GMPLS Label Space Minimization through Hypergraph Layouts

Jean-Claude Bermond — David Coudert —
Joanna Moulierac — Stéphane Pérennes —
Ignasi Sau — Fernando Solano Donado

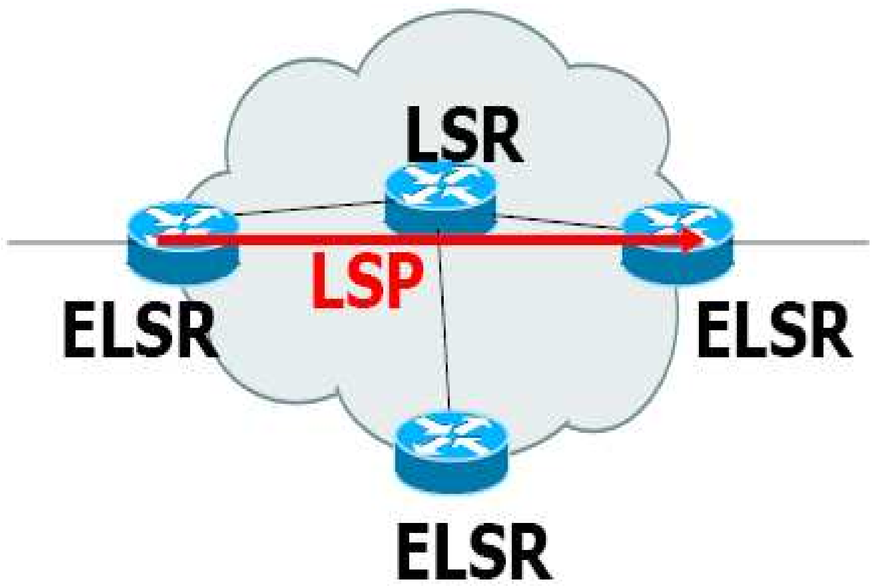
http://hal.archives-ouvertes.fr/docs/00/42/66/81/PDF/RR-7071.pdf

Outline

- GMPLS basic principles
- Our Problem: Minimization of label space
  - Modelling
  - Examples
- The problem on the path
- Proof of NP-Completeness
- Integer Linear Programming

MPLS terminology

- **LSR**: Label Switching Router
- **ELSR**: Edge Label Switching Router
- **LSP**: Label Switched Path
- **FEC**: Forwarding Equivalence Class

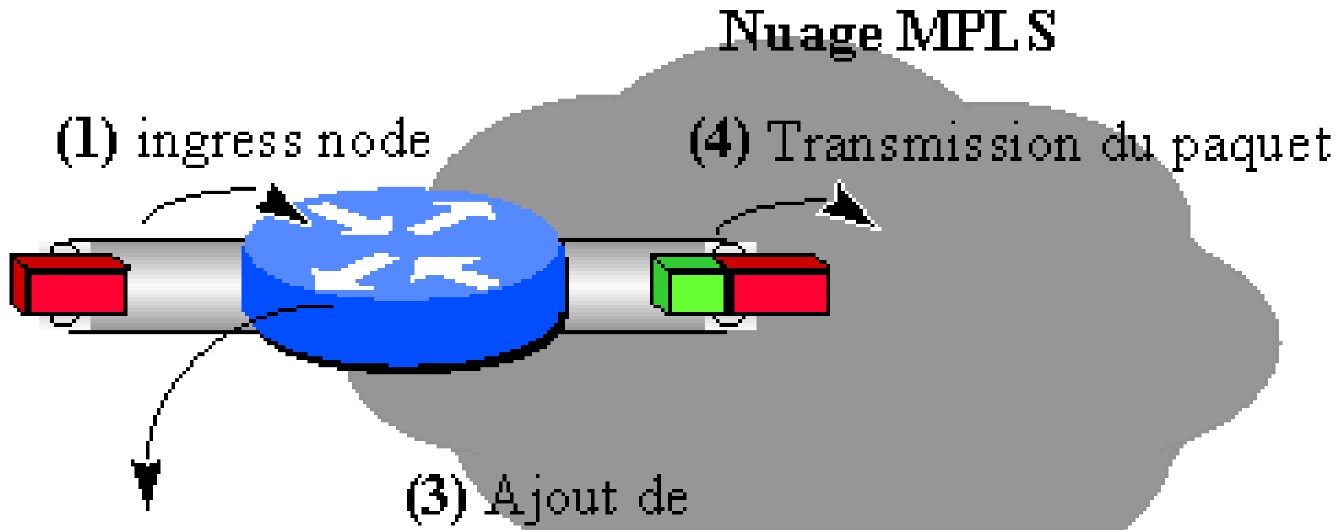


Forwarding MPLS

- **Forwarding**: based on the analysis of the label which gives the route towards a destination
- **Control**: creation and maintenance of the label table

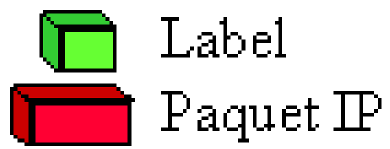
- Routes are learned from an IGP (*Interior Gateway Protocol*)
- Distribution of labels with a specific protocol (LDP, RSVP-TE)

At the entrance of the MPLS domain



(2)Table de comm

InPort	@Ipdst	Label	OuPor
P1	193.48.0.0/16	L5	P3
P2	193.48.0.0/16	L7	P4



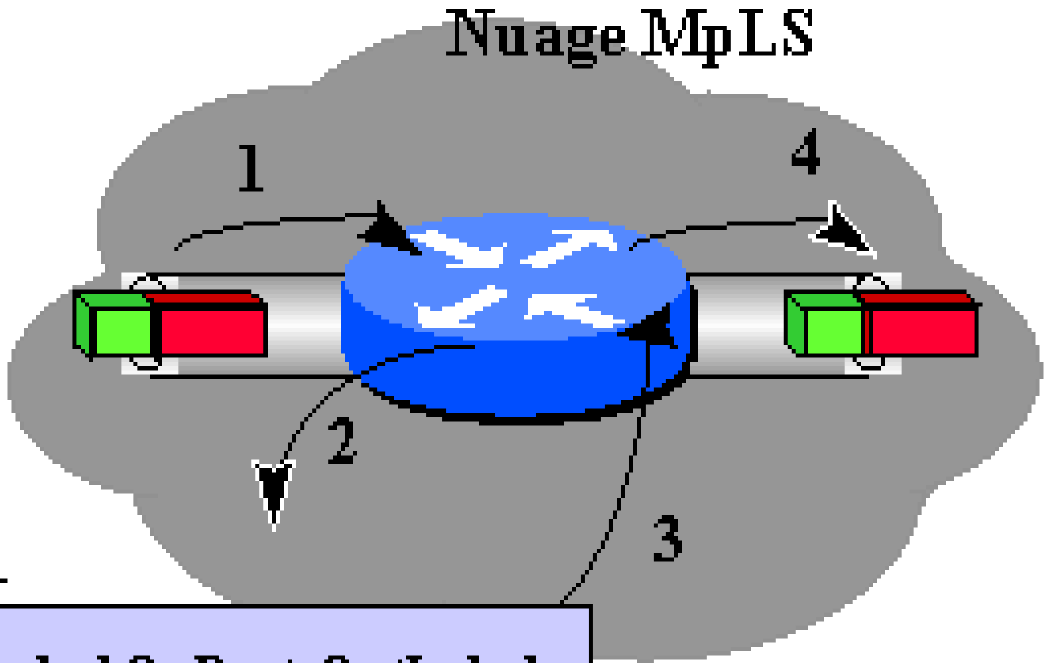
1 - Le paquet IP arrive sur l'ingress node

2 - Le protocole de routage IP détermine, à partir de l'adresse IP de l'eggress node, la FEC, le label et le port de sortie.

3 - Ajout de l'en-tête

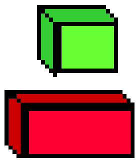
4 - Paquet IP + Label envoyé vers le noeud suivant

Inside the domain



LIB du LSR

InPort	InLabel	OuPort	OutLabel
P1	L5	P3	L21
P2	L7	P4	L15



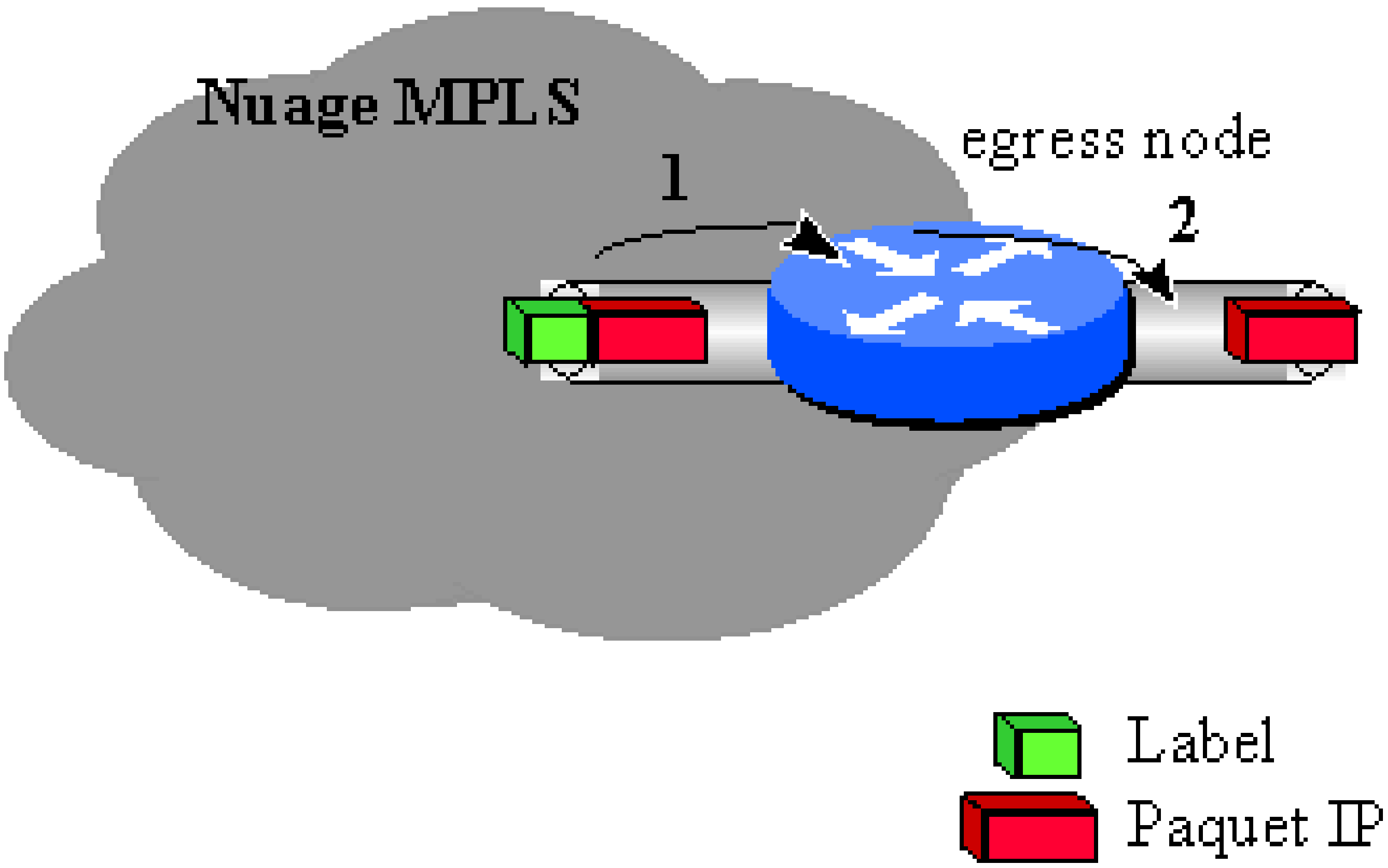
1 - Paquet IP + Label arrive sur le LSR

2 - Protocole de routage détermine, à partir de la LIB le next hop LSR ou E-LSR

3 - Mise à jour du label MPLS

4 - Paquet IP + Label envoyé vers le noeud suivant

At the exit of the MPLS domain



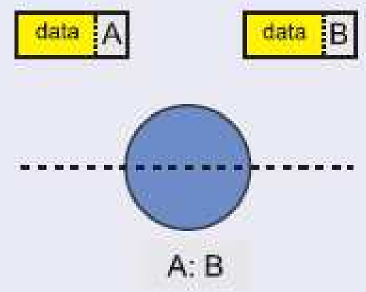
1 - Paquet IP + label arrive sur l'egress node

2 - Retrait du label et transmission du paquet IP à la couche réseau

Label swapping

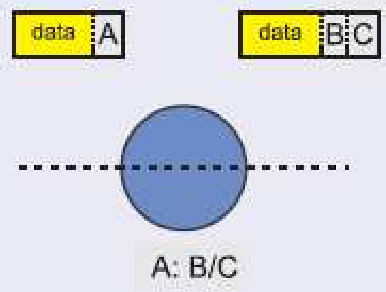
Packets contain a stack of labels...
Nodes may alter the stack of labels performing one of the following operations:

Swap the Top



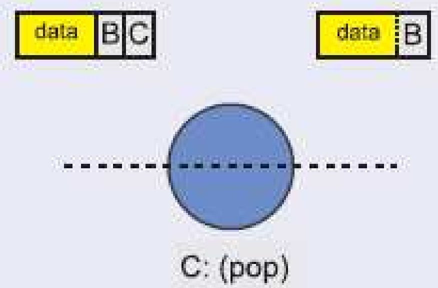
... with the neighbor's locally 'understandable' label.

Swap & Push



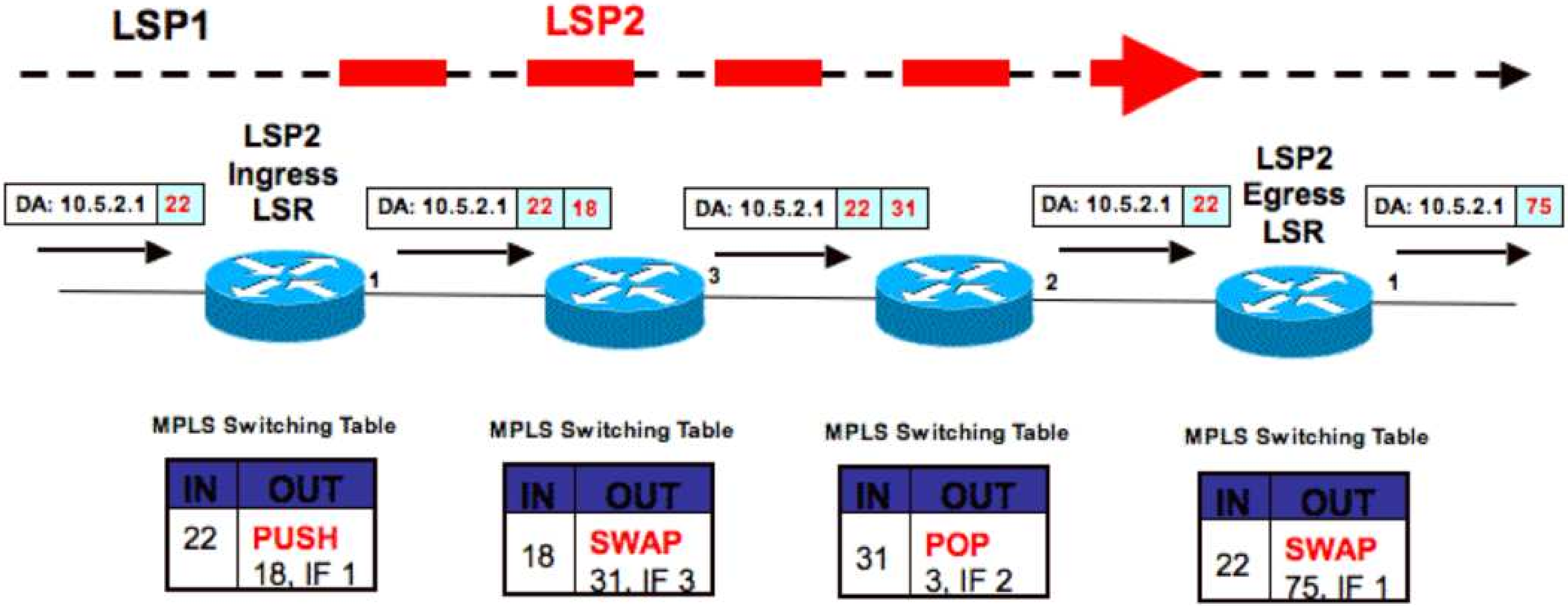
... as many as you want.

Pop the Stack



... only one.

Label stacking



GMPLS

About GMPLS switching...

- G/MPLS is a tag-switching technology (packet-based networks).
- Each packet is tagged (labeled), so it can be associated and treated as a single flow.
- Packet forwarding is based on the content of the label solely.

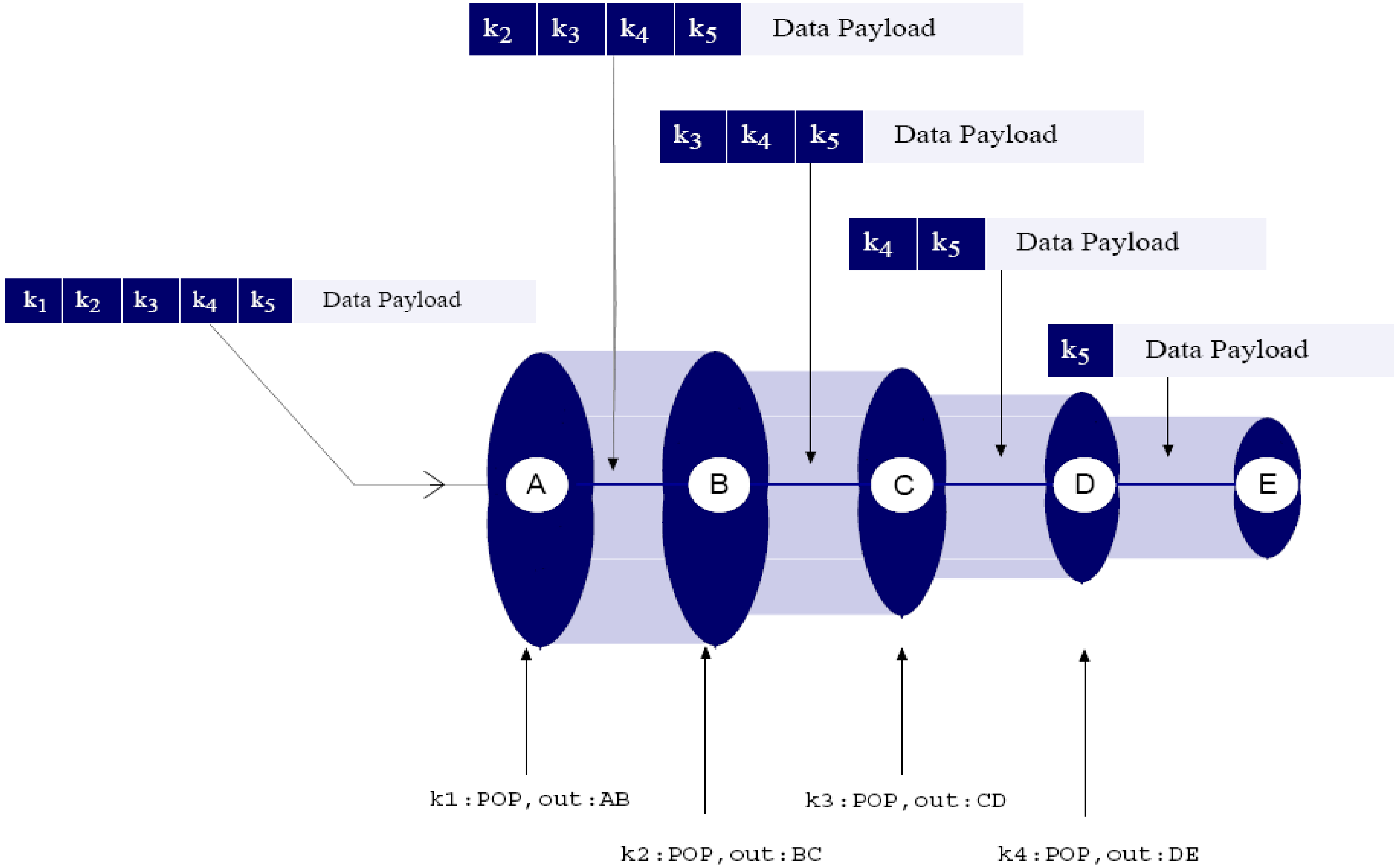
About AOLS...

1. All-Optical Label Switching (AOLS) is an all-optical hardware implementation of GMPLS switching for All-Optical Packet Switching technologies.
2. Label processing and packet forwarding decisions are all performed completely optically
3. No need for OEO regeneration... faster packet forwarding.
4. *One 'decoding' device is needed for each used label at each node*
5. Few labels... Labels are extremely valuable resources

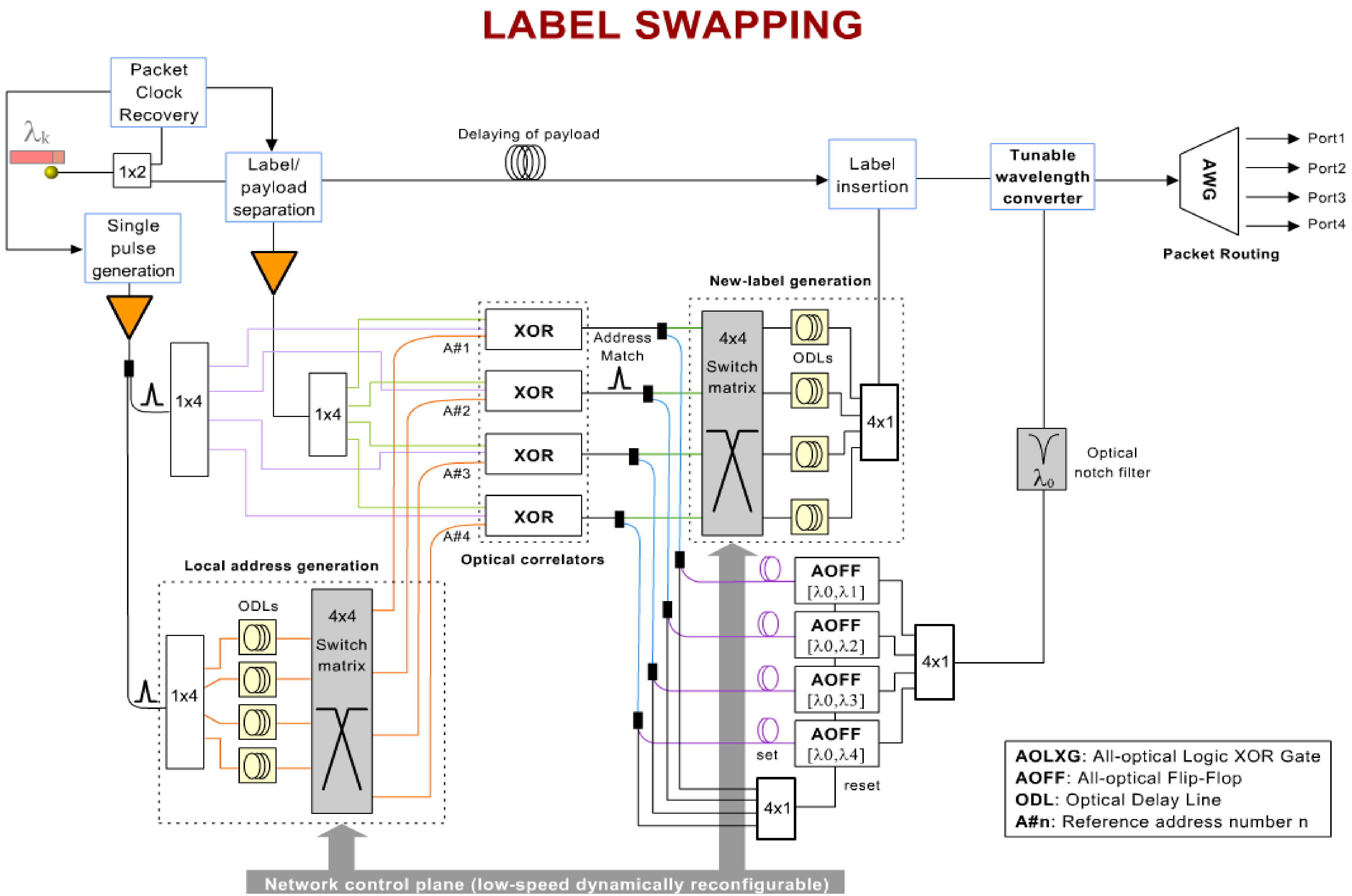
Hypothesis for our problem

- Traffic can enter in any node of the tunnel, but can exit only at the end of the tunnel
- We consider only label stacks of size 2
- Shortest paths used for the routing
- In AOLS network (experimental ones where labels are processed optically), one device is needed per label processed in nodes
  - Minimization of the total number of labels

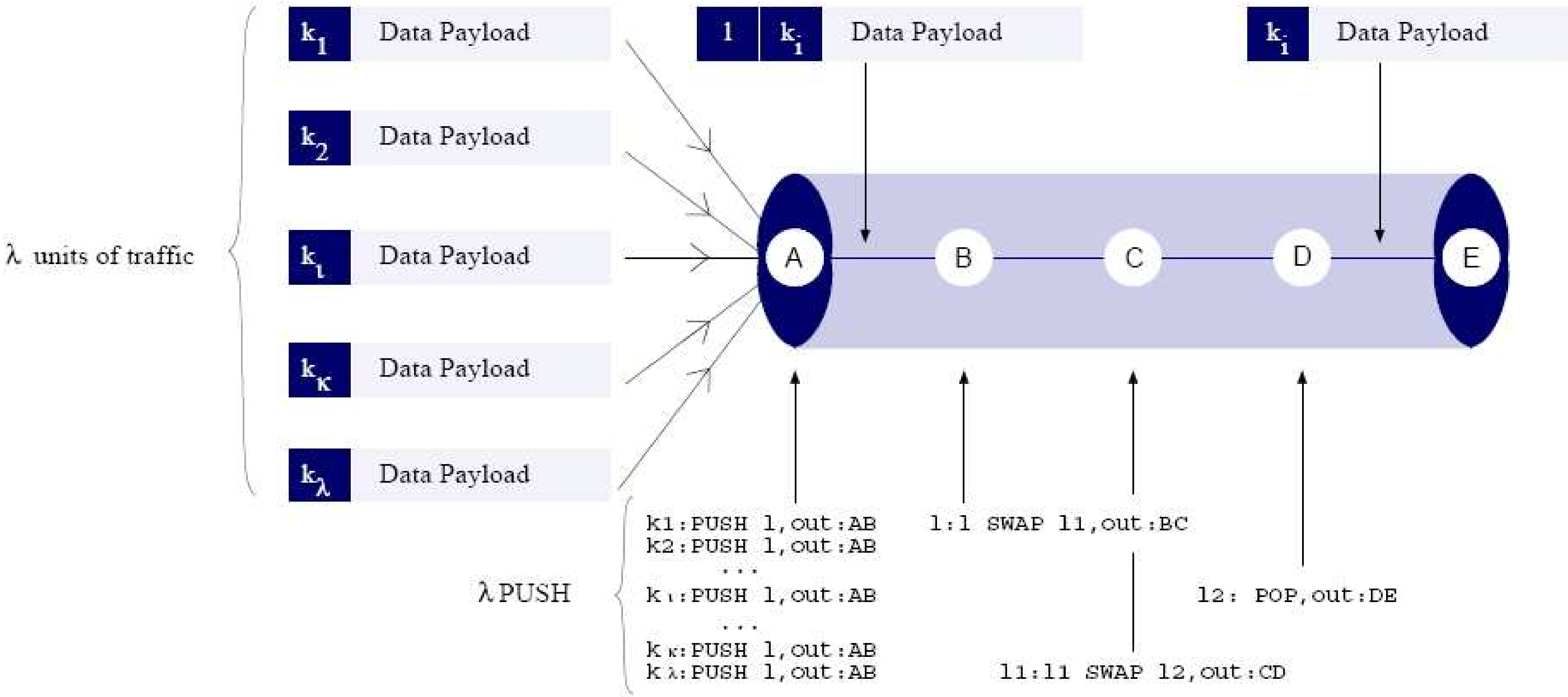
Label Stripping: the larger the stack the fewer the labels needed



LASAGNE Project: All-optical LAbel SwApping employing optical logic Gates in NEtwork nodes



Cost of a tunnel



Cost of the tunnel = #Units of traffic + length (tunnel)-1

= λ + 3

If no tunnel is used, the number of labels is : 4*λ

Notation

- $G = (V, E)$ is the underlying digraph (which can be symmetric or not) with $|V| = n$.
- $r_{i,j}$ is the request from node $i \in V$ to node $j \in V$, with multiplicity $m_{i,j}$, (the traffic being differentiated, $m_{i,j}$ represents the number of different flows from i to j and j is not necessarily the final destination of the request $r_{i,j}$). R is the set of all requests.
- $P(G)$ is the set of all simple dipaths in G.
- t stands for a tunnel, and T is the set of tunnels, that is $t \in T \subseteq P(G)$.
- ℓ is a length function on the arcs, that is $\ell : E \to \mathbb{N}^+$.
- A tunnel t has length $\ell(t) = \sum_{e \in t} \ell(e)$ and carries $w(t)$ flows, or as referred in the rest of the report, $w(t)$ units of traffic.

Label Space Reduction Problem

Note that, a priori, $w(t)$ depends on the routing policy. The cost $c(t)$ of a tunnel t is then $c(t) = w(t) + (\ell(t) - 1)$, and the cost of a set of tunnels T is

$$c(T) = \sum_{t \in T} (w(t) + \ell(t) - 1) . \quad (1)$$

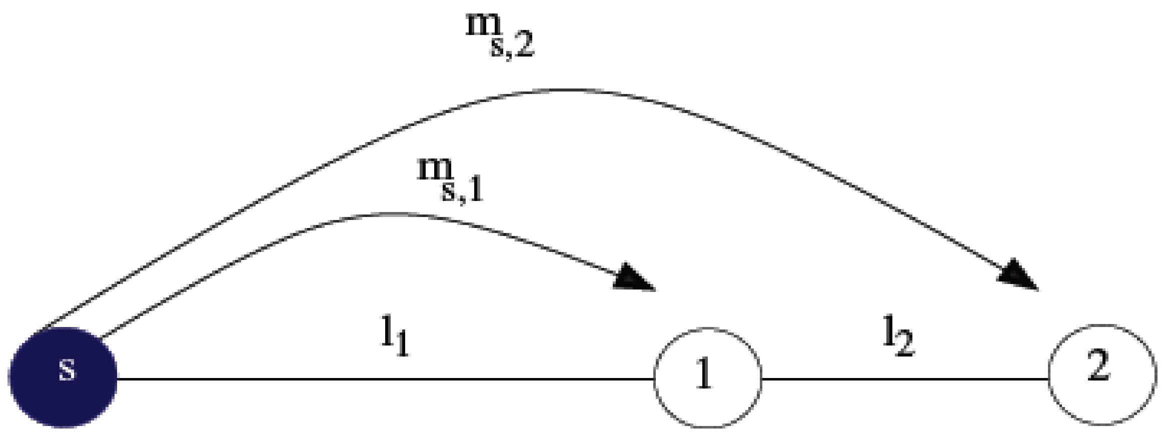
LSPR problem:

INPUT: Given a digraph G and a set R of requests,

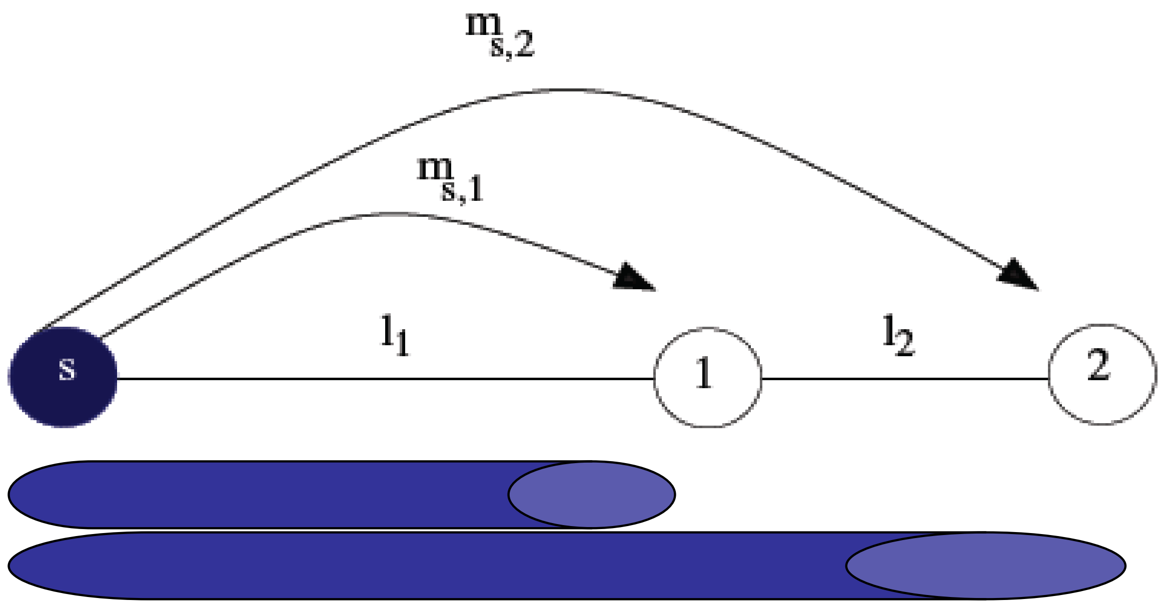
OUTPUT: find a set of tunnels enabling to route all the requests and a dipath of tunnels for each request,

OBJECTIVE: Minimize the total cost of tunnels where the cost is defined as in Equation (1)

A solution on a simple example



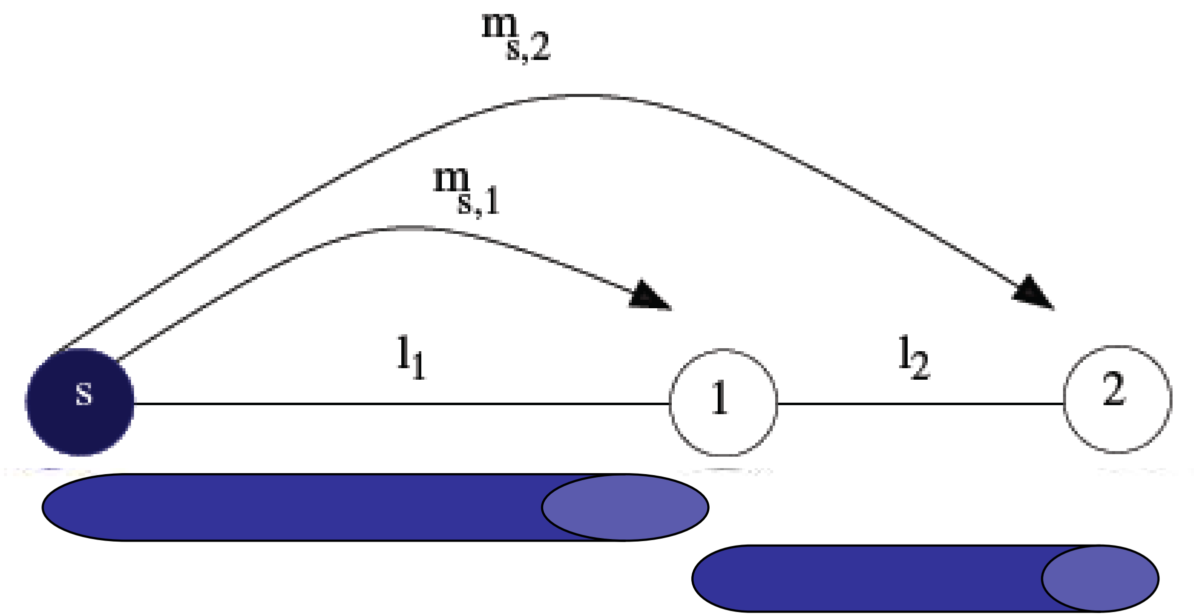
A solution on a simple example



If l_1 ≤ $m_{s,2}$, the solution is composed of tunnels (s,1) and (s,2)

Cost = $(m_{s,1} + l1 - 1) + (m_{s,2} + l1 + l2 - 1) = m_{s,1} + m_{s,2} + \mathbf{2l1} + l2 - 2$

A solution on a simple example



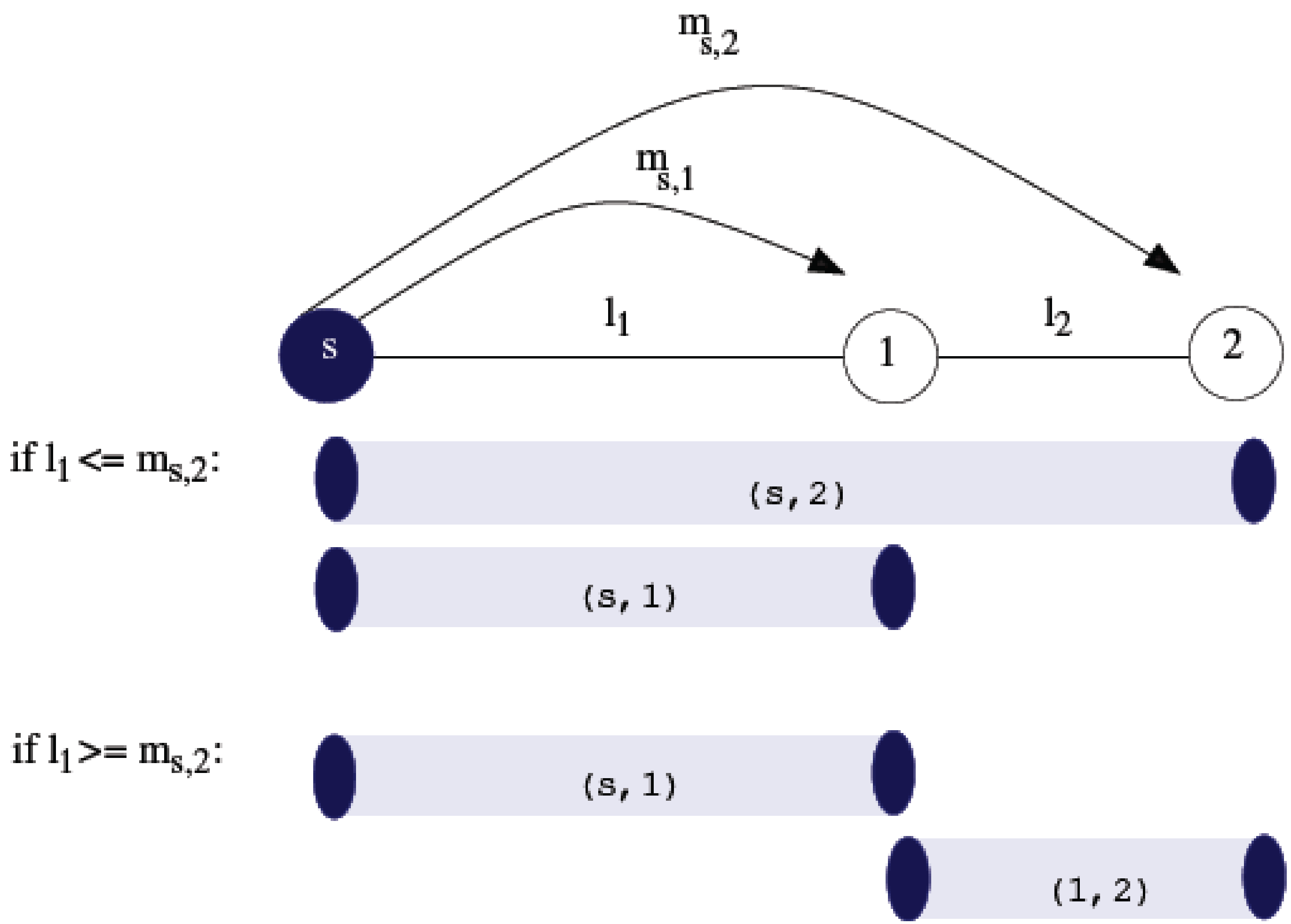
If $l_1 \leq m_{s,2}$, the solution is composed of tunnels (s,1) and (s,2)

Cost = $(m_{s,1} + l1 - 1) + (m_{s,2} + l1 + l2 - 1) = m_{s,1} + m_{s,2} + \mathbf{2l1} + l2 - 2$

Else, the solution is composed of tunnels (s,u_1) and (u_1,u_2)

Cost = $(m_{s,1} + m_{s,2} + l1 - 1) + (m_{s,2} + l2 - 1) = m_{s,1} + \mathbf{2\,m_{s,2}} + l1 + l2 - 2$

Which one the best solution?



Cost of a solution

$$c(\mathcal{T}) = \sum_{k=1}^{|\mathcal{T}|} (l(T_k) - 1) + \sum_{r=1}^{|\mathcal{R}|} h_r w_r .$$

the cost for the configuration of the tunnels

the cost for the requests to enter the tunnels

- hr is the number of consecutive tunnels for the request r in T,
- wr is the number of units of traffic of the request r and
- l(Tk) is the length of the tunnel Tk in terms of number of hops.

General network

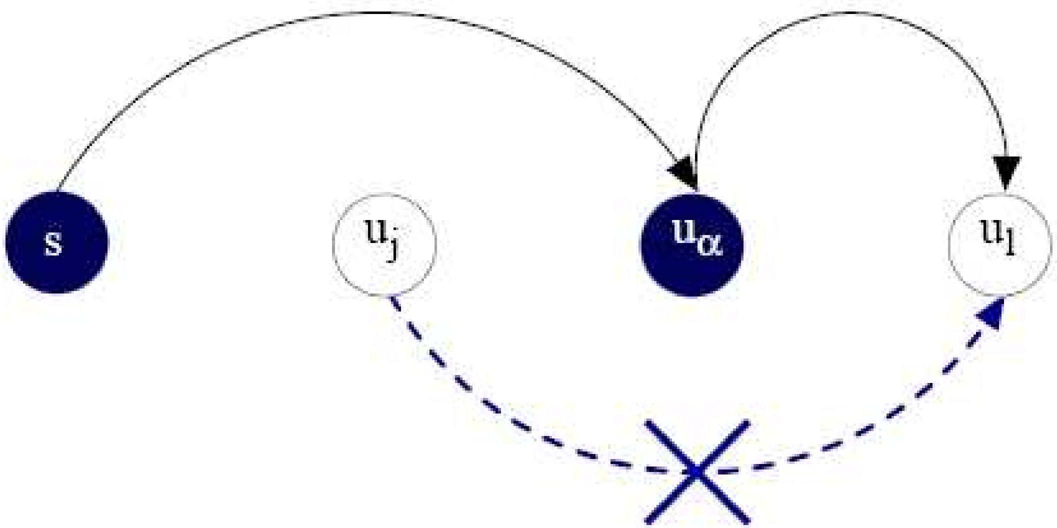
- In any network, there exists an optimal solution for the problem s.t. all the units of traffic of requests (si,uj) are routed via a unique dipath (set of consecutive tunnels) from si to uj
- *Proof…*
- In the path, there is only one tunnel ending in a node.

The path

- Some notations:
  - OPT(s1,s2,…,sk;[s1,n]) the cost of an optimal solution with k sources on the path [s1,n]
  - α(i) : For a given node i and an interval [u,v], let α(i) be the rightmost endvertex in [u,v] of a tunnel starting in i (said otherwise, (i,α(i)) is the longest tunnel issued from i and ending in [u; v]).
- On the path [1,n], when all the requests are destined to node n, the optimal solution is simply the tunnel from node i to node n where i is the leftmost source.

The line network, one source

- When the network is a directed line, with source s, an optimal solution T for LSPR-L1 problem is such that, if (s, u_α) is the longest tunnel from s, then there is no tunnel (u_j , u_l) in T with $j < \alpha < l$.
- *Proof…*



Cost of an optimal solution

Proposition 3. *The cost of an optimal solution* $OPT[i,j]$ *for problem* MINIMUM COST HYPERGRAPH LAYOUT *on the dipath* $[i,j]$ *with source* i *may be expressed as follows:*

$$OPT[i,j] = \min_{i<\alpha(i)\leq j} C_{\alpha(i)}[i,j]$$

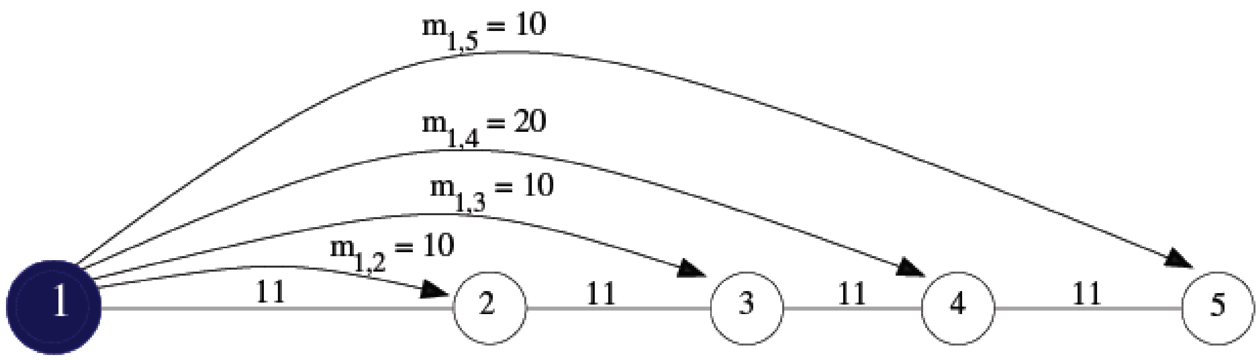
with $C_{\alpha(i)}[i,j] =$

$$\left(\sum_{k=\alpha(i)}^{j} m_{1,k} + \sum_{e \in E([i,\alpha(i)])} \ell(e) - 1 \right) + OPT[i,\alpha(i)-1] + OPT[\alpha(i),j].$$

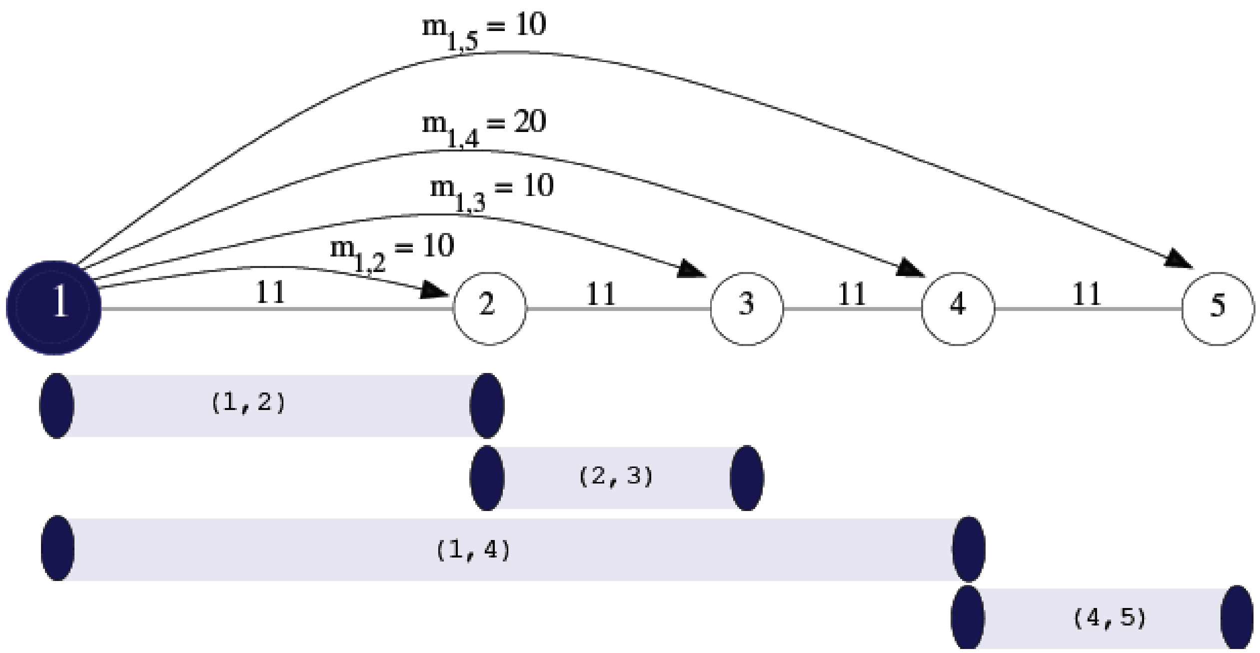
OPT is a table of size $n \times n$ indicating the costs all the subsolutions;
S is a table of size $n \times n$ indicating the $\alpha(i)$ associated to the optimal subsolutions;
M is a table of size n storing partial sums of weigths, $M[1] = 0$, $M[j] = \sum_{i=2}^{j} m_{1,i} = M[j-1] + m_{1,j}$

```
for i ∈ [1, n] do
    OPT[i, i] = 0;

for i ∈ [1, n − 1] do
    OPT[i, i + 1] = m_{1,i+1} + ℓ([i, i + 1]) − 1; //Cost of the tunnel (i, i + 1) carrying traffic towards i + 1
    S[i, i + 1] = i + 1;

for k ∈ [2, n] do
    for ∀i ∈ [1, n − k] do
        min = +∞;
        //We consider all requests from 1 to some node in the interval [i + 1, i + k]
        for ∀α(i) ∈ [i + 1, i + k] do
            //value is the cost of the solution if α(i) is the splitting point for subdipath [i, i + k]
            value = (M[i+k]−M[α(i)−1])+ℓ([i, α(i)])−1+OPT[i, α(i) − 1]+OPT[α(i), i + k];
            if value < min then
                min = value;
                OPT[i, i + k] = value;
                S[i, i + k] = α(i);

Compute the optimal set of tunnels from the table S;
```

Solution computed with dynamic programming algorithm



Solution computed with dynamic programming algorithm



	$s = 1$	2	3	4	5
$s = 1$	0	20	50 (2)	101 (3)	132 (4)
2	-	0	20	61 (4)	91 (4)
3	-	-	0	30	60 (4)
4	-	-	-	0	20
5	-	-	-	-	0

When the requests are uniform on the path

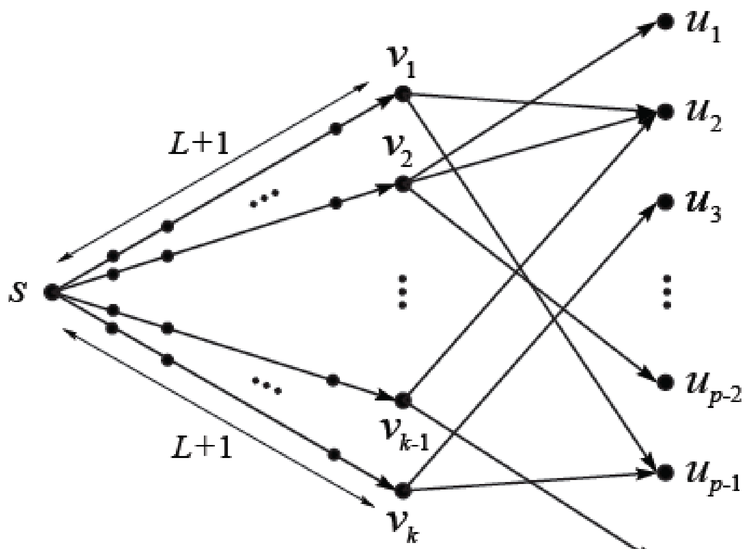
Proposition 4. *Let the network be a path* $[1, n]$ *with* $n = 2^q + r$, *where* $0 \leq r < 2^q$, *such that* $\ell([i, i+1]) = 1$ *for all* $i \in [1, n-1]$, *and with a unitary distribution, that is, for all* $\forall i \in [2, n], m_{1,i} = 1$. *Then the cost of an optimal solution is* $2^q(q-1) + 1 + (q+1)r$.

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
c^*	1	3	5	8	11	14	17	21	25	29	33	37	41	45	49	54	59	64	69

Proof of NP-Completeness

- A decision problem *C* is NP-complete if:
  1. *C* is in NP, and
  2. Every problem in NP is reducible to *C* in polynomial time.
- C is in NP if a candidate solution to *C* can be verified in polynomial time.
- A problem *K* is reducible to *C* if there is a polynomial-time many-one reduction, a deterministic algorithm which transforms any instance $k \in K$ into an instance $c \in C$, such that the answer to *c* is *yes* if and only if the answer to *k* is *yes*.
- To prove that *C* is an NP-complete problem, it is sufficient to show that an already known NP-complete problem reduces to *C*.

Reduction to Minimum Vertex Set-cover



To a set cover instance with sets S1, S2,…,Sk, with Si ⊆{a1,a2,…an}

We start with a distinguished node s

For each set Si, we introduce a node vi and a directed path of length L+1 from s to vi through L new vertices

For each element aj, we introduce a new vertex uj and for each vertex vi we add the arcs (vi,uj) if aj ∈Si

The requests are from s to uj, for j=1,…,p and with multiplicity=1.

ILP on the path

Input: Set of demands $D = 1, 2, \dots, d$, (s_d, u_d) with weigth w_d.
Output: Set of tunnels (i, j) with $i < j$.
Objective: Minimize

$$\sum_i^j y_{ij} + \sum_{i<j} u_{ij}$$

$$\sum_{j:i<j} x_{ij}^d - \sum_{k:k<i} x_{ki}^d = \begin{cases} 1 & \text{if } i = s_d; \\ 0 & \text{if } i \neq \{s_d, u_d\}; \\ -1 & \text{if } i = u_d; \end{cases} \quad \forall s_d \leq i \leq u_d \tag{1}$$

$$y_{ij} = \sum_{d \in D} x_{ij}^d . w_d \tag{2}$$

$$x_{ij}^d \leq u_{ij} \quad \forall d, i, j \tag{3}$$

$$u_{ij} \leq u_{kj} \quad \forall i < k < j \tag{4}$$

ILP for the general graph

- …

Conclusion

- Proof of NP-Completeness when LSPs for requests are given
- Algorithms for trees, meshs, general graphs…
- If paths are not directed? (i.e., we can go backwards)
- What if label stacks are larger than 2…

References

- **GMPLS Label Space Minimization through Hypergraph Layouts.** J-C. Bermond et al. INRIA Research Report N° 7071 October 2009
- **MPLS label stacking on the line network.** J.-C. Bermond et al. In Proc. of IFIP Networking, volume 5550 of LNCS, 2009.
- **Minimization of Label Usage in GMPLS Networks.** M. Pioro et al. Technical Report, Warsaw University of Technology, April 2009.