Efficient Implementation of Scheduling Algorithms for Optical Burst/Packet Switching

F. Callegati, A. Campi, W. Cerroni
DEIS – University of Bologna
(fcallegati,acampi,wcerroni)@deis.unibo.it

Problem

- Optical Burst Switching (OBS)
  – Scheduling has to be planned when the burst control packet (BCP) arrives at the network nodes
- Optical Packet Switching (OPS)
  – Scheduling has to be planned after reading the packet header in OPS

Calculate the exact transmission time of the burst/packet and planning the resource usage

Assumptions

- Variable length data payloads
- Burst/packet arrive asynchronously
- Network nodes :
  - full wavelength conversion
  - network paths are related to fibers
  - equipped with some sort of delay buffer

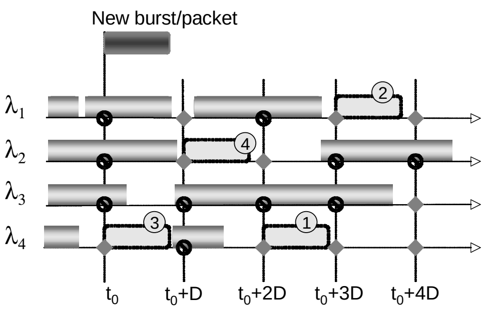
Switch control logic tasks

- Processing the BCP or the packet header
- Choosing the forwarding path
- Solve congestion:
  - Choose the wavelength on the designated output fiber
  - Choose the delay, in case all wavelengths are busy at the time of packet arrival
  - Drop the packet if no wavelength and delay are available.

Wavelength and Delay Selection (WDS) scheduling problem

WDS algorithms

- Delay oriented algorithms (D type)
  - Min lentgth (1)
  - Min lentgth-VF (3)
- Gap oriented algorithms (G type)
  - Min gap (2)
  - Min gap-VF (4)



WDS problem description

- De-couple information storage and search processing
  - Heap memory, efficient to store a large amount of data
  - Search with dedicated hardware, efficient computation
- Computation by simple min/max functions
- N-tree search structure
  - Min/max function on a subset of data
  - Perform all types of WDS algorithms
  - Costs depend on the number of the total paths and not on the number of free gaps
    - Fixed size costs

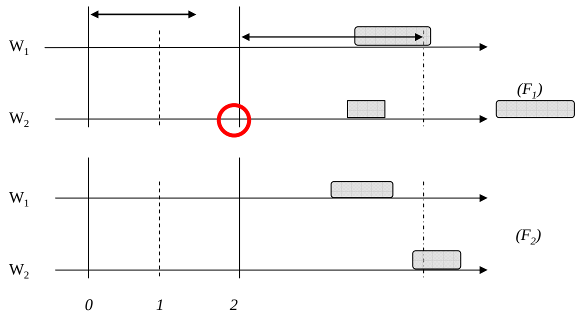
WDS problem formulation

- Define the set of paths
  - ***T*** : (*f*,*w*,*b*) | $f \in [1\ldots F]$, $w \in [1\ldots W]$, $b \in [0\ldots B]$
- The matrix $S = \{ s(f,w,b) : (f,w,b) \in T \}$
  - $s(f,w,b)$ is the time gap
    - At wavelength *w*
    - On fiber *f*
    - Between delay *bD* and either the next scheduled packet or the end of the scheduling time window.
  - $s(f,w,b) = -1$ if *f,w,b is not available*
- *Packet of length x arrives :*
  - $F_{f,w,b}(x) = s(f,w,b) - x$
  - T': $(f,w,b) \mid F_{f,w,b}(x) \geq 0$

WDS problem formulation

- D-type
  - Min lentgth: find (*f*,*w*,*b*) | max($F_{f,w,b}$(x)), $(f,w,b) \in T'$
  - Min lentgth-VF: find (*f*,*w*,*b*) | min(*b*), $(f,w,b) \in T'$
- G-type
  - Min gap: find (*f*,*w*,*b*) | max($F_{f,w,b}$ (x)), $(f,w,b) \in T'$
  - Min gap-VF: find (*f*,*w*,*b*) | min($F_{f,w,b}$ (x)), $(f,w,b) \in T'$

Example of matrix S, Min length



delays

wavelenght (rows) — fiber F_1

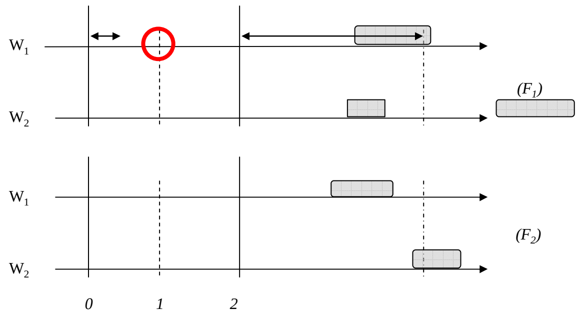
	0	1	2
0	-1	-1	3
1	-1	-1	-1

wavelenght (rows) — fiber F_2

	0	1	2
0	-1	4	3
1	-1	-1	3

fiber F_1 *fiber* F_2

Example of matrix S, Min length-VF



delays

wavelenght (rows) — fiber F_1

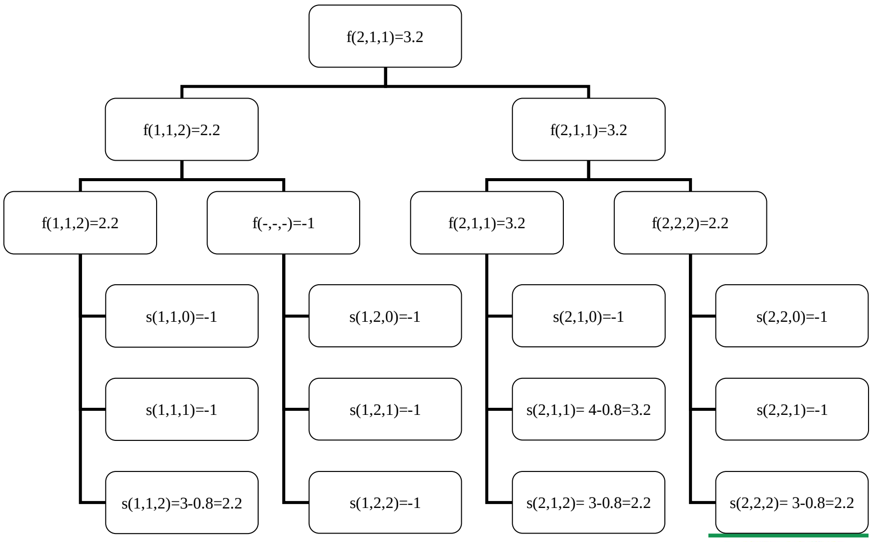
	0	1	2
0	0.6	-1	3
1	0.4	1.4	0.4

wavelenght (rows) — fiber F_2

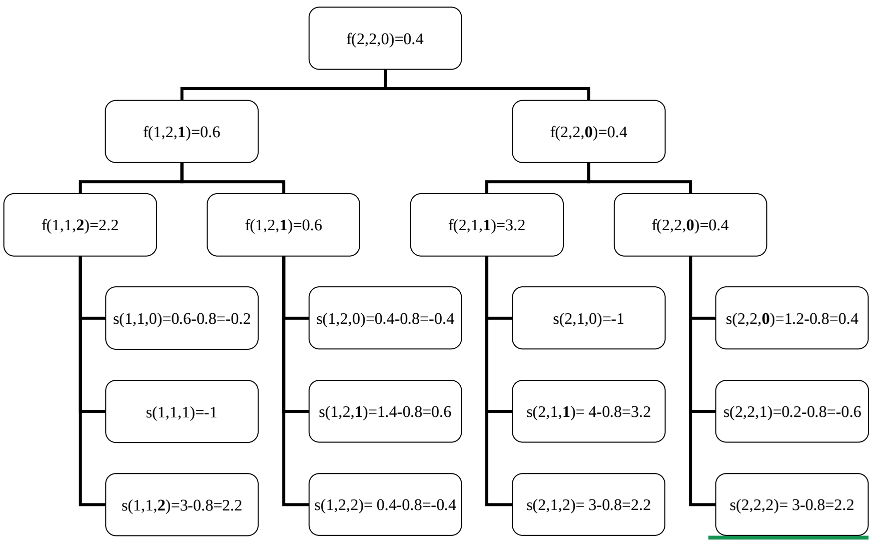
	0	1	2
0	-1	4	3
1	1.2	0.2	3

fiber F_1 *fiber* F_2

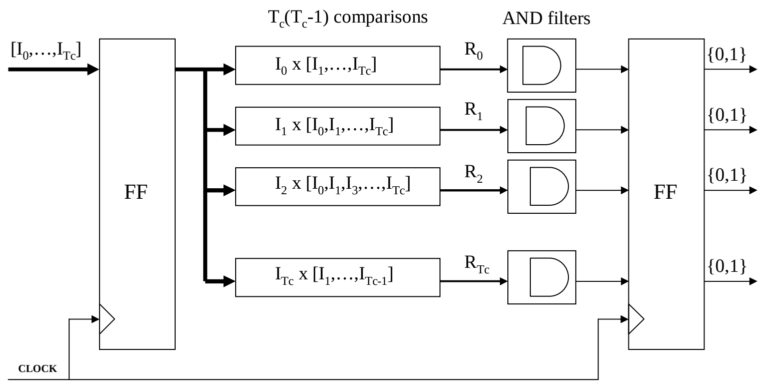
Example of tree, Min length, x=0.8



Example of tree, Min length-VF, x=0.8



Hardware search



Implementation Costs

- Cost depends on
  - Min/max search
  - Height of the tree
    - Depends to the performed algorithm
- Trade-off
  - Heigth vs. min/max hardware complexity

Open issues

- Buid matrix S in real-time
- Update matrix status $s(t_0)$ -> $S(t_1)$

Conclusion

- Not depending to the number of gaps
  - Fixed cost
- Subset of path (QoS)
- Not only search but algorithm oriented search
  - Min length, min gap
- Ready to be extended to parallel processing
  - Sub-optimal search