Fundamentals of Network Design Modelling by Integer Linear Programming (WDM networks case)

Network Design and Planning (sq2014)

Massimo Tornatore
Dept. of Computer Science
University of California, Davis
Outline

- Introduction to WDM optical networks and network design
- WDM network design and optimization
  - Integer Linear Programming approach
  - Physical Topology Design
    - Unprotected case
    - Dedicated path protection case
    - Shared path & link protection cases
  - References
- Heuristic approach
Success of optical communications

Main technical reasons

- **Optical fiber advantages**
  - Huge bandwidth (WDM)
  - Long range transmission (EDFA optical amplifiers)
  - Strength
  - Use flexibility (transparency)
  - Low noise
  - Low cost
  - Interference immunity
  - ....

- **Optical components**
  - Rapid technological evolution
  - Increasing reliability (not for all…)
  - Decreasing costs (not for all…)

- **Ok, but from a network perspective?**
  - *Convergence of services over a unique transport platform*
WDM optical networks: a “layered vision”

- **WDM layer fundamentals**
  - Wavelength Division Multiplexing: information is carried on high-capacity channels of different wavelengths on the same fiber
  - Switching: WDM systems transparently switch optical flows in the space (fiber) and wavelength domains

- **WDM layer basic functions**
  - Optical circuit (LIGHTPATH) provisioning for the electronic layers
  - Common transport platform for a multi-protocol electronic-switching environment
What’s a WDM System?

EO Converter

EO Converter

EO Converter

Ch 1
1300 nm

Ch 2
1310 nm

Ch n
850 nm

Passive Optical Multiplexer

Ch 1
Ch 2
Ch n
Wavelength Switching in WDM Networks
The concept of lightpath
Example: A European WDM Network

Helsinki

Madrid
The concept of lightpath
Example: A European WDM Network

Helsinki

Madrid
Outline

- Introduction to WDM network design and optimization
- Integer Linear Programming approach
- Physical Topology Design
  - Unprotected case
  - Dedicated path protection case
  - Shared path & link protection cases
- Heuristic approach
WDM networks: basic concepts

**Logical topology**

- **Logical topology (LT):** each link represent a lightpath that could be (or has been) established to accommodate traffic.
- A lightpath is a “logical link” between two nodes.
- Full mesh Logical topology: a lightpath is established between any node pairs.
- LT Design (LTD): choose, minimizing a given cost function, the lightpaths to support a given traffic.
WDM networks: basic concepts

Physical topology

- Physical topology: set of WDM links and switching-nodes
- Some or all the nodes may be equipped with wavelength converters
- The capacity of each link is dimensioned in the design phase
Different OXC configurations

*Jane Simmons, “Optical network design and planning”

Fig. 2.17 Examples of optical-switch architectures. (a) O-E-O architecture with electronic switch fabric and electronic interfaces on all ports (Section 2.6.1). (b) Photonic switch that switches the 1310-nm optical signal (Section 2.6.2). (c) A wavelength-selective all-optical switch (Section 2.6.3).
WDM networks: basic concepts

Mapping of the logical over the physical topology

Mapping is different according to the fact that the network is not (a) or is (b) provided with wavelength converters.

LP = LIGHTPATH

- $\lambda_1$
- $\lambda_2$
- $\lambda_3$

Optical wavelength channels

- Solving the resource-allocation problem is equivalent to perform a mapping of the logical over the physical topology
  - Also called Routing Fiber and Wavelength Assignment (RFWA)
- Physical-network dimensioning is jointly carried out
Lightpaths and Wavelength Routing

- Lightpath
- Virtual topology
- Wavelength-continuity constraint
- Wavelength conversion
Illustrative Example
Static WDM network planning
Problem definition

- **Input parameters, given *a priori***
  - Physical topology (OXC nodes and WDM links)
  - Traffic requirement (logical topology)
    - Connections can be **mono** or bidirectional
    - Each connection corresponds to one lightpath than must be setup between the nodes
    - Each connection requires the full capacity of a wavelength channel (no traffic grooming)

- **Parameters which can be specified or can be part of the problem**
  - Network resources: two cases
    - **Fiber-constrained**: the number of fibers per link is a preassigned global parameter (typically, in mono-fiber networks), while the number of wavelengths per fiber required to setup all the lightpaths is an output
    - **Wavelength-constrained**: the number of wavelengths per fiber is a preassigned global parameter (typically, in multi-fiber networks) and the number of fibers per link required to setup all the lightpaths is an output
Physical constraints

- Wavelength conversion capability
  - Absent (wavelength path, WP)
  - Full (virtual wavelength path, VWP)
  - Partial (partial virtual wavelength path, PVWP)

- Propagation impairments
  - The length of lightpaths is limited by propagation phenomena (physical-length constraint)
  - The number of hops of lightpaths is limited by signal degradation due to the switching nodes

- Connectivity constraints
  - Node connectivity is constrained; nodes may be blocking

Links and/or nodes can be associated to weights

- Typically, link physical length is considered
Routing can be
- Constrained: only some possible paths between source and destination (e.g. the $K$ shortest paths) are admissible
  - Great problem simplification
- Unconstrained: all the possible paths are admissible
  - Higher efficiency in network-resource utilization

Cost function to be optimized (optimization objectives)
- Route all the lightpaths using the minimum number of wavelengths (physical-topology optimization)
- Route all the lightpaths using the minimum number of fibers (physical-topology optimization)
- Route all the lightpaths minimizing the total network cost, taking into account also switching systems (physical-topology optimization)
Routing and Wavelength Assignment (RWA) [OzBe03]
- The capacity of each link is given
- It has been proven to be a NP-complete problem [ChGaKa92]
- Two possible approaches
  - Maximal capacity given ⇒ maximize routed traffic (throughput)
  - Offered traffic given ⇒ minimize wavelength requirement

Routing Fiber and Wavelength Assignment (RFWA)
- The capacity of each link is a problem variable
- Further term of complexity ⇒ Capacitated network
- The problem contains multicommodity flow (routing), graph coloring (wavelength) and localization (fiber) problems
- It has been proven to be a NP-hard problem (contains RWA)
Dimensions of Complexity…
Optimization problems

Classification

- Optimization problem – optimization version
  - Find the minimum-cost solution

- Optimization problem – decision version (answer is yes or no)
  - Given a specific bound $k$, tell me if a solution $x$ exists such that $x<k$

- Polynomial problem
  - The problem in its optimization version is solvable in a polynomial time

- NP problem
  - Class of decision problems that, under reasonable encoding schemes, can be solved by polynomial time non-deterministic algorithms

- NP-complete problem
  - A NP problem such that any other NP problem can be transformed into it in a polynomial time
  - The problem is very likely not to be in $P$
  - In practice, the optimization-problem solution complexity is exponential

- NP-hard problem
  - The problem in its decision version is not solvable in a polynomial time (is NP-complete) $\Rightarrow$ the optimization problem is harder than an NP-problem
  - Contains an NP-complete problem as a subroutine
WDM network optimal design is a very complex problem. Various approaches proposed

- **Mathematical programming (MP)**
  - Exact method (guarantees optimal solution)
  - Computationally expensive, not scalable
- Heuristic methods
  - An alternative to MP for realistic dimension problems

According to the cost function, the problem is

- Linear
- Non linear
Outline

- Introduction to WDM network design and optimization
- Integer Linear Programming approach
- Physical Topology Design
  - Unprotected case
  - Dedicated path protection case
  - Shared path & link protection cases
- Heuristic approach
Optimization problem solution
Mathematical programming solving

- WDM-network static-design problem can be solved with the mathematical programming techniques
  - In most cases the cost function is linear ⇒ linear programming
  - Variables can assume integer values ⇒ integer linear programming

- LP solution
  - Variables defined in the real domain
  - The well-known computationally-efficient Simplex algorithm is employed

- ILP solution
  - Variables defined in the integer domain
  - The optimal integer solution is found by exploring all integer admissible solutions
    - Branch and bound technique: admissible integer solutions are explored in a tree-like search
The ILP models: Notation

- $l,k$: link identifiers (source and destination nodes)
- $F_{l,k}$: number of fibers on the link $l,k$
- $x_{l,k}$: number of wavelengths on the link $l,k$
- $c_{l,k}$: weight of link $l,k$ (es. length, administrative weight, etc.)
  - Usually equal to $c_{k,l}$
Outline

- Introduction to WDM network design and optimization
- Integer Linear Programming approach
- Physical Topology Design
  - Unprotected case
  - Dedicated path protection case
  - Shared path & link protection cases
- References
- Heuristic approach
Approaches to WDM design

- Two basilar and well-known approaches [WaDe96],[Wi99]
  - FLOW FORMULATION (FF)
  - ROUTE FORMULATION (RF)
Flow Formulation (FF)

- Flow variable $x_{l,k}^{s,d}$
- Flow on link $(l,k)$ due to a request generated by source-destination couple $(s,d)$

- Fixed number of variables
- Unconstrained routing
Variables represent the amount of traffic (flow) of a given traffic relation (source-destination pair) that occupies a given channel (link, wavelength)

- Lightpath-related constraints
  - Flow conservation at each node for each lightpath (solenoidal constraint)
  - Capacity constraint for each link
  - (Wavelength continuity constraint)
  - Integrity constraint for all the flow variables (lightpath granularity)

- Allows to solve the RFWA problems with unconstrained routing
- A very large number of variables and constraint equations
ILP application to WDM network design

Flow formulation fundamental constraints

- **Solenoidality constraint**
  - Guarantees spatial continuity of the lightpaths (flow conservation)
  - For each connection request, the neat flow (tot. input flow – tot. output flow) must be:
    - zero in transit nodes
    - the total offered traffic (with appropriate sign) in s and d

- **Capacity constraint**
  - On each link, the total flow must not exceed available resources (# fibers x # wavelengths)

- **Wavelength continuity constraint**
  - Required for nodes without converters
Notation

- $c$: node pair (source $s_c$ and destination $d_c$) having requested one or more connections
- $x_{l,k,c}$: number of WDM channels carried by link $(l,k)$ assigned to a connection requested by the pair $c$
- $A_i$: set of all the nodes adjacent to node $i$
- $v_c$: number of connection requests having $s_c$ as source node and $d_c$ as destination node
- $W$: number of wavelengths per fiber
- $\lambda$: wavelength index ($\lambda=\{1, 2 \ldots W\}$)
## Unprotected case

**VWP, FF**

<table>
<thead>
<tr>
<th>Solenoidality</th>
<th>[\sum_{k \in A_i} x_{k,l,c} - \sum_{k \in A_i} x_{l,k,c} = \begin{cases} v_c &amp; \text{if } l = d_c \ -v_c &amp; \text{if } l = s_c \ 0 &amp; \text{otherwise} \end{cases} \quad \forall l, c]</th>
</tr>
</thead>
</table>

| Capacity      | \[\sum_{c} x_{l,k,c} \leq W \cdot F_{l,k} \quad \forall (l,k)\] |

| Integrity     | \[x_{l,k,c} \text{ integer} \quad \forall c,(l,k)\] \[F_{l,k} \text{ integer} \quad \forall (l,k)\] |

N.B. from now on, notation “\(\forall l,k\)” implies \(l \neq k\)
Note, the unsplittable case!

| Solenoidality       | \( \sum_{k \in A_l} x_{k,l,c} - \sum_{k \in A_l} x_{l,k,c} = \begin{cases} 
1 & \text{if } l = d_c \\
-1 & \text{if } l = s_c \\
0 & \text{otherwise} \end{cases} \forall l, c \) |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>( \sum_c v_c x_{l,k,c} \leq W \cdot F_{l,k} \quad \forall (l, k) )</td>
</tr>
<tr>
<td>Integrity</td>
<td>( x_{l,k,c} ) binary \quad \forall c, (l, k) ) \quad ( F_{l,k} ) integer \quad \forall (l, k) )</td>
</tr>
</tbody>
</table>
Cost functions

Some examples

- **RFWA**
  - Minimum fiber number $M$
    - Terminal equipment cost
      - VWP case
  - Minimum fiber mileage (cost) $M_C$
    - Line equipment [BaMu00]

- **RWA**
  - Minimum wavelength number
  - Minimum wavelength mileage
    - [StBa99],[FuCeTaMaJa03]
  - Minimum maximal wavelength number on a link
    - [Mu97]

\[
\begin{align*}
\min \sum_{(l,k)} F_{l,k} &= \min M \\
\min \sum_{(l,k)} c_{l,k} F_{l,k} &= \min M_C \\
\min \sum_{(l,k)} x_{l,k} &= \min \Gamma \\
\min \sum_{(l,k)} c_{l,k} x_{l,k} &= \min \Gamma_C \\
\min \left[ \max_{l,k} x_{l,k} \right] &= \min \Gamma_{MAX}
\end{align*}
\]
Observation

- What happens in the bidirectional case?
  - i.e., Each transmission channel provides the same capacity \( \lambda \) in both directions.
Extension to WP case

- In absence of wavelength converters, each lightpath has to preserve its wavelength along its path.
- This constraint is referred to as \textit{wavelength continuity} constraint.
- In order to enforce it, let us introduce a new index in the flow variable to analyze each wavelength plane.

\[
x_{l,k,c} \Rightarrow x_{l,k,c,\lambda}
\]

- The structure of the formulation does not change. The problem is simply split on different planes (one for each wavelength).
- The $v_c$ traffic is split on distinct wavelengths \( \Rightarrow v_{c,\lambda} \)
- The same approach will be applied for no-flow based formulations.
## Unprotected case

**WP, FF**

<table>
<thead>
<tr>
<th>Solenoidality</th>
<th>[ \sum_{k \in A_l} x_{k,l,c,\lambda} - \sum_{l \in A_l} x_{l,k,c,\lambda} = \begin{cases} v_{c,\lambda} &amp; \text{if } l = d_c \ -v_{c,\lambda} &amp; \text{if } l = s_c \ 0 &amp; \text{otherwise} \end{cases} ] [ \forall l, c, \lambda ]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[ \sum_{\lambda} v_{c,\lambda} = v_c ] [ \forall c ]</td>
</tr>
</tbody>
</table>

| Capacity     | \[ \sum_{c} x_{l,k,c,\lambda} \leq F_{l,k} \] \[ \forall (l, k), \lambda \] |

<table>
<thead>
<tr>
<th>Integrity</th>
<th>[ x_{l,k,c,\lambda} \text{ integer} ] [ \forall c, (l, k), \lambda ]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[ F_{l,k} \text{ integer} ] [ \forall (l, k) ]</td>
</tr>
<tr>
<td></td>
<td>[ v_{c,\lambda} \text{ integer} ] [ \forall c, \lambda ]</td>
</tr>
</tbody>
</table>
Flow (FF) vs Route (RF) Formulation

- Variable $x_{i,s,d}$: flow on link $i$ associated to source-destination couple $s$-$d$

- Fixed number of variables
- Unconstrained routing

Variable $r_{p,s,d}$: number of connections $s,d$ routed on the admissible path $p$

- Constrained routing
  - $k$-shortest path
- Sub-optimality?
All the possible paths between each sd-pair are evaluated \textit{a priori}

Variables represent which path is used for a given connection
- \( r_{p_{sd}} \): path \( p \) is used by \( r_{psd} \) connections between \( s \) and \( d \)

Path-related constraints
- Routing can be easily constrained (e.g. using the \( K \)-shortest paths)
  - Yen’s algorithm
- Useful to represent path-interference
  - Physical topology represented in terms of interference (crossing) between paths (e.g. \( i_{pr} = 1 \) (0) if path \( p \) has a link in common with path \( r \))

Number of variables and constraints
- Very large in the unconstrained case,
- Simpler than flow formulation when routing is constrained
New symbols

- \( r_{c,n} \): number of connections routed on the \( n \)-th admissible path between source destination nodes of the node-couple \( c \)
- \( R_{(l,k)} \): set of all admissible paths passing through link \((l,k)\)

<table>
<thead>
<tr>
<th>Solenoidality</th>
<th>( \sum_{n} r_{c,n} = v_c \quad \forall c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>( \sum_{r_{c,n} \in R_{l,k}} r_{c,n} \leq W \cdot F_{l,k} \quad \forall (l,k) )</td>
</tr>
<tr>
<td>Integrity</td>
<td>( r_{c,n} ) integer ( \forall (c,n) )</td>
</tr>
<tr>
<td></td>
<td>( F_{l,k} ) integer ( \forall (l,k) )</td>
</tr>
</tbody>
</table>
Unprotected case

**WP, RF**

- Analogous extension to FF case
  
  \( r_{c,n,\lambda} \) = number of connection routed on the \( n\)-th admissible path between node pair \( c \) (source-destination) on wavelength \( \lambda \)

<table>
<thead>
<tr>
<th>Solenoidality</th>
<th>( \sum \sum r_{c,n,\lambda} = v_{c,\lambda} ) ( \forall c, \lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \sum v_{c,\lambda} = v_c ) ( \forall c )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Capacity</th>
<th>( \sum \sum r_{c,n,\lambda} \leq F_{l,k} ) ( \forall (l, k), \lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( r_{c,n,\lambda} ) integer ( \forall (c, n), \lambda )</td>
</tr>
</tbody>
</table>

| Integrity              | \( F_{l,k} \) integer \( \forall (l, k) \)                               |
|                        | \( v_{c,\lambda} \) integer \( \forall c, \lambda \)                    |
ILP source formulation
New formulation derived from flow formulation

- Reduced number of variables and constraints compared to the flow formulation
- Allows to evaluate the absolute optimal solution without any approximation and with unconstrained routing
- Can not be employed in case path protection is adopted as WDM protection technique
  - Does not support link-disjoint routing
ILP source formulation

Source Formulation (SF) fundamental constraints

- New flow variable \( x_{l,k,s} = \sum_d x_{l,k}^{s,d} \)
  - Flow carried by link \( l \) and having node \( s \) as source
  - Flow variables do not depend on destinations anymore

- Solenodality
  - Source node
    - the sum of \( x_{l,k,s} \) variables is equal to the total number of requests originating in the node
  - Transit node
    - the incoming traffic has to be equal to the outgoing traffic plus the nr. of lightpaths terminated in the node
ILP source formulation
An example

- 2 connections requests
  - 1 to 4
  - 1 to 3

- Solution
  - \( X_{1,2,1}, X_{1,6,1}, X_{2,3,1}, X_{6,5,1}, X_{5,3,1}, X_{3,41} = 1 \)
  - Otherwise \( X_{i,l,k,1} = 0 \)

- 1° admissible solution
  - \( L_A \) 1-2-3-4
  - \( L_B \) 1-6-5-3

- 2° admissible solution
  - \( L_A \) 1-2-3
  - \( L_B \) 1-6-5-3-4

Both routing solutions are compatible with the same SF solution.
Unprotected case
VWP, SF

- New symbols
  - $x_{l,k,i}$: number of WDM channels carried by link $l$, $k$ assigned connections originating at node $i$
  - $C_{i,j}$: number of connection requests from node $i$ to node $j$

- See [ToMaPa02]

<table>
<thead>
<tr>
<th>Solenoidality</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sum_{k \in A_i} x_{i,k,i} = \sum_{j} C_{i,j} = S_i \quad \forall i$</td>
</tr>
<tr>
<td>$\sum_{k \in A_i} x_{k,l,i} = \sum_{k \in A_i} x_{l,k,i} + C_{i,l} \quad \forall i, l (i \neq l)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sum_{i} x_{l,k,i} \leq W \cdot F_{l,k} \quad \forall (l, k)$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Integrity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{l,k,i}$ integer $\quad \forall i, (l,k)$</td>
</tr>
<tr>
<td>$F_{l,k}$ integer $\quad \forall (l,k)$</td>
</tr>
</tbody>
</table>
Unprotected case

**WP, SF**

- Analogous extension to FF case

<table>
<thead>
<tr>
<th>Solenoidality</th>
<th>( \sum_{k \in A_i} x_{i,k,i,\lambda} = s_{i,\lambda} \quad \forall i, \lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \sum_{\lambda} s_{i,\lambda} = S_i = \sum_{i,l} C_{i,l} \quad \forall i )</td>
</tr>
<tr>
<td></td>
<td>( \sum_{k \in A_i} x_{k,l,i,\lambda} = \sum_{k \in A_i} x_{l,k,i,\lambda} + c_{i,j,\lambda} \quad \forall i, l, \lambda, (i \neq l) )</td>
</tr>
<tr>
<td></td>
<td>( \sum_{\lambda} c_{i,l,\lambda} = C_{i,l} \quad \forall i, l, (i \neq l) )</td>
</tr>
</tbody>
</table>

| Capacity     | \( \sum_{i} x_{l,k,i,\lambda} \leq F_{l,k} \quad \forall (l, k, \lambda) \) |

<table>
<thead>
<tr>
<th>Integrity</th>
<th>( x_{l,k,i,\lambda} ) integer ( \forall l,k,i,\lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( F_{l,k} ) integer ( \forall l,k )</td>
</tr>
<tr>
<td></td>
<td>( s_{i,\lambda} ) integer ( \forall i, \lambda )</td>
</tr>
<tr>
<td></td>
<td>( c_{i,l,\lambda} ) integer ( \forall i, j, \lambda, (i \neq l) )</td>
</tr>
</tbody>
</table>

Analogous extension to FF case
**ILP source formulation**

*Two-step solution of the optimization problem*

- The source formulation variables $x_{l,k,s}$ and $F_{l,k}$
  - Do not give a detailed description of RWFA of each single lightpath
  - Describe each tree connecting a source to all the connected destination nodes (a subset of the other nodes)
  - Define the optimal capacity assignment (dimensioning of each link in terms of fibers per link) to support the given traffic matrix

**STEP 1:** Optimal dimensioning computation by exploiting SF (identification problem, high computational complexity)

**STEP 2:** RFWA computation after having assigned the number of fibers of each link evaluated in step 1 (multicommodity flow problem, negligible computational complexity)
ILP source formulation

**Complexity comparison between SF and FF**

<table>
<thead>
<tr>
<th>Formulation</th>
<th># const.</th>
<th># variab.</th>
</tr>
</thead>
<tbody>
<tr>
<td>VWP source</td>
<td>$2L + S \cdot N$</td>
<td>$2L(1 + S)$</td>
</tr>
<tr>
<td>VWP flow</td>
<td>$2L + C \cdot N$</td>
<td>$2L(1 + C)$</td>
</tr>
<tr>
<td>VWP route</td>
<td>$2L + C$</td>
<td>$C \cdot R + 2L$</td>
</tr>
<tr>
<td>WP source</td>
<td>$W(2L + NS) + C + S$</td>
<td>$W(S + 2L \cdot S + C) + 2L$</td>
</tr>
<tr>
<td>WP flow</td>
<td>$W(2L + NC) + C$</td>
<td>$W \cdot C(1 + 2L) + 2L$</td>
</tr>
<tr>
<td>WP route</td>
<td>$C(W + 1) + 2L \cdot W$</td>
<td>$C \cdot R \cdot W + C \cdot W + 2L$</td>
</tr>
</tbody>
</table>

### Symbols
- $N$ nodes
- $L$ links
- $R$ average number of paths per node pair
- $W$ wavelengths per fiber
- $C$ connection node-pairs
- $S$ source nodes requiring connectivity

### The second step has a negligible impact on the SF computational time
- $F_{l,k}$ are no longer variables but known terms

### Fully-connected virtual topology: $C = N \cdot (N-1)$, $S=N$
- Worst case (assuming $L \ll N \cdot (N-1)$)
Case-study networks

*Network topology and parameters*

**NSFNET**
- $N = 14$ nodes
- $L = 22$ (bidir)links
- $C = 108$ connected pairs
- 360 (unidir)conn. requests

**EON**
- $N = 19$ nodes
- $L = 39$ (bidir)links
- $C = 342$ connected pairs
- 1380 (unidir)conn. requests

- Static traffic matrices derived from real traffic measurements
- Hardware: 1 GHz processor, 460 Mbyte RAM
- Software: CPLEX 6.5
Case-study networks
SF vs. FF: variables and constraints (VWP)

- The number of constraints decreases by a factor
  - 9 for the NSFNet
  - 26 for the EON
- The number of variables decreases by a factor
  - 8.5 for the NSFNet
  - 34 for the EON
- These simplifications affect computation time and memory occupation, achieving relevant savings of computational resources

<table>
<thead>
<tr>
<th>Network/Formulation</th>
<th># const.</th>
<th># variab.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NsfNet/source</td>
<td>284</td>
<td>570</td>
</tr>
<tr>
<td>NsfNet/flow</td>
<td>2552</td>
<td>4840</td>
</tr>
<tr>
<td>EON/source</td>
<td>517</td>
<td>1560</td>
</tr>
<tr>
<td>EON/flow</td>
<td>13650</td>
<td>53430</td>
</tr>
</tbody>
</table>
Case-study networks

*SF vs. FF: variables and constraints (WP)*

- In the WP case
  - The number of variables and constraints linearly increases with W
  - The gaps in the number of variables and constraints between FF and SF increase with W
- The advantage of source formulation is even more relevant in the WP case

![Graph showing the number of variables and constraints vs. number of wavelengths, W]
Case-study networks
SF vs. FF: time, memory and convergence

- The values of the cost function $M_{\text{source}}$ obtained by SF are always equal or better (lower) than the corresponding FF results ($M_{\text{flow}}$).
- Coincident values are obtained if both the formulations converge to the optimal solution
  - Validation of SF by induction
- Memory exhaustion (Out-Of-Memory, O.O.M) prevents the convergence to the optimal solution. This event happens more frequently with FF than with SF.

<table>
<thead>
<tr>
<th>Memory occupation</th>
<th>Computational time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>W</strong></td>
<td><strong>SF</strong></td>
</tr>
<tr>
<td>2</td>
<td>0.39MB</td>
</tr>
<tr>
<td>4</td>
<td>O.O.M</td>
</tr>
<tr>
<td>8</td>
<td>5MB</td>
</tr>
<tr>
<td>16</td>
<td>47MB</td>
</tr>
<tr>
<td>32</td>
<td>180MB</td>
</tr>
</tbody>
</table>
Outline

- Introduction to WDM network design and optimization
- Integer Linear Programming approach
- Physical Topology Design
  - Unprotected case
  - Dedicated path protection case
  - Shared path & link protection cases
- Heuristic approach
Protection in WDM Networks (1)
Motivations (bit rate)

- Today WDM transmission systems allow the multiplexing on a single fiber of up to 160 distinct optical channels
  - recent experimental systems support up to 256 channels:
- A single WDM channel carries from 2.5 to 40 Gb/s (ITU-T G.709)
- The loss of a high-speed connection operating at such bit rates, even for few seconds, means huge waste of data!!
- The increase in WDM capacity associated with the tremendous bandwidth carried by each fiber and the evolution from ring to mesh architectures brought the need for suitable protection strategies into foreground.
- Example: 1ms outage for a 100G x 100Waves fiber [10Tbit/s] means 10Gbit=1.25Gbyte of data lost (n.b.: 1 cd-rom is 0.9 GByte)
Even though fiber are very resilient, the geographical dimension of a backbone network lead to very high chances that the network is operating in a fault state.

Example: failure per 1000Km per year (2001 statistics) ≈ 2 (*)

What happens on a continental network?! Hundreds of failures….
Immediate Causes of Fiber Cable Failures

Customer’s concerns:

- Bandwidth
- Availability
- Fee
- etc.

Operator’s concerns:

- Resource
- Protection
- Penalty

Network design decisions for protection are very important.
3 performance metrics for protection

- Resource occupation (resource overbuild)

- Availability
  - Probability to find the service up

- Protection Switching Time

- Availability goes in tradeoff with the other two!!
Dedicated Path Protection (DPP)

- 1+1 or 1:1 dedicated protection (>50% capacity for protection)
  - Both solutions are possible
  - Each connection-request is satisfied by setting-up a lightpath pair of a working + a protection lightpaths
  - RFWA must be performed in such a way that working and protection lightpaths are link disjoint

  ⇒ Additional constraints must be considered in network planning and optimization

- Transit OXCs must not be reconfigured in case of failure
- The source model can not be applied to this scenario
“Max half” formulation (MH)

Equation set (VWP case)

- This formulation does not need an upgrade of unprotected flow variable
- See [ToMaPa04]

<table>
<thead>
<tr>
<th>Solenoidality</th>
<th>$\sum_{(l,k) \in I_i^+} x_{l,k,c} - \sum_{(l,k) \in I_i^-} x_{l,k,c} = \begin{cases} 2 \cdot v_c &amp; \text{if } i = d_c \ -2 \cdot v_c &amp; \text{if } i = d_c \ 0 &amp; \text{otherwise} \end{cases} \quad \forall (i,c)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Half</td>
<td>$x_{l,k,c} + x_{k,l,c} \leq v_c \quad \forall l,c$</td>
</tr>
<tr>
<td>Capacity</td>
<td>$\sum_c x_{l,k,c} \leq W \cdot F_{l,k} \quad \forall (l,k)$</td>
</tr>
<tr>
<td>Integrity</td>
<td>$x_{l,k,c} \text{ integer} \quad \forall c,(l,k)$</td>
</tr>
<tr>
<td></td>
<td>$F_{l,k} \text{ integer} \quad \forall (l,k)$</td>
</tr>
</tbody>
</table>
"Max half" formulation (MH)

Limitations

😊 Same number of variables and constraints as the unprotected flow formulation

😊 Allows to evaluate the absolute optimal solution without any approximation and with unconstrained routing in almost all cases

😊 Requires an *a posteriori control* to verify the feasibility of obtained solution

😊 Problem:

In conclusion, each *unity of flow* must be modelled *independently*, such that it can be protected *independently*.

Fig. 1. This routing assignment satisfies the necessary condition enforced by “Max Half” formulation, but it does not satisfies link-disjointness.
New symbols

- \( x_{l,k,c,t} \) = number of WDM channels carried by link \((l,k)\) assigned to the \(t\)-th connection between source-destination couple \(c\)

**Rationale:** for each connection request, route a link-disjoint connection \(\implies\) route two connections and enforce link-disjointness between them.

| Solenoidality | \( \sum_{k \in A_l} x_{k,l,c,t} - \sum_{k \in A_l} x_{l,k,c,t} \) = \[
\begin{cases}
2 & \text{if } l = d_c \\
-2 & \text{if } l = s_c \\
0 & \text{otherwise}
\end{cases}
\ \forall l, c, t |
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Link-disjoint</td>
</tr>
<tr>
<td>Capacity</td>
</tr>
</tbody>
</table>
| Integrity | \( x_{l,k,c,t} \) binary \ \forall (c,t),(l,k) \\
| | \( F_{l,k} \) integer \ \forall (l,k) |
## New symbols

- \( r_{c,n,t} = 1 \) if the \( t\)th connection between source destination node couple \( c \) is routed on the \( n\)th admissible path
- \( R_{(l,k)} \) = set of all admissible paths passing through link \( (l,k) \)

### Solenoidality

\[
\sum_{n} r_{c,t,n} = 2 \quad \forall \ c,t
\]

### Link - disjoint

\[
\sum_{r_{c,t,n} \in R_{l,k} \cup R_{k,l}} r_{c,t,n} \leq 1 \quad \forall \ c,t,l,k
\]

### Capacity

\[
\sum_{r_{c,t,n} \in R_{l,k}} r_{c,t,n} \leq W \cdot F_{l,k} \quad \forall l,k
\]

### Integrity

\( r_{c,t,n} \) binary \( \forall c,t,n \)

\( F_{l,k} \) integer \( \forall l,k \)
### Dedicated Path Protection (DPP)

*Route formulation (RF) II, VWP*

- Substitute the single path variable \( r_{c,n} \) by a protected route variable \( r'_{c,n} \) (\(~ a \) cycle)
- No need to explicitly enforce link disjointness
- Identical formulation to unprotected case
- How do we calculate the minimum-cost disjoint paths? Suurballe

<table>
<thead>
<tr>
<th>Solenoidal ity</th>
<th>( \sum_{n} r'_{c,n} = v_c \quad \forall c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>( \sum_{r'<em>{c,n} \in R'</em>{l,k}} r'<em>{c,n} \leq W \cdot F</em>{l,k} \quad \forall l, k )</td>
</tr>
<tr>
<td>Integrity</td>
<td>( r'_{c,n} ) integer \quad \forall c, n</td>
</tr>
<tr>
<td></td>
<td>( F_{l,k} ) integer \quad \forall l, k</td>
</tr>
</tbody>
</table>
Dedicated Path Protection (DPP)

**Flow formulation, WP**

- **New symbols**
  - \( x_{l,k,c,t,\lambda} \) = number of WDM channels carried by wavelength \( \lambda \) on link \( l,k \) assigned to the \( t \)-th connection between source-destination couple \( c \)
  - \( V_{c,\lambda} \) = traffic of connection \( c \) along wavelength \( \lambda \)

| Solenoidality | \[
\sum_{k \in A_l} x_{k,l,c,t,\lambda} - \sum_{k \in A_l} x_{l,k,c,t,\lambda} = \begin{cases} 
V_{c,t,\lambda} & \text{if } l = d_c \\
-V_{c,t,\lambda} & \text{if } l = s_c \\
0 & \text{otherwise} 
\end{cases} \quad \forall l, c, t, \lambda; \\
\sum_{\lambda} V_{c,t,\lambda} = 2 \quad \forall c
\] |
|---|---|
| Link - disjoint | \[
\sum_{\lambda} x_{l,k,c,t,\lambda} + x_{k,l,c,t,\lambda} \leq 1 \quad \forall l, k, c, t
\] |
| Capacity | \[
\sum_{(c,t)} x_{l,k,c,t,\lambda} \leq F_{l,k} \quad \forall l, k, \lambda
\] |
| Integrity | \[
x_{l,k,c,t,\lambda} \text{ binary} \quad \forall c, t, l, k, \lambda \\
F_{l,k} \text{ integer} \quad \forall l, k \\
V_{c,t,\lambda} \text{ binary} \quad \forall c, \lambda
\] |
## Dedicated Path Protection (DPP)

**Route formulation (RF), WP**

### Solenoidality

\[
\sum_{c,t} r_{c,t,n,\lambda} = v_{c,t,\lambda} \quad \forall (c,t), \lambda \\
\sum_{\lambda} v_{c,t,\lambda} = 2 \quad \forall c,t
\]

### Link - disjoint

\[
\sum_{\lambda} \sum_{r_{c,t,n,\lambda} \in R_l,k \cup R_k,l} r_{c,t,n,\lambda} \leq 1 \quad \forall (c,t),(l,k)
\]

### Capacity

\[
\sum_{r_{c,t,n,\lambda} \in R_l,k} r_{c,t,n,\lambda} \leq F_{l,k} \quad \forall (l,k), \lambda
\]

### Integrity

- \( r_{c,t,n,\lambda} \) binary \( \forall (c,t), n, \lambda \)
- \( F_{l,k} \) integer \( \forall l,k \)
- \( v_{c,t,\lambda} \) bynary \( \forall (c,t), \lambda \)
**Dedicated Path Protection (DPP)**

*Route formulation (RF) II, WP*

- \( r_{c,n,\lambda} \) = number of connections between source-destination couple \( c \) routed on the \( n\)-th admissible couple of disjoint paths having one path over wavelength \( \lambda_1 \) and the other over \( \lambda_2 \)

<table>
<thead>
<tr>
<th>Solen.</th>
<th>[ \sum_{n} r'_{c,n,\lambda_1,\lambda_2} = v_c, \lambda_1, \lambda_2 \quad \forall c, \lambda_1, \lambda_2 ]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[ \sum_{\lambda_1, \lambda_2} v_c, \lambda_1, \lambda_2 = v_c \quad \forall c ]</td>
</tr>
<tr>
<td>Cap.</td>
<td>[ \sum_{\lambda_2} \sum_{r'<em>{c,n,\lambda_1,\lambda_2}} + \sum</em>{\lambda_2} \sum_{r'<em>{c,n,\lambda_2,\lambda_1}} \leq F</em>{l,k} \quad \forall l, k, \lambda_1 ]</td>
</tr>
<tr>
<td>Int.</td>
<td>[ r'_{c,n,\lambda_1,\lambda_2} \text{ integer} \quad \forall c, n, \lambda_1, \lambda_2 ]</td>
</tr>
<tr>
<td></td>
<td>[ F_{l,k} \text{ integer} \quad \forall l, k ]</td>
</tr>
</tbody>
</table>
Complexity comparison between DPP RF-WP formulations

- Nr of variables for $r_{ctn\lambda} \rightarrow R \times C \times T \times W$
  - $R \times C$ number of single route variables

- Nr of variables for $r_{cn\lambda'} \rightarrow R' \times C \times W^2$
  - $R' \times C$ number of protected route variables
  - For $R=R'$, this is preferable if $T>W$
Outline

- Introduction to WDM network design and optimization
- Integer Linear Programming approach
- Physical Topology Design
  - Unprotected case
  - Dedicated path protection case
  - Shared path & link protection cases
- References
- Heuristic approach
Shared Path Protection (SPP)

- Sharing is a way to decrease the capacity redundancy and the number of lightpaths that must be managed.

- Protection-resources sharing
  - Protection lightpaths of different channels share some wavelength channels
  - Based on the assumption of single point of failure
  - Working lightpaths must be link (node) disjoint

- Very complex control issues
  - Also transit OXCs must be reconfigured in case of failure
    - Signaling involves also transit OXCs
    - Lightpath identification and tracing becomes fundamental
How to model SPP: the Link Vector (1)

- **Given:**
  - Graph $G(V,E)$
  - Set of connections $L^i$ (already routed)

- **Link vector**

  $$\nu_e = \{ \nu^e_{e'} | \forall e' \in E, 0 \leq \nu^e_{e'} \leq \lambda(e') \}$$

  $$\nu^*_e = \max_{\forall e'} \nu^e_{e'}$$

- Specified in IETF (see, e.g., RSVP-TE Extensions For Shared-Mesh Restoration in Transport Networks)
How to model SPP: the Link Vector (2)

(a) Sample network and connections

<table>
<thead>
<tr>
<th>Initial state : $e_5$</th>
<th>$e_1$</th>
<th>$e_2$</th>
<th>$e_3$</th>
<th>$e_4$</th>
<th>$e_5$</th>
<th>$e_6$</th>
<th>$e_7$</th>
<th>$e_8$</th>
<th>$V_{e_5}^*$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 0, 0, 0, 0, 0, 0, 0)</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection A arrival: $e_5$</td>
<td>(0, 1, 1, 0, 0, 0, 0, 0)</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection B arrival: $e_5$</td>
<td>(1, 2, 1, 0, 0, 0, 0, 0)</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection C arrival: $e_5$</td>
<td>(1, 2, 1, 0, 0, 1, 0, 0)</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) Evolution of link vector for $e_5$
Two ILP approaches for SPP

- Explore Shared path protection by both classical approaches

- Flow Formulation
  - Binary variables $x_{l,k,c,t,p}$ associated to the flow on each link $l,k$ for each single connection request $c,t$ ($p=w$ working flow, $p=s$ protection flow)

- Route Formulation
  - 1 approach: integer variables $r_{c,n}$ associated to each simple path $n$ joining the node pair $c$ (e.g. $s,d$).
  - 2 approach: integer variables $r'_{c,n}$ associated to the $n$-th possible working-spare route pair that joins each $s-d$ node pair $c$
### How to calculate «max» in ILP

#### Objective

\[
\min \left( \max_{(l,k)} \sum_c x_{l,k,c} \right) = \min T
\]

#### Solenoidality

\[
\sum_{k \in A_l} x_{k,l,c} - \sum_{k \in A_l} x_{l,k,c} = \begin{cases} 
  v_c & \text{if } l = d_c \\
  -v_c & \text{if } l = s_c \\
  0 & \text{otherwise}
\end{cases} \quad \forall l,c
\]

#### Max

\[
T \geq \sum_c x_{l,k,c} \quad \forall (l,k)
\]

#### Capacity

\[
\sum_c x_{l,k,c} \leq W \cdot F_{l,k} \quad \forall (l,k)
\]

#### Integrity

- \( x_{l,k,c} \) integer \quad \forall c,(l,k)
- \( F_{l,k} \) integer \quad \forall (l,k)
## Shared Path Protection (DPP)

**Flow formulation, VWP**

| Solenoidality | \[ \sum_{k \in A_l} x_{k,l,c,t} - \sum_{k \in A_l} x_{l,k,c,t} = \begin{cases} 1 & \text{if } l = d_c \\ -1 & \text{if } l = s_c \\ 0 & \text{otherwise} \end{cases} \forall l, c, t; \]
| \[ \sum_{k \in A_l} y_{k,l,c,t} - \sum_{k \in A_l} y_{l,k,c,t} = \begin{cases} 1 & \text{if } l = d_c \\ -1 & \text{if } l = s_c \\ 0 & \text{otherwise} \end{cases} \forall l, c, t; \] |

| Link - disjoint | \[ x_{l,k,c,t} + x_{k,l,c,t} + y_{k,l,c,t} + y_{k,l,c,t} \leq 1 \forall l, k, c, t \] |

| Capacity | \[ P_{lk} + \sum_{(c,t)} x_{l,k,c,t} \leq W \cdot F_{l,k} \forall l, k \]
| \[ P_{lk} \geq \sum_{(c,t)} z_{lk,ct}^{ij} \forall (l,k), (i, j) \] |

| Sharing | \[ z_{lk,ct}^{ij} \geq x_{i,j,c,t} + y_{l,k,c,t} - 1 \forall (c,t),(l,k),(i, j) \]
| \[ z_{lk,ct}^{ij} \leq x_{i,j,c,t}, \quad z_{lk,c}^{ij} \leq y_{i,j,c,t} \] |

| Integrity | \[ x_{l,k,c,t} \text{ binary } \forall (c,t),(l,k) \]
| \[ F_{l,k} \text{ integer } \forall (l,k) \]
| \[ z_{lk,ct}^{ij} \text{ binary } \forall (c,t),(l,k),(i, j) \] |
Shared Path Protection case  
**VWP, RF**

- **New symbols**
  - $R_{l,k}$ includes all the working-spare routes whose working path is routed on link $l,k$
  - $R_{l'\,(l,k)}$ includes all the working-spare routes whose working path is routed on bidirectional link $l'$ and whose spare path is routed on link $(l, k)$

<table>
<thead>
<tr>
<th>Solenoidality</th>
<th>$\sum_{n} r'<em>{c,n} \leq v</em>{c} \quad \forall \ c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>$\sum_{(c,n)\in R_{l,k}} r'<em>{c,n} + \sum</em>{(c,n)\in R_{l',(l,k)}} r'<em>{c,n} \leq W \cdot F</em>{l,k}$</td>
</tr>
<tr>
<td></td>
<td>$\forall (l,k), l' (l \neq l)$</td>
</tr>
<tr>
<td>Integrity</td>
<td>$r'_{c,n}$ integer $\quad \forall (c,n)$</td>
</tr>
<tr>
<td></td>
<td>$F_{l,k}$ integer $\quad \forall (l,k)$</td>
</tr>
</tbody>
</table>

- Similar formulations can be found in [MiSa99], [RaMu99], [BaBaGiKo99],

---

WDM Network Design
Shared Path Protection case

**WP, RF**

- New symbols
  - Variable $r_{c,n,\lambda_1,\lambda_2}$, where $\lambda_1$ indicates the wavelength of the working path and $\lambda_2$ indicates the wavelength of the spare path.
  - $R^{(l,k,\lambda_2)}_l$ includes all the working-spare routes, whose working path is routed on bidirectional link $l'$ and whose spare path is routed on link $(l, k)$.

<table>
<thead>
<tr>
<th>Solenoidal ity</th>
<th>$\sum_{n} r'_{c,n,\lambda_1,\lambda_2} \leq v_c \quad \forall \ c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>$\sum_{(c,n,\lambda_2)\in R^{(l,k,\lambda_2)}<em>l} r'</em>{c,n,\lambda_1,\lambda_2} + \sum_{(c,n)\in R^{(l,k,\lambda_2)}<em>l} r'</em>{c,n,\lambda_1,\lambda_2} \leq W \cdot F_{l,k}$</td>
</tr>
<tr>
<td></td>
<td>$\forall (l,k), l', \lambda_1 (l \neq l)$</td>
</tr>
<tr>
<td>Integrity</td>
<td>$r'_{c,n,\lambda_1,\lambda_2}$ integer $\quad \forall (c,n,\lambda_1,\lambda_2)$</td>
</tr>
<tr>
<td></td>
<td>$F_{l,k}$ integer $\quad \forall (l,k)$</td>
</tr>
</tbody>
</table>

- All the previous formulations and a additional one can be found in [CoToMaPaMa03].
Link Protection

- Dedicated Link Protection (DLP)
  - Each link is protected by providing an alternative routing for all the WDM channels in all the fibers
  - Protection switching can be performed by *fiber switches* (fiber cross-connects) or *wavelength switches*
  - Signaling is local; transit OXCs of the protection route can be pre-configured
  - Fast reaction to faults
  - Some network fibers are reserved for protection

- Shared Link Protection (SLP)
  - Protection fibers may be used for protection of more than one link (assuming single-point of failure)
  - The capacity reserved for protection is greatly reduced
**Link Protection**

- Different protected objects are switched
  1) Fiber level
  2) Wavelength level

*FAULT EVENT*
## Link Protection

**VWP, FF (Fiber Protection Switch)**

- **New symbols**
  - \( Y_{(l,k),(L,q)} \) expresses the number of backup fibers needed on link \((l,k)\) to protect link \((L,q)\) failure

<table>
<thead>
<tr>
<th>Solenoidality</th>
<th>[ \sum_{k \in A_l} x_{l,k,c} - \sum_{k \in A_l} x_{k,l,c} = \begin{cases} v_c &amp; \text{if } i = d_c \ -v_c &amp; \text{if } i = s_c \ 0 &amp; \text{otherwise} \end{cases} ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>(working)</td>
<td>[ \sum_{k \in A_l(l,k \neq L,q)} Y_{l,k,(L,q)} - \sum_{k \in A_l(l,k \neq L,q)} Y_{k,l,(L,q)} = \begin{cases} F_{L,q} &amp; \text{if } l = L \ -F_{L,q} &amp; \text{if } l = q \ 0 &amp; \text{otherwise} \end{cases} ]</td>
</tr>
<tr>
<td>(spare)</td>
<td>[ \sum_{c} x_{l,k,c} \leq W \cdot F_{l,k} \quad \forall (l,k) ]</td>
</tr>
<tr>
<td>Capacity</td>
<td>[ Y_{l,k,(L,q)} \quad \forall (l,k), (L,q) ]</td>
</tr>
<tr>
<td>Integrity</td>
<td>[ x_{l,k,c} \text{ integer} \quad \forall c, (l,k) ]</td>
</tr>
<tr>
<td></td>
<td>[ F_{l,k} \text{ integer} \quad \forall (l,k) ]</td>
</tr>
</tbody>
</table>
Link Protection

Cost functions and sharing constraints

- Cost function (dedicated case)
  \[
  \min \sum_{l,k} F_{l,k} + \sum_{l,k} \sum_{L,q} Y_{l,k,(L,q)}
  \]

- Cost function (shared case)
  \[
  \min \sum_{l,k} F_{l,k} + \sum_{l,k} T_{l,k}
  \]
  s.t. \[ T_{l,k} \geq Y_{l,k,(L,q)} \quad \forall (l,k), (L,q) \]

- OSS. Wavelength channel level protection design
  - Relaxing the integer constraints on \( Y \), each channel is independently protected (while not collecting all the channels owing to the same fiber)

- See also [RaMu99]
Link Protection

Summary results

- Comparison between different protection technique on fiber needed to support the same amount of traffic

- Switching protection objects at fiber or wavelength level does not sensibly affects the amount of fibers.
  - This difference increase with the number of wavelength per fiber
Traffic Grooming
Definition

- Optical WDM network
  - multiprotocol transport platform
  - provides connectivity in the form of optical circuits (lightpaths)

Electronic layers

<table>
<thead>
<tr>
<th>SDH</th>
<th>ATM</th>
<th>IP</th>
</tr>
</thead>
</table>

Optical layers

- WDM Layer
- Optical transmission

A Big Difference between Electronic Traffic Requests and Optical Lightpath Capacity!
Traffic Grooming
Multi-layer routing

Logical Topology

Suppose Lightpath Capacity: 10 Gbit/sec
EXAMPLE CONNECTIONS ROUTING
C1 (STM1 between 4 → 2) on Lp1
C2 (STM1 between 4 → 7) on Lp1 and Lp2
C3 (STM1 between 1 → 5) on Lp3
C4 (STM1 between 4 → 6) on Lp1 and Lp2 and Lp4
Given:

- \( N \): number of nodes in the network.
- \( W \): number of wavelengths per fiber. We assume all of the fibers in the network carry the same number of wavelengths.
- \( P_{mn} \): number of fibers interconnecting node \( m \) and node \( n \).
- \( P_{mn} = 0 \) for node pair which is not physically adjacent to each other. \( P_{mn} = P_{nm} = 1 \) if and only if there exists a direct physical fiber link between nodes \( m \) and \( n \).
- \( P^w_{mn} \): wavelength \( w \) on fiber \( P_{mn} \). \( P^w_{mn} = P_{mn} \).
- \( TR_i \): number of transmitters at node \( i \).
- \( RR_i \): number of receivers at node \( i \). Note that, in this set of ILP formulation, we assume all the nodes are equipped with tunable transceivers, which can be tuned to any of \( W \) wavelengths.

- \( C_i \): capacity of each channel (wavelength).
- \( \Lambda \): traffic matrix set. \( \Lambda = \{\Lambda_y\} \), where \( y \) can be any allowed low-speed streams, \( 1, 3, 12, \) etc. In our study, \( y \in \{1, 3, 12, 48\} \). \( \Lambda_{y,sd} \) is the number of OC-\( y \) connection requests between node pair \((s, d)\).

Variables:

- **Virtual topology:**
  - \( V_{ij} \): number of lightpaths from node \( i \) to node \( j \) in virtual topology. \( V_{ij} = 0 \) does not imply that \( V_{ji} = 0 \).
  - \( V_{ij}^w \): number of lightpaths from node \( i \) to node \( j \) on wavelength \( w \). Note that, if \( V_{ij}^w > 1 \), the lightpaths between node \( i \) and \( j \) on wavelength \( w \) may take different paths.

- **Physical topology route:**
  - \( P^w_{ij} \): number of lightpaths between nodes \((i, j)\) routed through fiber link \((m, n)\) on wavelength \( w \).

- **Traffic route:**
  - \( \lambda_{i,j,y}^{s,d} \): The \( t \)th OC-\( y \) low-speed traffic request from node \( s \) to node \( d \) employing lightpath \((i, j)\) as an intermediate virtual link.
  - \( S_{sd}^{y,t} \): \( S_{sd}^{y,t} = 1 \) if the \( t \)th OC-\( y \) low-speed connection request from node \( s \) to node \( d \) has been successfully routed; otherwise, \( S_{sd}^{y,t} = 0 \).
Formulation (Keyao Zhu JSAC03)

- Optimize: Maximize the total successfully-routed low-speed traffic.

\[
\text{Maximize: } \sum_{u,s,d,t} y \cdot S_{ud}^{y,t}. \tag{1}
\]

- Constraints:
  - On virtual-topology connection variables
    \[
    \sum_j V_{ij} \leq TR_i \quad \forall i \tag{2}
    \]
    \[
    \sum_i V_{ij} \leq RR_j \quad \forall j \tag{3}
    \]
    \[
    \sum_w V_{ij} = V_{ij} \quad \forall i, j \tag{4}
    \]
  - On physical route variables
    \[
    \sum_m P_{ij, w}^{m} = \sum_n P_{kj, w}^{n} \quad \text{if } k \neq i, j \quad \forall i, j, w, k \tag{6}
    \]
    \[
    \sum_m P_{ij, w}^{m} = 0 \quad \forall i, j, w \tag{7}
    \]
    \[
    \sum_n P_{kj, w}^{n} = 0 \quad \forall k, j, w \tag{8}
    \]
    \[
    \sum_n P_{ij, w}^{n} = V_{ij}^{w} \quad \forall i, j, w \tag{9}
    \]
    \[
    \sum_m P_{ij, w}^{m} = V_{ij}^{w} \quad \forall i, j, w \tag{10}
    \]
    \[
    \sum_m P_{ij, w}^{m} \leq P_{mn}^{w} \quad \forall m, n, w \tag{11}
    \]
    \[
    P_{ij, w}^{m} \in \{0, 1\}. \tag{12}
    \]
    - On virtual-topology traffic variables
      \[
      \sum_i \lambda_{i,j}^{s,t} = S_{s}^{y,t}
      \]
      \[
      \forall s, \ y \in \{1, 3, 12, 48\} \quad t \in [1, \Delta_y, sd] \tag{13}
      \]
      \[
      \sum_j \lambda_{s,t}^{d} = S_{s}^{y,t}
      \]
      \[
      \forall s, d, \ t \in [1, \Delta_y, sd] \tag{14}
      \]
      \[
      \sum_i \lambda_{i,j}^{s,t} = \sum_j \lambda_{k,j}^{s,t} \quad \text{if } k \neq s, d \quad \forall s, d, k, t \tag{15}
      \]
      \[
      \sum_i \lambda_{s,t}^{d} = 0
      \]
      \[
      \forall s, d \ y \in \{1, 3, 12, 48\} \quad t \in [1, \Delta_y, sd] \tag{16}
      \]
      \[
      \sum_j \lambda_{s,t}^{d} = 0
      \]
      \[
      \forall s, d \ y \in \{1, 3, 12, 48\} \quad t \in [1, \Delta_y, sd] \tag{17}
      \]
      \[
      \sum_{y,t} \sum_{s_d} y \cdot \lambda_{i,j}^{s,t} \leq V_{ij} \times C' \quad \forall i, j \tag{18}
      \]
      \[
      S_{s}^{y,t} \in \{0, 1\}. \tag{19}
      \]
Logical Operators: AND, OR, XOR, XNOR

- AND, OR, XOR etc can be expressed by using binary variables

<table>
<thead>
<tr>
<th>AND</th>
<th>1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 0</td>
</tr>
<tr>
<td>0</td>
<td>0 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OR</th>
<th>1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 1</td>
</tr>
<tr>
<td>0</td>
<td>1 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>XOR</th>
<th>1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 1</td>
</tr>
<tr>
<td>0</td>
<td>1 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>XNOR</th>
<th>1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 0</td>
</tr>
<tr>
<td>0</td>
<td>0 1</td>
</tr>
</tbody>
</table>

\[
z \leq x \quad z \geq x \quad r = x \ AND \ y
\]

\[
z \leq y \quad z \geq y \quad t = x \ OR \ y \quad z = 1 - (x \ XOR \ y)
\]

\[
z \geq x + y - 1 \quad z \leq x + y \quad z = t - r
\]

Some other linear operations

- If any of the $y$ is true, $x$ is true
- If $x \geq y$, $z$ is true
- If $x = y$, $C_y$ is true

\[ x \geq \frac{\sum_i y_i}{M} \]
\[ z \geq \frac{x - y + 1}{M}; \]
\[ A \geq \frac{x - y + 1}{M}; \]
\[ B \geq \frac{y - x + 1}{M}; \]
\[ C_y = A \land B; \]
Logical Operators: AND, OR, XOR, XNOR

- AND, OR, XOR etc can be expressed by using binary variables

<table>
<thead>
<tr>
<th>AND</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OR</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>XOR</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ y \leq x_1, x_2 \]
\[ y \geq x_1, x_2 \]
\[ y \geq x_1 + x_2 - 1 \]
\[ y \leq x_1 + x_2 \]

- Negated operators (NAND, NOR, XNOR) are implemented with an additional variable \( z = 1 - y \), where \( y \) is the “direct” operation

\[ y \leq x_1 + x_2 \]
\[ y \geq x_1 - x_2 \]
\[ y \geq x_2 - x_1 \]
\[ y \leq 2 - x_1 - x_2 \]
Logical Operators: AND, OR, XOR, XNOR

Also, they can be extended to a general $N$-variable case:

**AND**

\[ y \leq x_i \quad \forall i \in [1, N] \]

\[ y \geq \sum_{i=1}^{N} x_i - (N - 1) \]

**OR**

\[ \bigotimes_{i=1}^{N} y \geq \sum_{i=1}^{N} x_i \]

**XOR**

\[ y = x_1 \operatorname{xor} (x_2 \operatorname{xor} (x_3 \operatorname{xor} \ldots )) \]

Probably no simpler option exists.

Arbitrarily large number (at least equal to $N$)
A arduous challenge
  - NP-completeness/hardness coupled with a huge number of variables
    • In many cases the problem has a very high number of solutions (different virtual-topology mappings leading to the same cost-function value)
  - Practically tractable only for small networks

Simplifications
  - RFWA problem decomposition: e.g., first routing and then f/w assignment
  - Constrained routing (route formulation, see in the following)
  - Relaxed solutions (randomized rounding)
  - Other methodologies:
    • Column generation, Lagrangean relaxation, etc..

ILP, when solved with approximate methods, loses one of its main features: the possibility of finding a guaranteed minimum solution
  - Still ILP can provide valuable solutions
Simplification can be achieved by removing the integer constraint

- Connections are treated as fluid flows (multicommodity flow problem)
  - Can be interpreted as the limit case when the number of channels and connection requests increases indefinitely, while their granularity becomes indefinitely small
  - Fractional flows have no physical meaning in WDM networks as they would imply bifurcation of lightpaths on many paths
- LP solution is found

In some cases the closest upper integer to the LP cost function can be taken as a lower bound to the optimal solution

Not always it works…

See [BaMu96]
References

- **Articles**
References

Outline

- Introduction to WDM network design and optimization
- Integer Linear Programming approach to the problem
- Physical Topology Design
  - Unprotected case
  - Dedicated path protection case
  - Shared path protection case
  - Link protection
- Heuristic approach
Heuristic: method based on reasonable choices in RFWA that lead to a sub-optimal solution

- Connections are routed one-by-one
  - *In this case we will refer to an example, based on the concept of auxiliary graph*

Heuristic strategies can be:

- Deterministic: greedy vs. local search
  - Generic definitions in the following, together with a large example
- Stochastic:
  - E.g.: simulated annealing, tabu search, genetic algorithms
  - Not covered in this course
Greedy Heuristic (1)

Framework

- **Greedy**
  - Builds the solution step by step starting from scratch
  - Starts from an empty initial solution
  - At each iteration an element is added to the solution, such that
    - the partial solution is a partial feasible solution, namely it is possible to build a feasible solution starting from the partial one
    - the element added to the solution is the best choice, with respect to the current partial solution (the greedy is a myopic algorithm)

- **Features**
  - Once a decision is taken it is not discussed anymore
  - The number of iterations is known in advance (polynomial)
  - Optimality is usually not guaranteed.
Greedy Heuristic (2)

- We have to define
  - Structure of the solutions and the elements which belong to it
  - Criterion according to which the best element to be added is chosen
  - Partial solution feasibility check
begin {
    X := \emptyset, S := \text{set of choices/elements};
    \text{repeat} {
        \text{select the best element } e;
        S := S \setminus \{e\};
        \text{if}(X \cup e \text{ is a feasible partial solution})
        \text{then add } e \text{ to the partial solution } X;
    } \text{until } (S = \emptyset \text{ || } X \text{ completed})
} \text{end}
Given a feasible solution the Local Search tries to improve it.

- Starts with an initial feasible solution: the current solution $x^*$
- Returns the best solution found $x_b$
- At each iteration a set of feasible solutions close to the current one, the *neighborhood* $N(x^*)$, is generated
- A solution $x$ is selected among the neighbor solutions, according to a predefined policy, such that $x$ improves upon $x^*$
- If no neighbor solution improves upon the current one, (or stopping conditions are verified,) the procedure stops, otherwise $x_b = x^* = x$
Local search Heuristic (2)

Remarks

- Local search builds a set of solutions
- The set of neighbor solutions is built by partially modifying the current solution applying an operator called *move*
- Each neighbor solution can be reached from the current one by applying the move
- Local search moves from feasible solution to feasible solution
- Local search stops in a local optimum

We have to define

- The initial solution
- The way on which the neighborhood is generated:
  - The solution representation: a solution is represented by a vector $x$
  - The move which is applied to build the neighbors
  - selection policy
  - (stopping conditions)
Local search Heuristic (3)
Algorithm scheme

LocalSearch(P, x_b) {
    begin {
        select initial solution x*;
        stop = false;
        repeat {
            the neighborhood $N(x^*)$ is built;
            the neighbor $\tilde{x}$ is selected;
            if (no neighbor solution improves upon $x^*$)
                stop := true;
            else
                $x_b := x^* := \tilde{x}$; stop := false;
            } until (stop = true)
        } end
Static WDM mesh networks

*Optimization problem definition*

- **Design problem**
  - Routing, fiber and wavelength assignment for each lightpath

- **Design variables**
  - Number of fibers per link
  - Flow/routing variables
  - Wavelength variables

- **Cost function**: total number of fibers

- **Scenarios**:
  - With or without wavelength conversion
Heuristic for static design (RFWA)
A deterministic approach for fiber minimization

- Optimal design of WDM networks under static traffic
- A deterministic heuristic method based on one-by-one RFWA is applied to multifiber mesh networks:
  - RFWA for all the lightpaths is performed separately and in sequence (greedy phase)
  - Improvement by lightpath rerouting (consolidation phase)
- It allows to setup lightpaths so to minimize the amount of fiber deployed in the network
Heuristic static design (RFWA)
Network and traffic model specification

- Pre-assigned **physical topology** graph (OXCs and WDM links)
  - WDM **multifiber** links
    - Composed of multiple **unidirectional** fibers
    - Pre-assigned number of WDM **channels per fiber** (global network variable $W$)
  - OXCs
    - Strictly non-blocking space-switching architectures
      - In short, no block in the nodes
    - Wavelength conversion capacity: 2 scenarios
      - No conversion: Wavelength Path (WP)
      - Full-capability conversion in all the OXCs: Virtual Wavelength Path (VWP)

- Pre-assigned **logical topology**
  - Set of requests for optical connections
    - OXCs are the sources and destinations (add-drop function)
    - Multiple connections can be demanded between an OXC pair
    - Each connection requires a unidirectional lightpath to be setup
Heuristic static design (RFWA)
Routing Fiber Wavelength Assignment (RFWA)

(Some possible) Routing criteria

- Shortest Path (SP) → selects the shortest source-destination path (# of crossed links)
  - Different metrics are possible, e.g.:
    - Number of hops ($mH$)
    - Physical link length in km ($mL$)
- Least loaded routing (LLR) → avoids the busiest links
  - E.g., among the k-shortest paths, choose the one whose most loaded link is less loaded
- Least loaded node (LLN) → avoids the busiest nodes
Heuristic static design (RFWA)
Routing Fiber Wavelength Assignment (RFWA)

- Wavelength assignment algorithms
  - Pack → the most used wavelength is chosen first
  - Spread → the least used wavelength is chosen first
  - Random → random choice
  - First Fit → pick the first free wavelength

- Similarly, for fiber assignment criteria
  - First fit, random, most used, least used
Heuristic static design (RFWA)

Wavelength layered graph

- Physical topology
  (3 wavelengths)

- Equivalent wavelength layer graph

Heuristic static design (RFWA)
Multifiber (or Extended) wavelength layered graph

- Physical topology (2 fiber links, 2 wavelengths)

- Different OXC functions displayed on the layered graph
  - 1 - add/drop
  - 2 - fiber switching
  - 3 - wavelength conversion
  - 3 - fiber switching + wavelength conversion
Heuristic static design (RFWA)

*Multifiber (extended) wavelength layer graph*

- Enables the joint solution of R, F and W in RFWA
- Network topology is replicated \( W \cdot F \) times and nodes are opportunely linked among various layers
Heuristic static design (RFWA)

Layered graph utilization

- The layered graph is “monochromatic”
- Routing, fiber and wavelength assignment are solved together
- Links and nodes are weighted according to the routing, fiber and wavelength assignment algorithms
- The Dijkstra Algorithm is finally applied to the layered weighted graph
- Worst case complexity

\[ O(NFW^2) \]

\[
N = \# \text{ of nodes} \\
F = \# \text{ of fibers} \\
W = \# \text{ of wavelengths}
\]
Heuristic: Consolidation Phase
*Heuristic design and optimization scheme*

- **Connection request sorting rules**
  - Longest first
  - Most requested couples first
  - **Balanced**
  - Random

- **Processing of an individual lightpath**
  - Routing, Fiber and Wavelength Assignment (RFWA) criteria
  - Dijkstra’s algorithm performed on the multifiber layered graph
Heuristic for WDM mesh networks design

Case-study

- National Science Foundation Network (NSFNET): USA backbone
  - Physical Topology
    - 14 nodes
    - 44 unidirectional links
  - Design options
    - \( W: 2, 4, 8, 16, 32 \)
    - 8 RFWA criteria
Heuristic static design (RFWA)

RFWA-criteria labels
Heuristic static design (RFWA)
Total number of fibers

- The initial sorting rule resulted almost irrelevant for the final optimized result (differences between the sorting rules below 3%)
- Hop-metric minimization performs better
  - Variations of M due to RFWAs and conversion in the mH case below 5%
Heuristic static design (RFWA)

Fiber distribution in the network

- NSFNet with $W = 16$ wavelengths per fiber, mL SP RFWA criteria
- The two numbers indicate the number of fibers in the VWP and WP network scenarios
- Some links are idle
Heuristic static design (RFWA)

Wavelength conversion gain

- Wavelength converters are more effective in reducing the optimized network cost when $W$ is high

\[
G_M = \frac{M_{WP} - M_{VWP}}{M_{WP}} \times 100
\]
Heuristic static design (RFWA)

Saturation factor

$$\rho = \frac{C}{W \times M}$$

- A coarser fiber granularity allows us to save fibers but implies a smaller saturation factor
- VWP performs better than WP
  - Variations of $\rho$ due to RFWAs and metrics in the VWP case below 5%

\[ C = \text{number of used wavelength channels} \]
Heuristic static design (RFWA)

Optimization: longer lightpaths...

\[ \Delta = \frac{C - C_{SP}}{C_{SP}} \times 100 \]

- \( C_{SP} \) = number of used wavelength channels with SP routing and unconstrained resources (capacity bound)

Compared to the initial routing (shortest path, which is also the capacity bound) the fiber optimization algorithm increases the total wavelength-channel occupation (total lightpath length in number of hops):
- \( C \) is increased of max 10% of \( C_{SP} \) in the worst case.
Heuristic static design (RFWA) …*traded for fewer unused capacity*

\[ U = \left( 1 - \frac{C}{W \times M} \right) \times 100 \]

- Compared to the initial routing (shortest path) the fiber optimization algorithm decreases the total number of unused wavelength-channels
  - \( U \) is halved in the best case

**Notes**

- *Initial SP routing* curve: data obtained in the VWP scenario (best case)
- *Optimized solution* curve: averaged data comprising the WP and VWP scenarios
Heuristic static design (RFWA)

Convergence of the heuristic optimization

- The algorithm appears to converge well before the last iteration
- Improvements of the computation time are possible
- Notes:
  - Convergence is rather insensitive to the initial sorting rule and to the RFWA criteria
Case-study networks
Comparison of SF with heuristic optimization (VWP)

- ILP by SF is a useful benchmark to verify heuristic dimensioning results
  - Approximate methods do not share this property
A heuristic method for multifiber WDM network optimization under static traffic has been proposed and applied to various physical network scenarios.

Good sub-optimal solutions in terms of total nr of fibers can be achieved with a reasonable computation time.

The method allows to inspect various aspects such as RFWA performance comparison and wavelength conversion effectiveness.

Future possible developments:
- Upgrade to include also lightpath protection in the WDM layer
- Improvement of the computation time / memory occupancy
Past / current research lines

Industrial partnerships


OTN design and simulation

ASON/GMPLS control plane

Multi-domain routing

Semi-transparent OTN, physical impairments;

SLA-aware routing;

Multi-layer TE, learning algorithms

Resilience, protection, availability

CoreCom

Telecom Italia

Past / current research lines

Industrial partnerships


OTN design and simulation

ASON/GMPLS control plane

Multi-domain routing

Semi-transparent OTN, physical impairments;

SLA-aware routing;

Multi-layer TE, learning algorithms

Resilience, protection, availability

CoreCom

Telecom Italia

Past / current research lines

Industrial partnerships


OTN design and simulation

ASON/GMPLS control plane

Multi-domain routing

Semi-transparent OTN, physical impairments;

SLA-aware routing;

Multi-layer TE, learning algorithms

Resilience, protection, availability

CoreCom

Telecom Italia

Past / current research lines

Industrial partnerships


OTN design and simulation

ASON/GMPLS control plane

Multi-domain routing

Semi-transparent OTN, physical impairments;

SLA-aware routing;

Multi-layer TE, learning algorithms

Resilience, protection, availability

CoreCom

Telecom Italia

Past / current research lines

Industrial partnerships


OTN design and simulation

ASON/GMPLS control plane

Multi-domain routing

Semi-transparent OTN, physical impairments;

SLA-aware routing;

Multi-layer TE, learning algorithms

Resilience, protection, availability

CoreCom

Telecom Italia

Past / current research lines

Industrial partnerships


OTN design and simulation

ASON/GMPLS control plane

Multi-domain routing

Semi-transparent OTN, physical impairments;

SLA-aware routing;

Multi-layer TE, learning algorithms

Resilience, protection, availability

CoreCom

Telecom Italia

Past / current research lines

Industrial partnerships


OTN design and simulation

ASON/GMPLS control plane

Multi-domain routing

Semi-transparent OTN, physical impairments;

SLA-aware routing;

Multi-layer TE, learning algorithms

Resilience, protection, availability

CoreCom

Telecom Italia

Past / current research lines

Industrial partnerships


OTN design and simulation

ASON/GMPLS control plane

Multi-domain routing

Semi-transparent OTN, physical impairments;

SLA-aware routing;

Multi-layer TE, learning algorithms

Resilience, protection, availability

CoreCom

Telecom Italia

Past / current research lines

Industrial partnerships


OTN design and simulation

ASON/GMPLS control plane

Multi-domain routing

Semi-transparent OTN, physical impairments;

SLA-aware routing;

Multi-layer TE, learning algorithms

Resilience, protection, availability

CoreCom

Telecom Italia

Past / current research lines

Industrial partnerships


OTN design and simulation

ASON/GMPLS control plane

Multi-domain routing

Semi-transparent OTN, physical impairments;

SLA-aware routing;

Multi-layer TE, learning algorithms

Resilience, protection, availability

CoreCom

Telecom Italia

Past / current research lines

Industrial partnerships


OTN design and simulation

ASON/GMPLS control plane

Multi-domain routing

Semi-transparent OTN, physical impairments;

SLA-aware routing;

Multi-layer TE, learning algorithms

Resilience, protection, availability

CoreCom

Telecom Italia

Past / current research lines

Industrial partnerships


OTN design and simulation

ASON/GMPLS control plane

Multi-domain routing

Semi-transparent OTN, physical impairments;

SLA-aware routing;

Multi-layer TE, learning algorithms

Resilience, protection, availability

CoreCom

Telecom Italia

Past / current research lines

Industrial partnerships


OTN design and simulation

ASON/GMPLS control plane

Multi-domain routing

Semi-transparent OTN, physical impairments;

SLA-aware routing;

Multi-layer TE, learning algorithms

Resilience, protection, availability

CoreCom

Telecom Italia

Past / current research lines

Industrial partnerships


OTN design and simulation

ASON/GMPLS control plane

Multi-domain routing

Semi-transparent OTN, physical impairments;

SLA-aware routing;

Multi-layer TE, learning algorithms

Resilience, protection, availability

CoreCom

Telecom Italia

Past / current research lines

Industrial partnerships


OTN design and simulation

ASON/GMPLS control plane

Multi-domain routing

Semi-transparent OTN, physical impairments;

SLA-aware routing;

Multi-layer TE, learning algorithms

Resilience, protection, availability

CoreCom

Telecom Italia
Past / current research lines

Network design and simulation tools

OPTCORE

OTN design and simulation

Resilience, protection, availability

SLA-aware routing;

Semi-transparent OTN, physical impairments;

ASON/GMPLS control plane

Multi-domain routing

Carrier-grade Ethernet

Multi-layer TE, learning algorithms


year
OptCore: the WDM optimization tool

OptCore is the tool to assist WDM network operators in their offline design and optimization duty

- Finds the best matching between the physical topology and the set of requests for lambda connections
- Supports all the main WDM-layer protection techniques
- Provides data for OXC and OADM configuration
- Allows to inspect and try several network scenarios, including transparent, opaque or partially-opaque WDM networks
- Solves green-field design as well as network re-design under legacy element constraints
- Is applicable to large systems with 128 and more wavelengths per fiber, multi-fiber links and high-dense connectivity
- Performs dynamic-traffic simulation to assess the capability of a network to support unexpected lambda connection requests and traffic expansion
OptCore architecture and scheme

- OC-layer topology (OC request matrix)
- Network physical topology (links and nodes)
- Wavelength-conversion capability
- Number of wavelengths per fiber \( W \)
- Protection scheme (e.g. unprotected, dedicated, shared)
- Routing, fiber and wavelength assignment algorithms
- Initial conditions (e.g. initial # fibers)

User interface (XML-based)
Processing tool
User interface (XML-based)

- Working and spare resources allocation
- OXC and converters configuration
- Cost-function minimization
- Initial conditions (e.g. initial # fibers)
OptCore graphical interface

Workspace window
OptCore graphical interface

Output display (I)
OptCore graphical interface

Output display (II)
Integer Linear Programming (ILP)
- The most general exact method
- Exponential computational complexity $\Rightarrow$ not scalable
- The number of variables and constraints is huge
  - It can be often decreased by choosing an appropriate formulation (e.g. constrained-route, source, etc.)
- Variables defined in the integer domain
  - Branch and bound technique required

Heuristic methods
- Polynomial-complexity algorithms
- No guarantee that the solution found is the optimum
  - Close to ILP results in many tested cases
- A very good alternative to ILP for realistic-dimension problems
- Additional slides on Bhandary’s algorithm for link-disjoint paths
- Courtesy of Grotto Networking
Diverse Route Computation

- **Problem:**
  - Find two disjoint paths between the same source and destination, with minimum total cost (working + protection)
    - We may want the paths node diverse (stronger)

- **Applications:**
  - Dedicated Path Protection
2-step: diverse paths via pruning

Find the first shortest path, then prune those links

Primary path
cost = 3
2-step: diverse paths via pruning

Find the second shortest path in the modified graph

Backup path cost = 12
Total for both paths= 15
2-step: diverse paths via pruning

Questions
- Are these the lowest cost set of diverse paths?
  - Why should they be? We computed each separately…
- Are there situations where this approach will completely fail?
  - Let’s look at another network…
2-step approach on trap topology

Step 1. Find the shortest path from source to destination

Step 2. Prune out links from the path of step 1.
2-step approach on trap topology

Step 3. Can’t find another path from 1 to 6 since we just separated the graph.

Hmm, maybe a solution doesn’t exist? Or maybe we need a new algorithm?
Diverse Path Calculation Methods

Answer: Find a better algorithm

Algorithms:
- Suurballe’s algorithm: transforms the graph in a way that two regular Dijkstra computations can be performed.
Diverse Path Algorithm: Example

Step 1: Compute least cost primary path

Primary path is A-B-C-Z, cost=3
Diverse Path Algorithm: Example

Step 2: Treat graph as directed. In all edges in the primary path: Set forward cost to $\infty$. Set reverse cost to negative of original cost.
Diverse Path Algorithm: Example

Step 3: Find least cost back-up path

Back-up path is A-C-B-Z, cost=6
Diverse Path Algorithm: Example

Step 4: Merge paths. Remove edges where primary and back-up traverse in opposite directions.

New primary is A-B-Z, back-up is A-C-Z, total cost = 9