluation - Small Scale Benchmark

- Optimality of heuristic
- Compared variants
  - Fixed grid EON
  - Flex grid EON
  - Varying latency and differential delay for both cases



Evaluation - scalability

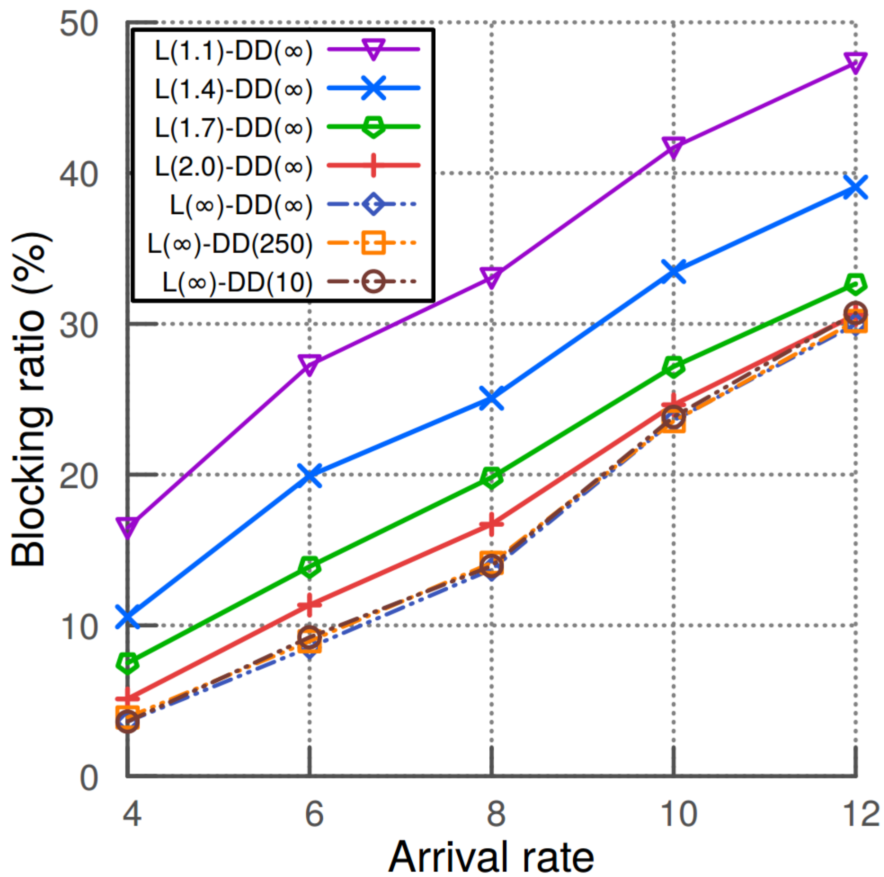




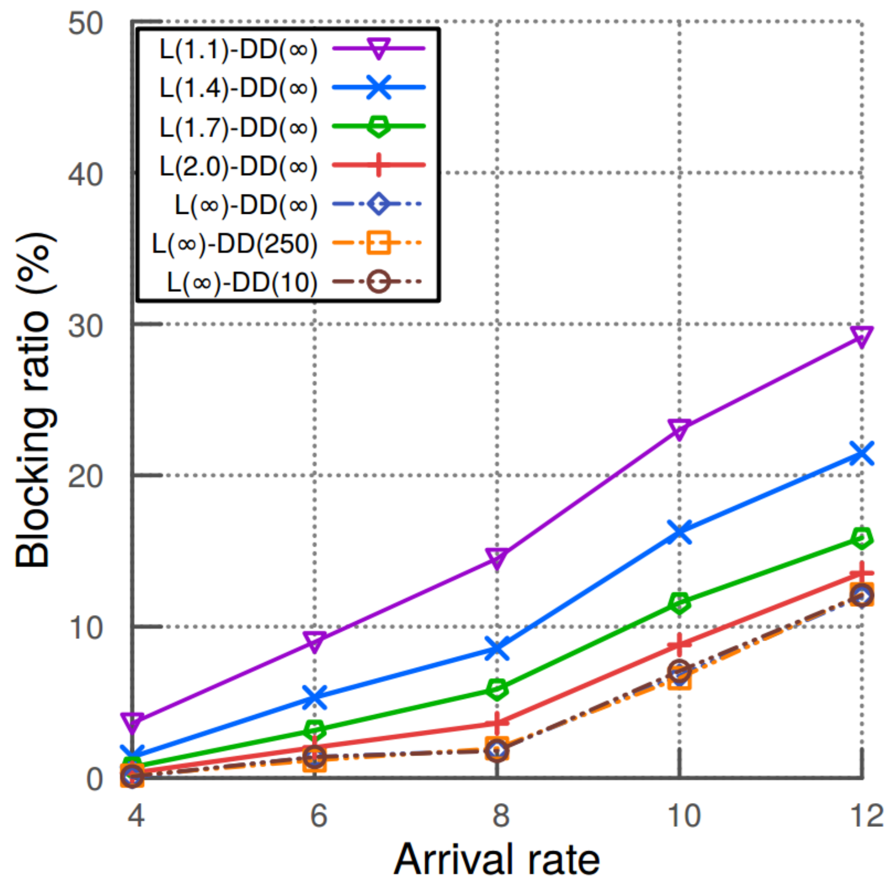
Evaluation - Steady State Analysis

- Arrival and departure time for VNs
  - Arrival rate: Poisson distribution
    - 4 to 12 VNs per 100 time units
  - VN life time: Exponential distribution
    - Mean of 100 time units
- VN and SN properties
  - 8 VNodes
  - Random number of VLinks: 8 to 28
  - Nobel Germany flex grid EON: **320 slices of 12.5 GHz**
- Simulation time: **10000** time units
  - Excluding the first 1000 time units
- 5 different simulation scenarios
- Report VN **blocking ratio**
  - Percentage of VNs that could not be embedded

Evaluation - Steady State Analysis

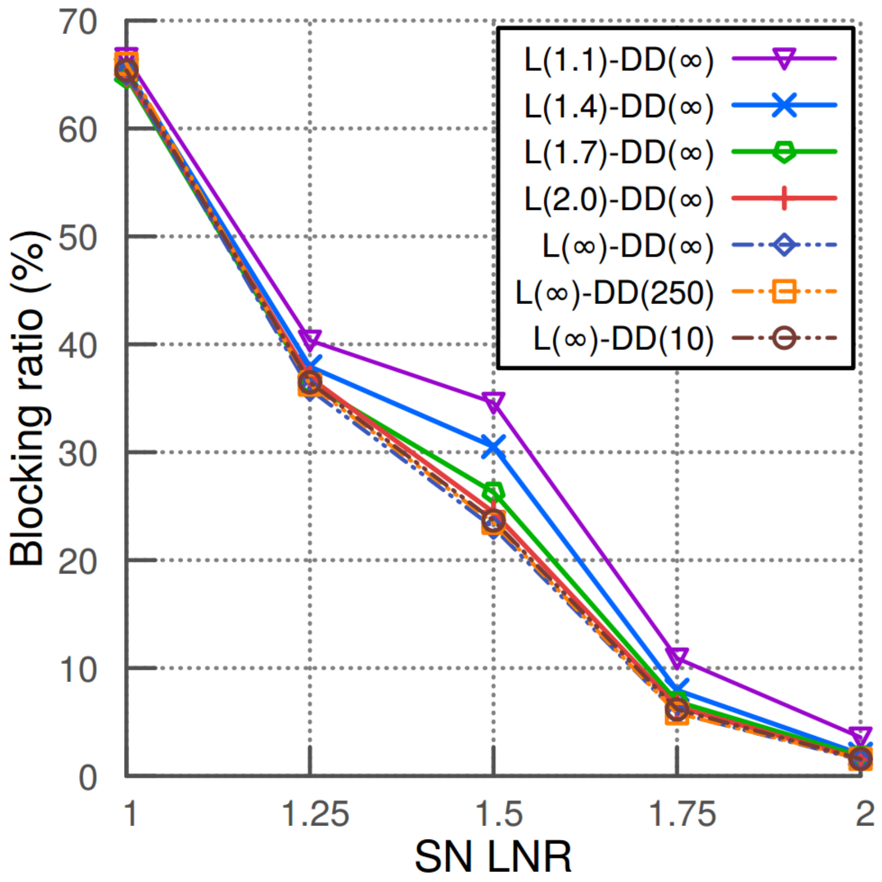


Fixed Grid

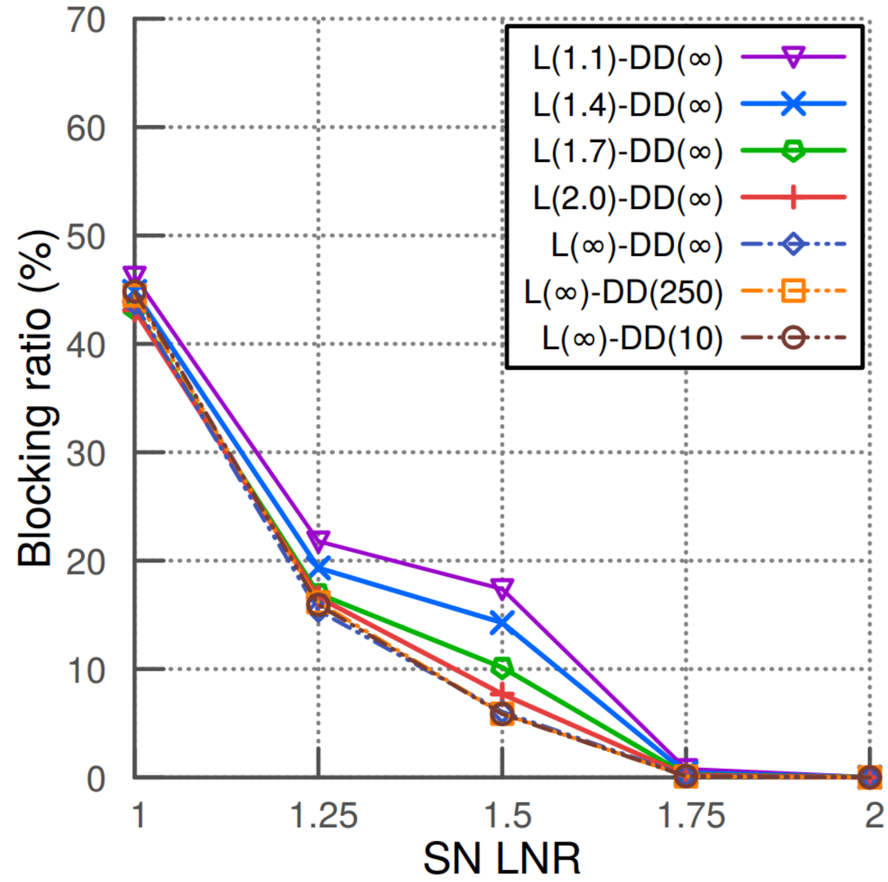


Flex Grid

Evaluation - Steady State Analysis



Fixed Grid



Flex Grid

Conclusion & Future Work

- Virtual network embedding over EON
  - Path-based latency guarantees
  - Considering full flexibility in all transmission parameters of an EON
- An ILP based optimization model
- A faster heuristic algorithm that obtains near optimal solutions
- Key takeaways
  - Latency constraints has less impact on spectrum usage but profound impact on blocking
  - Flexibilities of an EON help reduce these impact
- Future work
  - Different cost function to decrease blocking probability
  - Design an admission control to maximize the revenue

Thank You!