Modelli analitici: SPIW

SPIW: probabilità di perdita

- Modello analitico proposto può essere usato per valutare le prestazioni di questa architettura:
  - In questo caso i pacchetti sulla stessa lunghezza d'onda contendono fra loro per solo B TWC;
- Per la MS-B&S, solo P_{bwc} deve essere adattata, tutto il resto del modello è identico:

$$P_{loss} = P_u + P_b \cdot \left(1 - \frac{P_u}{P_b}\right) \cdot P_{bwc}$$

$$P_u = \sum_{h=M+1}^{N \cdot M} \left(1 - \frac{M}{h}\right) \binom{N \cdot M - 1}{h - 1} \left(\frac{p}{N}\right)^{h-1} \left(1 - \frac{p}{N}\right)^{N \cdot M - h}$$

$$P_b = \sum_{h=2}^{N} \left(1 - \frac{1}{h}\right) \binom{N - 1}{h - 1} \left(\frac{p}{N}\right)^{h-1} \left(1 - \frac{p}{N}\right)^{N-h}$$

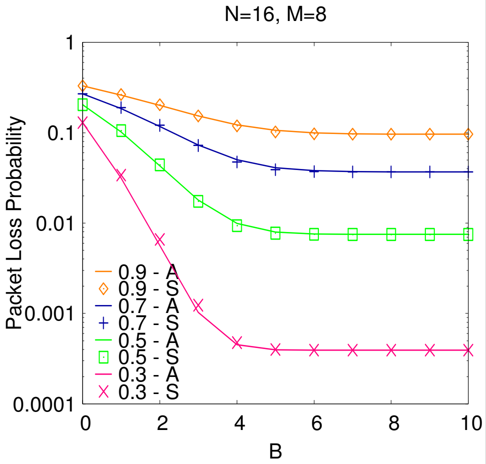
$$P_{bwc} = \sum_{h=B+1}^{N} \left(1 - \frac{B}{h}\right) \binom{N - 1}{h - 1} (A_{wc})^{h-1} (1 - A_{wc})^{N-h}$$

$\Downarrow$

fino a N pachetti contendono per B TWCs

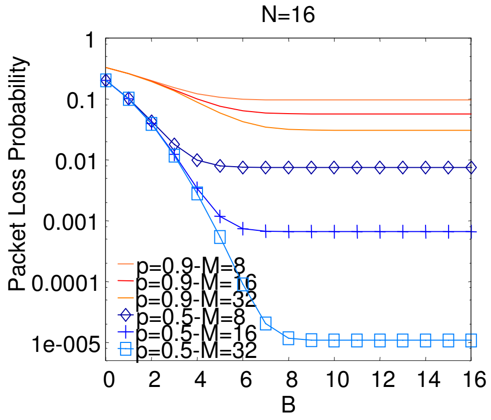
SPIW: model accuracy

- Simulation set-up:
  - N=16, M=8;
- Packet loss probability as a function of the number of TWC blocks is evaluated in according to different values of load per wavelength;
- Multistage architecture allows to save TWC blocks;
- Packet loss is high due to the lack of optical buffer;



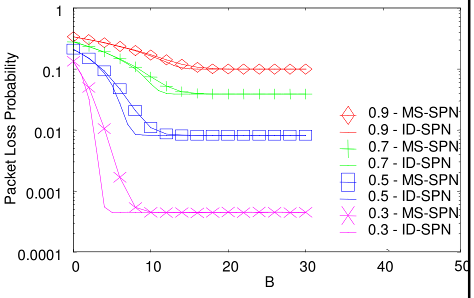
SPIW: packet loss evaluation

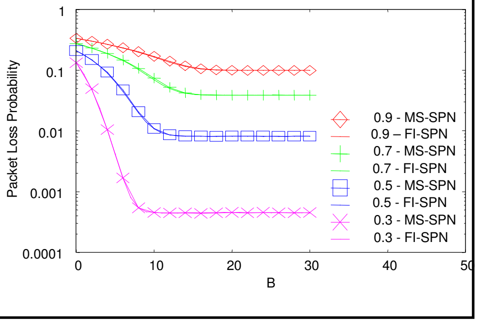
- Set-up: N=16, M=8-16-32;
- Packet loss probability as a function of number of TWC blocks varying number of wavelengths per fibre and load;
- When number of wavelengths increases, packet loss decreases;



Confronto fra architettura SPIW e architettura di riferimento (ID-SPN): prestazioni

- Stesso valore asintotico della PLP
- Con tunable-input TWCs, ID-SPN permette un risparmio maggiore in termini di TWC
- Con fixed-input TWCs, stesse prestazioni





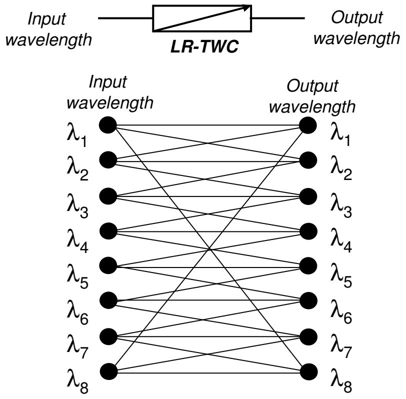
Modelli analitici: SPN limited range

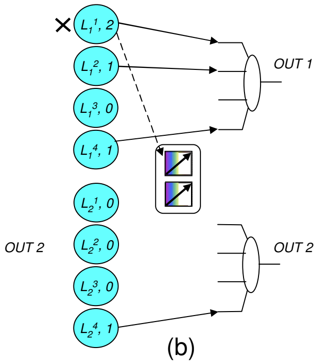
Utilizzo dei convertitori Limited Range per ulteriore risparmio

- Stesse architetture equipaggiate con convertitori limited range
- Il modello analitico proposto va esteso, non stravolto

Probabilita’ di perdita con limited range: modello analitico (1)

- Tre possibili cause di perdita del pacchetto
  - Fibra di uscita congestionata
  - Range di conversione limitato
  - Numero di convertitori insufficiente





(b)

Probabilita’ di perdita con limited range: modello analitico (2)

$$P_{loss} = P_u + P_b \left(1 - \frac{P_u}{P_b}\right) P_{inconv}$$

Probabilita’ che il pacchetto non possa essere convertito

dove:

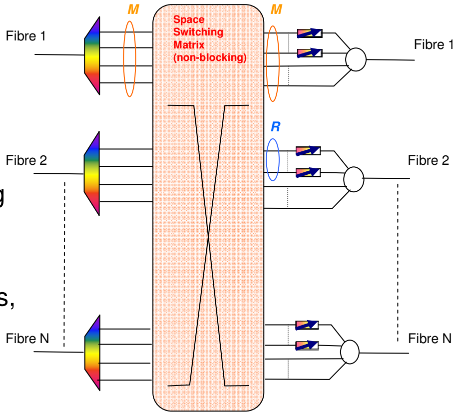
$$P_{inconv} = P_{lr} + (1 - P_{lr})\, P_{bwc}$$

e P_{lr} e’ la probabilita’ che il pacchetto sia perso dovuto al range di conversione limitato

Modelli analitici: SPL

Shared per output link architecture

- Each output link is provided with a dedicated set of R TWCs.
- The TWCs of the same set can be used only by packets addressed to the related output link.
- The optical switching fabric can be configured to forward a packet to a link with or without TWCs, depending on traffic conditions.
- Input packets are first considered for forwarding on the same wavelength on input. If channel contention occurs, wavelength conversion is performed.
- Not all packets need conversion at the same time slot, so some TWC saving can be achieved



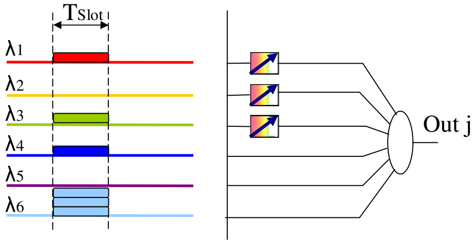
Packet loss probability

- Packet loss probability is evaluated on single output fibre “j”;
- Packets are lost when output fibre j is congested;
- Congestion on output fibre j occurs when:
  - Case 1: all wavelengths are busy, as well as full wavelength conversion;
  - Case 2: the number of packets that needs conversion is higher than the number of TWCs;

Loss probability: case 1

- G number of different wavelengths sending packets to output j
- M – R channels without TWCs
- Case 1: G ≥ M – R
  - M – R packets are transmitted to the output channels without TWC;
    - All channels without TWC are exploited
  - R packets are transmitted to the output channels with TWC;
  - Total packets transmitted: M, remaining packets are lost;
  - Number of transmitted packets is the same as in the full conversion case;

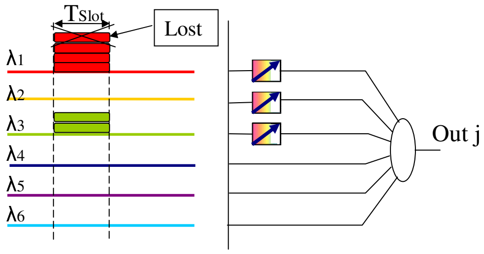
Example:
M=6,
R=3,
G = 4 > M – R;



Loss probability: case 2

- G number of different wavelengths sending packets to output j
- M – R channels without TWC
- Case 2: G < M – R
  - G packets are transmitted to the output channels without TWC;
  - R packets are transmitted to the output channels with TWC;
  - Total packets transmitted: G + R < M, remaining packets are lost.
- Not all channels without TWC can be exploited to transmit packets
  - loss is higher than in the full wavelength conversion case;

Example:
M=6,
R=3,
G = 2 < M – R;



Analytical model: variables

- p: arrival probability on a wavelength in a time slot;
- A_j: probability of an arrival for output fibre j;
- G: number of active wavelengths (with at least one packet for output j)
- D_G: probability of G active wavelengths;
- $\Gamma_{h|G}$: probability of h arrivals to output j given G active wavelengths
- N^j_l: number of lost packets on output j in a time slot;
- N^j_o: number of packets offered to output j in a time slot;
- P_l: packet loss probability.

Expression of packet loss probability P_l

- P_l can be expressed in relation to the generic output j by definition as the ratio between the averages of N^j_l and N^j_o :

$$P_l = \frac{E[N^j{}_l]}{E[N^j o]}$$

- Under the assumption of uniform traffic the average number of packets offered to output j in a time slot, N^j_o, is given by

$$E[N_o^j] = N \cdot M \cdot p \cdot \frac{1}{N} = M \cdot P$$

- and the average number of lost packets on output j in a time slot, N^j_l, is given by

$$E[N_l^j] = \sum_{G=1}^{M} E[N_l^j | G] \cdot D_G$$

  - where

G = 1..M

$$D_G = \binom{M}{G} A_j^G (1 - A_j)^{M-G}$$

Average number N^j_l of lost packets on output j

- N^j_l is calculated by taking into account the contributions of case 1 and case 2

$$E[N_l^j]=\sum_{G=1}^{M-R-1}\sum_{h=G+R+1}^{NG}(h-(G+R))\cdot\Gamma_{h|G}D_G+\sum_{G=M-R}^{M}\sum_{h=M+1}^{NG}(h-M)\cdot\Gamma_{h|G}D_G$$

- Expressions of $\Gamma_{h|G}$ for G=0 and G=1
  - $$\Gamma_{h|0}=\begin{cases}1 & h=0\\ 0 & h>0\end{cases}\qquad \Gamma_{h|1}=\begin{cases}\dfrac{\binom{N}{h}\left(\frac{p}{N}\right)^h\left(1-\frac{p}{N}\right)^{N-h}}{1-\left(1-\frac{p}{N}\right)^N} & 0<h\le N\\ 0 & h=0,h>N\end{cases}$$

where $1-\left(1-\frac{p}{N}\right)^N$ represents the normalizing factor representing the probability of h>0

Recursive formula for $\Gamma_{h|G}$

- When $G=2$

$$\Gamma_{h|2}=\sum_{x=1}^{\min(N,h-1)}\Gamma_{x|1}\cdot\Gamma_{(h-x)|1}=\sum_{x=1}^{\min(N,h-1)}\frac{\binom{N}{x}\left(\frac{p}{N}\right)^x\left(1-\frac{p}{N}\right)^{N-x}}{1-\left(1-\frac{p}{N}\right)^N}\cdot\frac{\binom{N}{h-x}\left(\frac{p}{N}\right)^{h-x}\left(1-\frac{p}{N}\right)^{N-(h-x)}}{1-\left(1-\frac{p}{N}\right)^N}$$

- When $2\le G\le M$

$$\Gamma_{h|G}=\begin{cases}\displaystyle\sum_{x=1}^{\min(N,h-G+1)}\Gamma_{x|1}\cdot\Gamma_{(h-x)|(G-1)} & G\le h\le G\cdot N\\ 0 & h<G,h>G\cdot N\end{cases}$$

Packet Loss Probability

$$P_{Loss}=\underbrace{\frac{1}{M\cdot P}\cdot\sum_{G=1}^{(M-R)-1}\sum_{h=G+R+1}^{N\cdot G}\left(h-(G+R)\right)\cdot\Gamma_{h|G}\cdot D_G}_{G<M-R}+\underbrace{\frac{1}{M\cdot P}\cdot\sum_{G=M-R}^{M}\sum_{h=M+1}^{N\cdot G}(h-M)\cdot\Gamma_{h|G}\cdot D_G}_{G\geq M-R}$$

$$D_G=\left(\frac{G!}{M!\cdot(M-G)!}\right)\cdot\left(1-\left(1-\frac{P}{N}\right)^N\right)^G\cdot\left(\left(1-\frac{P}{N}\right)^N\right)^{M-G}$$

- D_G is the probability that G different input wavelengths send packets to output fibre j;
- $\Gamma_{h|G}$ is the probability of h packet arrivals, given that G different input wavelengths send packets to output fibre j;

$$\Gamma_{h|G}=\begin{cases}0 & h<G\\ \sum_{x=1}^{h-(G-1))}\Gamma_{x|1}\cdot\Gamma_{(h-x)|(G-1)} & G<h\leq N\\ \sum_{x=1}^{N}\Gamma_{x|1}\cdot\Gamma_{(h-x)|(G-1)} & N<h\leq G\cdot N\\ 0 & h>G\cdot N\end{cases}$$

$$\Gamma_{h|1}=\frac{\left(\frac{h!}{N!\cdot(N-h)!}\right)\cdot\left(\frac{P}{N}\right)^h\cdot\left(1-\frac{P}{N}\right)^{N-h}}{1-\left(1-\frac{P}{N}\right)^N}\qquad 0<h\leq N$$

Packet loss probability: special cases

- Full wavelength conversion

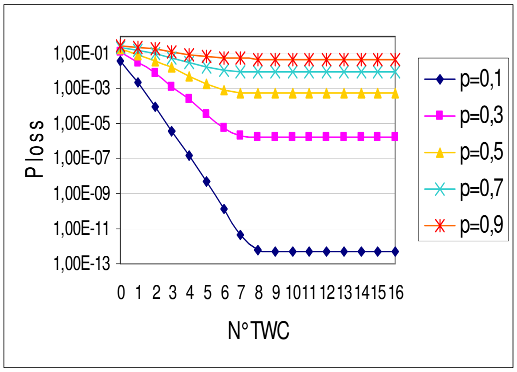
$$P_l=\frac{1}{Mp}\sum_{G=0}^{M}\sum_{h=M+1}^{NG}(h-M)\cdot\Gamma_{h|G}\cdot D_G$$

- No wavelength conversion

$$P_l=\frac{1}{Mp}\sum_{G=1}^{M}\sum_{h=G+1}^{NG}(h-G)\cdot\Gamma_{h|G}\cdot D_G$$

Packet loss probability

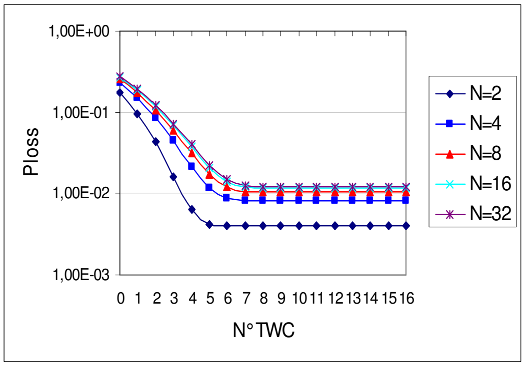
- Simulation set-up:
  - N=8, M=16;
- Packet loss probability as a function of the number of TWCs is evaluated in according to different values of load per wavelength;



- The results show the minimum number of wavelength converters needed to achieve the same loss of full wavelength conversion;

Packet loss probability

- Simulation set-up:
  - M=16, P=0.7;
- Packet loss probability as a function of the number of TWCs is evaluated in according to different switch size;



- Packet loss probability is slightly depend on switch size;

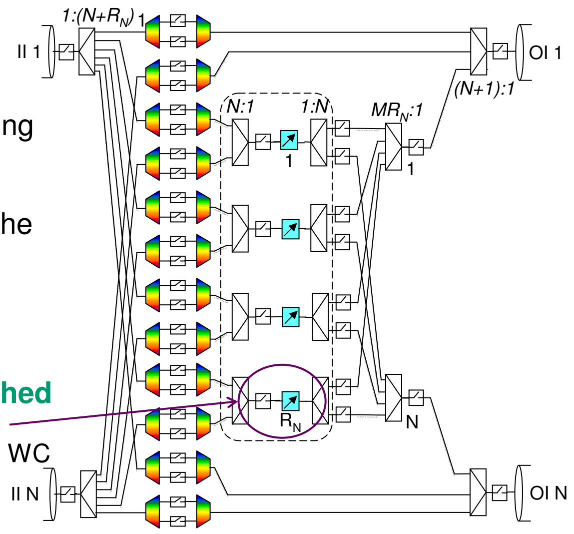
Esercitazione di laboratorio
Simulazione di diverse architetture
SPN,SPIW,SPOW,SPL

Practical architectures

- The proposed sharing schemes must be realized taking available optical technology into account
  - spitters/coupler, MUX/DEMUX, switching gates like Semiconductor Optical Amplifiers (SOA) switch…
- Different implementations can be proposed, based on broadcast-&-Select (B&S), wavelength routing (using Arrayed Waveguide Gratings, AWGs), space diversity and so on
- B&S solutions based on optical gates (SOA or MEMS technologies as example) are presented

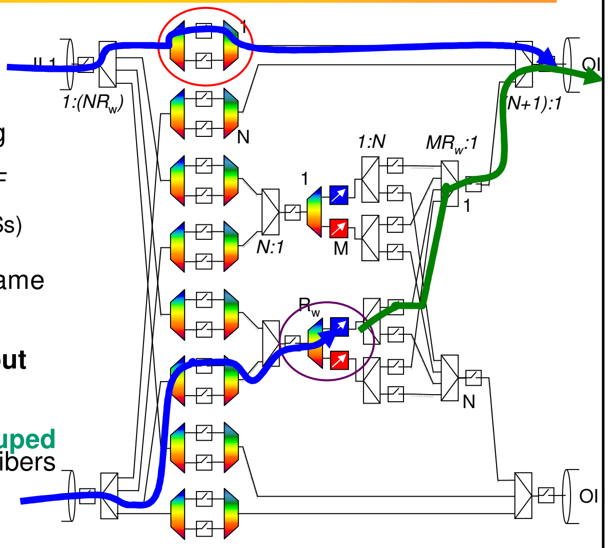
SPN Architecture

- **N** input/output fibers, **M** wavelengths
- B&S SOA-based switching matrix
- WCs shared among all the input channels
- **R** TWCs
- Each **WC must be reached by all the input fibers**
    - N WSs to reach a single WC



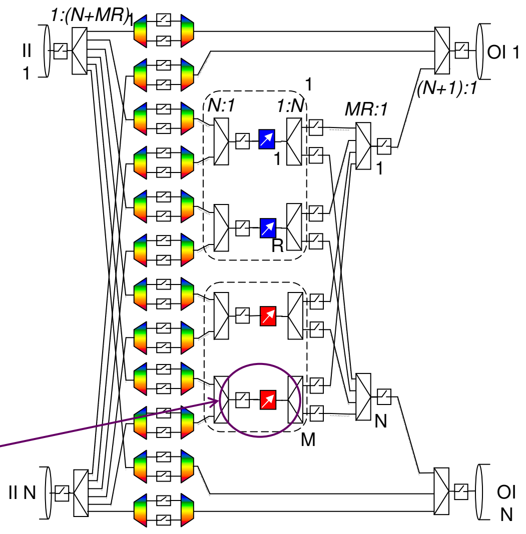
SPIW Architecture

- **N** input/output fibers, **M** wavelengths
- B&S SOA-based switching matrix
    - SOA employed as ON/OFF gates
    - Wavelength Selectors (WSs)
- **R** WCs dedicated to the same input wavelength
- **Fixed-input/tunable output** WCs
- Allows **M WCs to be grouped** and reached by the input fibers in a simple way
    - N WSs to reach M WCs



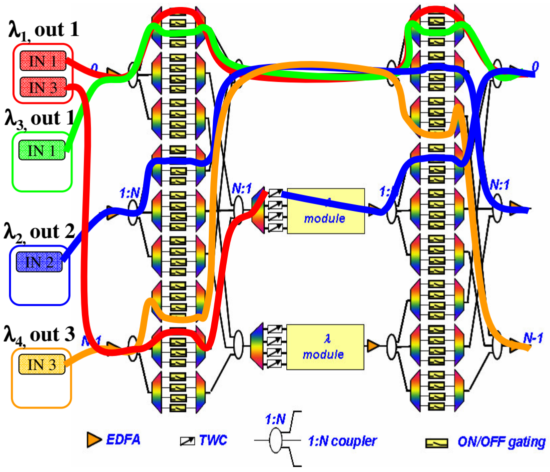
SPOW architecture

- **N** input/output fibers, **M** wavelengths
- SOA-based switching matrix
- WCs **shared among the same output wavelength**
- **R_w** WCs dedicated to the same output wavelength
- **Tunable-input/fixed-output** WCs
  - simpler devices
- Each WC must be reached by all the input fibers
  - N WSs to reach a single WC



MS-SPIW architecture (multistadio)

- Step 1: pacchetti in ingresso sulla stessa lunghezza d'onda e diretti alla stessa uscita sono raggruppati
  - Contendono per lo stesso canale di uscita;
  - Pacchetti in gruppi differenti sono contention free
- Step 2: un pacchetto per ogni gruppo e' spedito senza conversione, altri pacchetti vanno convertiti
- Massimizza il numero di pacchetti spediti senza conversione



Architetture considerate

- Architettura SPL (shared per link):
  - convertitori di lunghezza d'onda condivisi fra pacchetti diretti alla stessa fibra di uscita
  - simulatore: **SPL.c**
- Architettura SPN (shared per node):
  - convertitori di lunghezza d'onda condivisi fra tutti i pacchetti
  - simulatore: **SPN.c**
- Architettura SPIW (shared per input wavelength):
  - convertitori di lunghezza d'onda condivisi fra i pacchetti in ingresso sulla stessa lunghezza d'onda
  - simulatore: **SPIW.c**
- Architettura SPOW (shared per output wavelength):
  - convertitori di lunghezza d'onda condivisi fra i pacchetti spediti in uscita sulla stessa lunghezza d'onda
  - simulatore: **SPOW.c**
- Architettura SPIW multistadio (MS-SPIW):
  - architettura modulre con convertitori di lunghezza d'onda condivisi fra i pacchetti in ingresso sulla stessa lunghezza d'onda
  - simulatore: **MS-SPIW.c**

Simulatori: considerazioni generali

- Simulatori permettono di calcolare il valor medio della probabiltà di perdita per le architetture di commutazione considerate
- Valor medio della perdita dipende dal carico e dal numero di convertitori utilizzati
- In tutti e 3 i simulatori possono essere eseguite simulazioni in cascata variando il numero di convertitori e il carico per lunghezza d'onda
- E' possibile impostare il valore desiderato per l'intervallo di confidenza (es: intervallo di confidenza inferiore al 5% del valor medio della perdita con probabilità 95%)
  - si eseguono più simulazioni e si calcolano valor medio, varianza e intervallo di confidenza

Struttura dei simulatori

```
main   {
    definizione variabili;
    inizializzazione variabili;
    acquisizione parametri da linea di comando;
    while (N° conv min < N° conv attuale < N° conv max)  {
        while (carico min < carico attuale < carico max)                {
            while (numero di pacchetti è inferiore a quello desiderato)  {
                //  simulo un time slot
                inizializzazione hardware architettura;
                generazione degli arrivi del time slot corrente;
                applicazione algoritmo di scheduling per time slot corrente;
            }
        }
    }
}
```

Simulatori: procedure in comune (1)

- Tutti e 3 i simulatori contengono delle procedure 'comuni', per la generazione di eventi pseudo-casuali
- Procedura ***Ran2()***: si occupa della generazione di una sequenza pseudo-casuale di numeri
  - ogni volta che viene richiamata restituisce un numero compreso nell'intervallo 0 e 1 (1 escluso)
  - viene utilizzata dalle altre procedure che generano valori casuali (random)
- Procedura ***Genera_out()***: restituisce una fibra di destinazione scelta a caso fra le N possibili (utilizza Ran2)
- Procedura ***Genera_arrivo()*** : restituisce la probabilità di presenza (p)/assenza (1-p) di un pacchetto su un canale di ingresso (utilizza Ran2)

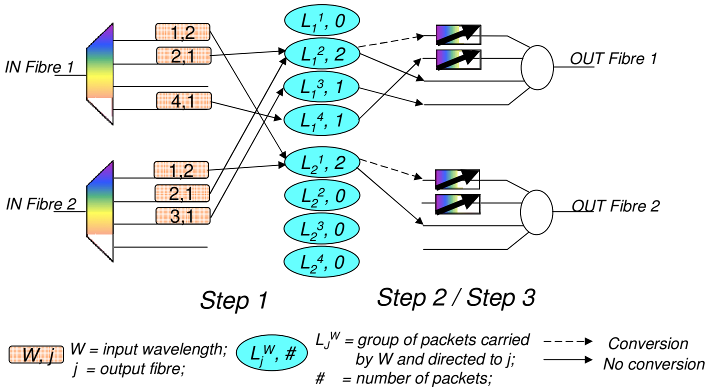
Simulatori – main (1): acquisizione dati

- Tutti i simulatori hanno la stessa organizzazione, l’impalcatura è la stessa
- In ogni simulazione vengono anzitutto eseguite queste parti in comune a tutti i simulatori
  - definizione delle variabili
  - acquisizione dei dati in ingresso tramite la procedura ***acquisisci_dati()***
    - si occupa di acquisire da linea di comando i parametri da utilizzare nel corso della simulazione
  - inizializzazione delle variabili

Simulatori – main (2): inizializzazione architettura e generazione dei pacchetti

- Vengono simulati time slot fino a quando il numero di pacchetti generati non raggiunge il valore desiderato
- In ogni time slot eseguo
  - inizializzazione hardware dell’ architettura con funzione ***inizializza_hardware()***
    - inizializza come liberi tutti i convertitori, le lunghezze d’onda di uscita e ogni altro dispositivo hardware all’inzio di ogni time slot
  - generazione degli arrivi per il time slot corrente utilizzando la funzione ***Genera_arrivi_Tslot()***
    - si occupa della generazione dei pacchetti su tutte le lunghezze d’onda di ogni fibra di ingresso all’inizio di ogni time slot (utilizza Genera_arrivo e Genera_out)

Simulatori – main (3) : algoritmo di scheduling



- Step 1: le fibre di ingresso sono scandite in maniera sequenziale e vengono formati dei gruppi di pacchetti arrivati sulla stessa lunghezza d'onda e diretti alla stessa fibra di uscita

- Step 2: i pacchetti vengono "schedulati" tenendo conto del gruppo a cui appartengono (un pacchetto da ogni set può essere spedito senza conversione)

Simulatori - main (4): algoritmo di scheduling

- Lo scheduling dei pacchetti utilizza la funzione ***sel_lambda(*** λ_{in}***, out)*** (si differenzia da architettura ad architettura)

- Procedura ***Sel_lambda(***λ_{in}***, out)***: procedura che effettua lo scheduling dei pacchetti. Viene richiamata ogni volta che c'è presenza di un pacchetto su un canale di ingresso; vengono passati la lunghezza d'onda su cui il pacchetto arriva e la sua fibra di destinazione.
  - la procedura calcola la lunghezza d'onda su cui spedire il pacchetto nella fibra di destinazione tenendo conto dello stato di occupazione delle fibre e del numero di convertitori disponibili
  - se il pacchetto non può essere spedito restituisce un valore nullo (-1)

- Questa procedura dipende dall'architettura considerata per cui ognuno dei 3 simulatori ha una sua sel_lambda()

Procedura *Sel_lambda(λ_{in}, out)* per SPL.c

```
/* SPEDISCO IL PACCHETTO SENZA CONVERSIONE SE POSSIBILE */
If ($\lambda_{in}$ è libera nella fibra di destinazione)  {
    if (c'è almeno un link ottico libero sulla interfaccia di uscita out)
        occupo il link utilizzato
        occupo $\lambda_{in}$ nella fibra di uscita
        restituisco $\lambda_{in}$
    }
    else (se c'è almeno un convertitore libero)    {
        occupo il convertitore utilizzato
        occupo $\lambda_{in}$ nella fibra di uscita
        restituisco $\lambda_{in}$
    }
}
else   /* SPEDISCO IL PACCHETTO SFRUTTANDO LA CONVERSIONE */
    if (c'e una $\lambda_z$ libera nella fibra out e c'è un convertitore libero nell'interfaccia out)   {
        occupo il convertitore utilizzato
        occupo $\lambda_z$ nella fibra di uscita
        restituisco $\lambda_z$
    }

Restituisco -1   /* PACCHETTO PERSO */
```

Procedura *Sel_lambda(λ_{in}, out)* per SPN.c

```
/* spedisco il pacchetto senza conversione */
If ($\lambda_{in}$ è libera nella fibra di destinazione)  {
    occupo il link utilizzato
    occupo $\lambda_{in}$ nella fibra di uscita
    restituisco $\lambda_{in}$
}
else   /* spedisco il pacchetto sfruttando la conversione */
    if (c'e una $\lambda_z$ libera nella fibra out e c'è un convertitore libero)   {
        occupo il convertitore utilizzato
        occupo $\lambda_z$ nella fibra di uscita
        restituisco $\lambda_z$
    }

Restituisco -1   /* PACCHETTO PERSO */
```

Procedura *Sel_lambda(λ_{in}, out)* per MS-SPIW.c

```
/* spedisco il pacchetto senza conversione */
If ($\lambda_{in}$ è libera nella fibra di destinazione)   {
      occupo $\lambda_{in}$ nella fibra di uscita
      occupo il link utilizzato
      restituisco $\lambda_{in}$
}
else    {     /* spedisco il pacchetto sfruttando la conversione */
      if (c'è una $\lambda_z$ libera nella fibra out e un convertitore dedicato a $\lambda_{in}$)   {
            occupo il convertitore all'interno del blocco dedicato a $\lambda_{in}$
            occupo $\lambda_z$ nella fibra out e a monte del blocco di convertitori
            restituisco $\lambda_z$
      }
}

Restituisco   -1   /* PACCHETTO PERSO */
```

SIMULATORI: SPL.c (1)

- Simulatore architettura SPL: **SPL.c**
  - Dati richiesti in ingresso:
    - Numero di fibre di IN/OUT
    - Numero di lunghezze d'onda per fibra
    - Numero minimo di convertitori per simulazioni in cascata
    - Numero massimo di convertitori per simulazioni in cascata
    - Incremento del numero di convertitori fra 2 simulazioni
    - Valore minimo del carico per simulazioni in cascata
    - Valore massimo del carico per simulazioni in cascata
    - Valore dell'intervallo di confidenza da considerare (se si decide di calcolarla)
    - Numero di simulazioni da considerare per l'intervallo di confidenza
    - Numero di pacchetti da simulare

SIMULATORI: SPL.c (2)

- File di output SPL.c
  - **Risultati_SPL.txt**: contiene un riassunto dei risultati ottenuti per tutte le simulazioni effettuate
  - **Ploss_SPL.txt**: contiene la probabilità di perdita per tutte le simulazioni effettuate al variare del carico (x) e del numero di convertitori (y)
  - **Utilconv_SPL.txt**: contiene l'utilizzazione media dei convertitori in un time slot al variare del carico (x) e del numero di convertitori (y)
  - **Utillinee_SPL,txt**: contiene l'utilizzazione media dei link privi di convertitori in un time slot al variare del carico (x) e del numero di convertitori (y)

SIMULATORI: SPN.c (1)

- Simulatore architettura SPN: **SPN.c**
  - Dati richiesti in ingresso:
    - Numero di fibre di IN/OUT
    - Numero di lunghezze d'onda per fibra
    - Numero minimo di convertitori per simulazioni in cascata
    - Numero massimo di convertitori per simulazioni in cascata
    - Incremento del numero di convertitori fra una simulazione e l'altra
    - Valore minimo del carico per simulazioni in cascata
    - Valore massimo del carico per simulazioni in cascata
    - Probabilità di perdita su cui calcolare l'intervallo di confidenza
    - Valore dell'intervallo di confidenza da considerare (se si decide di calcolarla)
    - Numero di simulazioni da considerare per l'intervallo di confidenza
    - Numero di pacchetti da simulare

SIMULATORI: SPN.c (2)

- File di output SPN.c
  - **Risultati_SPN.txt**: contiene un riassunto dei risultati ottenuti per tutte le simulazioni effettuate
  - **Ploss_SPN.txt**: contiene la probabilità di perdita media per tutte le simulazioni effettuate al variare del carico (x) e del numero di convertitori (y)
  - **Plossconv_SPN.txt**: contiene la probabilità di perdita dovuta alla mancanza di convertitori per tutte le simulazioni effettuate al variare del carico (x) e del numero di convertitori (y)
  - **Plossout_SPN.txt**: contiene la probabilità di perdita dovuta a contesa nelle fibre di uscita per tutte le simulazioni effettuate al variare del carico (x) e del numero di convertitori (y)
  - **Utilconv_SPN.txt**: contiene l'utilizzazione media dei convertitori in un time slot al variare del carico (x) e del numero di convertitori (y)
  - **Utillinee_SPN,txt**: contiene l'utilizzazione media dei link privi di convertitori in un time slot al variare del carico (x) e del numero di convertitori (y)

Esercitazioni guidate in laboratorio (1)

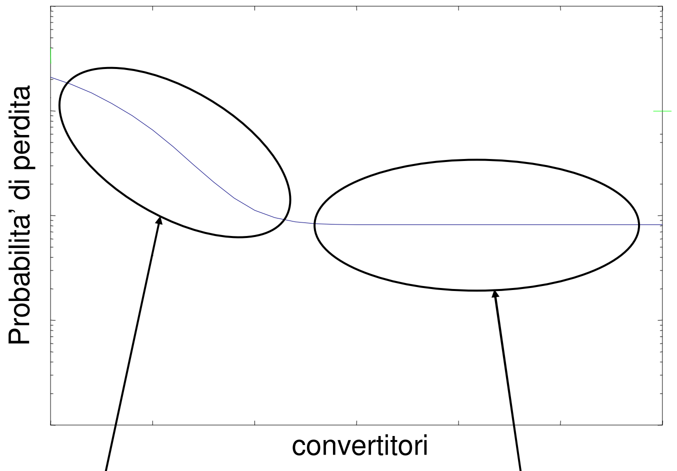
- Confronto fra le architetture
  - Valutazione delle prestazioni in termini di probabilità di perdita
  - Confrontare su uno stesso grafico la perdita per le due architetture
  - Valutare quale architettura fornisce le migliori prestazioni a parità di carico e di numero (totale) di convertitori

Esercitazioni guidate in laboratorio (3)

- Confronto fra i tempi di simulazione
  - In funzione del carico
  - In funzione delle dimensioni del commutatore
  - A parita’ di precisione.

- Confronto tra tempo di simulazione e tempo di esecuzione del modello
- Confronto analisi/simulazione

Probabilita’ di perdita in funzione dei convertitori



perdita aggiuntiva dovuta a mancanza di convertitori

Valore asintotico: dovuto a perdita per blocco in uscita

Complexity (Number of Optical Devices)

- Each sharing scheme requires a different organization of the optical devices
- Different architectures must be compared in terms of complex (and expensive) optical devices employed
- Here, we consider WCs and SOAs, which are active components
- Complexity in terms of WCs and SOAs employed

	total # WCs	**total # SOAs (switching purposes)**
SPN	$C_{SPN} = R$	$G_{SPN} = N(NM + (M + 1)C_{SPN})$
SPIW	$C_{SPIW} = MR$	$G_{SPIW} = N(NM + 2C_{SPIW})$
SPOW	$C_{SPOW} = MR$	$G_{SPOW} = N(NM + MC_{SPOW})$
SPL	$C_{SPL} = NR$	$G_{SPL} = N(NM + MC_{SPL})$

SPIW requires less SOAs when architectures equipped with the same number of WCs

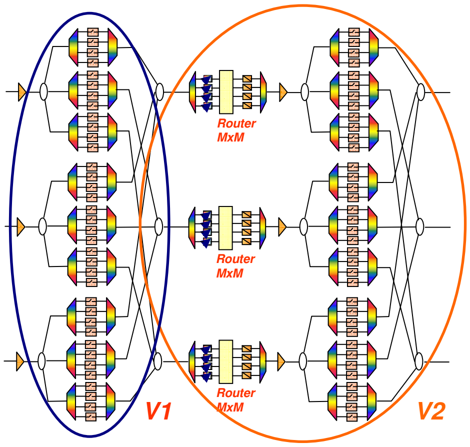
---

Advantages and Disadvantages of Sharing Architectures

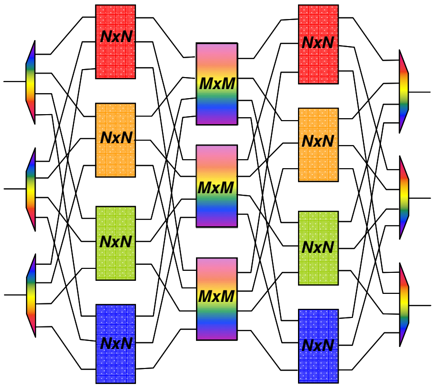
	# WCs	***Kind of WCs***	***#SOAs***	***performance/ complexity trade-off***
SPN	lowest	TWCs (complex)	highest	fair
SPL	high (especially when N high, M low)	TWCs (complex)	near to SPN with some save	not good
SPIW	good (when N high, M low) fair in other cases	FTWCs (easier than TWCs)	lowest	good (when N high, M low)
SPOW	low (near to SPN)	FWCs (easiest)	near to SPN with some save	good

Architettura MS-SPN ed equivalente spaziale

- N=3, M=4;
- Architettura V1 e V2 in cascata;

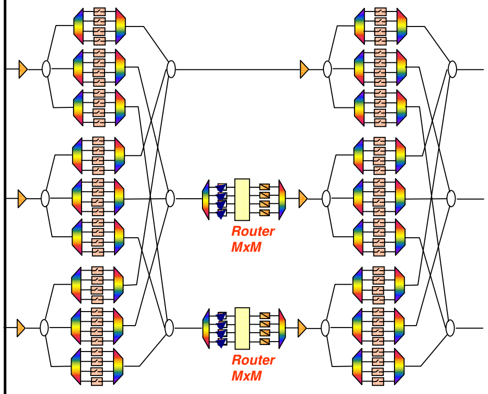


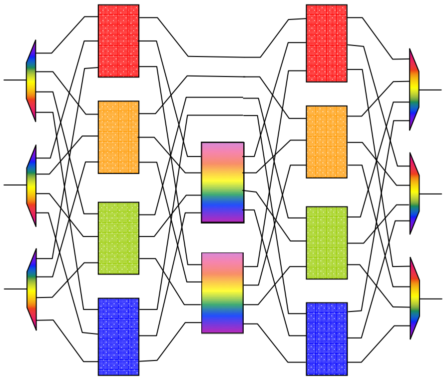
- E' ancora riarrangiabile non bloccante come la V2, ma permette di realizzare la condivisione dei convertitori fra gli ingressi (shared per node), mentre la V2 è shared per link);



Architettura MS-SPN con conversione parziale

- N=3, M=4;
- Non + non blocc riarrangiabile;



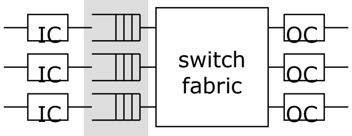


Buffered Architectures

Contention resolution in the time domain

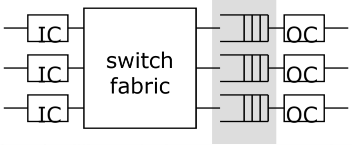
- Queues are used to delay optical packet forwarding when the contended resource is available
- Optical queues are obtained by means of fibre delay lines (FDL)
- The maximum packet delay is limited and related to the FDL length
- Different queuing schemes can be adopted

Buffer placement at the node



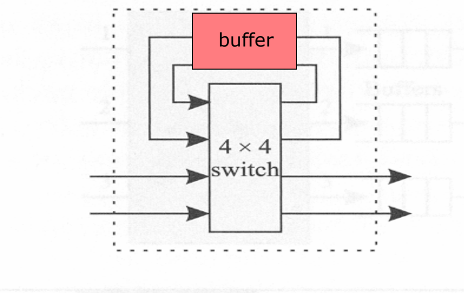
Input buffer

- Simple, FIFO
- Head of the line (HOL) blocking



Output buffer

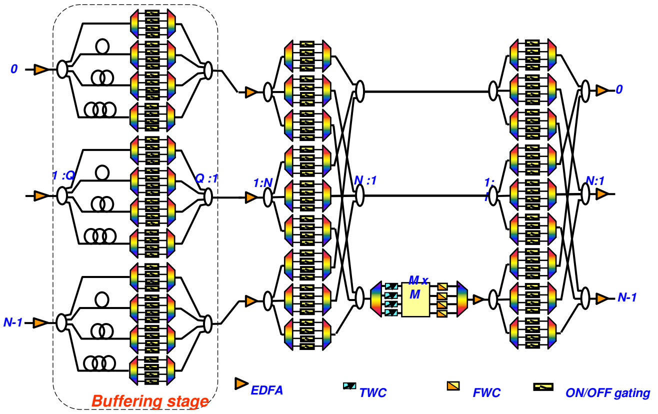
- No HOL blocking
- More difficult to implement



Recirculation buffers

- Shared by all inputs
- Requires larger switch size

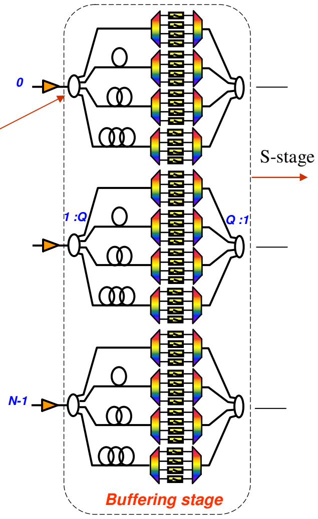
The multi-stage architecture with input buffers



- A queuing stage based on the broadcast and select principle is added
- The behaviour of the S-λ-S sub-system is the same as before

The queueing stage

- A power coupler generates Q copies of the multi-wavelength bundle of channels
  - D = k T, k=0..Q-1; T is the packet time
- Each FDL is followed by a wavelength selector
  - Optical packets are available at the first S-stage with all possible delays D
  - Input queuing scheduling can be applied



Queue scheduling

- Slot by slot operation
- Packets on the same input channel are organized in a list
- FIFO
  - Lists are served on a FIFO basis
  - Round robin service among lists
- Window
  - Allow to overcome the HOL phenomenon
  - If packets are present in the list, within a window of size W, which belong to a contention-free path, they are considered for forwarding

Scheduling of packet forwarding through the switch aims at minimizing wavelength conversion

Queue Scheduling pseudo code (FIFO)

```
step 1:
for (k=0; k < N; k++)
  for (j=0; j < M; j++)
    if (there is a packet carried by $\lambda_j$ on input fiber k)
      the packet is inserted on the tail of list $L_k^j$;

step 2:
for (k=0; k < N; k++)
  for (j=0; j < M; j++)  {
    in=(k+RRF) mod N;
    $\lambda$=(j+RRW) mod M;
    if ($L_{in}^{\lambda}$ != NULL)  {
      out=output fiber of the packet on the head of $L_{in}^{\lambda}$;
      tx=sel_lambda($\lambda$, out);
      if (tx>0)
        the packet is forwarded and removed from the head of $L_{in}^{\lambda}$;
      if ( (tx== -1) && (T - $C_{in}^{\lambda}$ == Q -1) )
        the packet is removed from the head of $L_{in}^{\lambda}$;
    }
  }
RRF=(RRF+1) mod N;
if (RRF==0)     RRW=(RRW+1) mod M;
```

Step 1: lists are formed for each channel

Step 2:

a packet is extracted from the head of each list (FIFO)

RRF and RRW are counters that assure fairness

Sel_lambda is a procedure to find a path through the switch

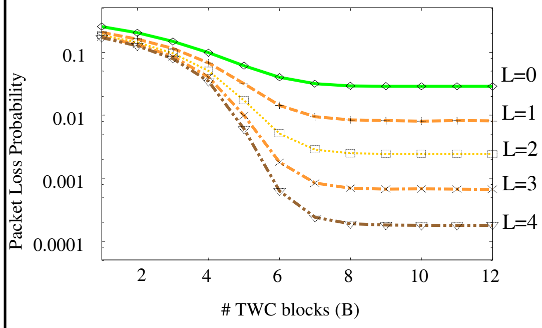
Quality of service

- Input queuing allows also quality of service differentiation in the node
- A simple QoS algorithm is considered to manage two QoS classes
  - The scheduling algorithm considers high priority packets first, if present in the list
  - Low priority packets are then searched for, otherwise

Simulation set up

- Bernoulli independent arrivals on input channels
- Uniform addressing scheme
- FIFO-RR and W-RR scheduling
- Confidence interval at 95%
  - Less than or equal to the 5% of the mean

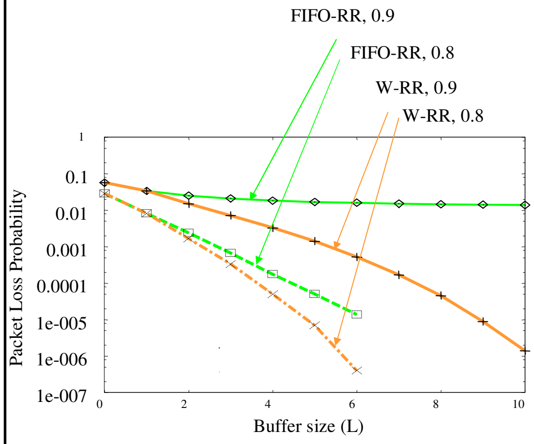
Packet loss improvement



- N=16, M=16
- FIFO-RR
- p=0.8
- L buffer size
- Remarkable improvement when internal block is overcome (asymptotic region)

Output contention resolution in time domain

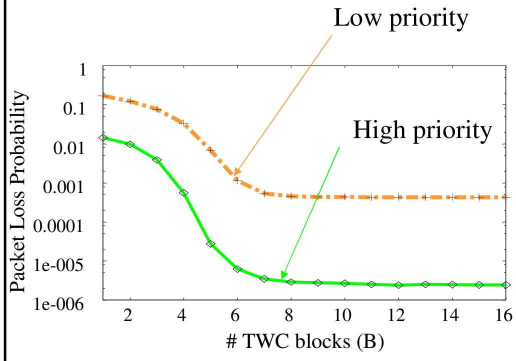
Effects of the queue scheduling



- N=16, M=16
- B=N
- FIFO-RR,W-RR
- p = 0.8 and 0.9
- The benefit becomes more evident as buffer length increases

---

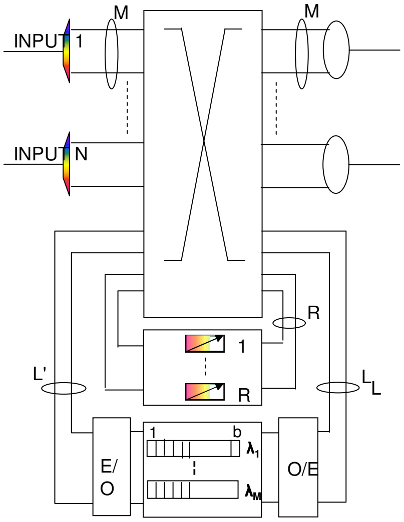
Quality of service aspects



- N=16, M=16
- L=5
- p=0.8
- Remarkable QoS differentiation even with low buffer values

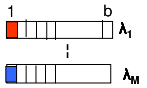
Single stage with recirculation shared buffer

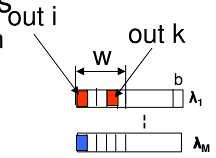
- N input/output fibers
- M wavelengths per fiber
- Strictly non-blocking space switching fabric
  - (NM+R+L') x (NM+R+L)
- R Full Range Tunable Wavelength Converters (TWCs) shared per node
- Queuing stage with M queues (one per wavelength), with size b
  - FDLs
  - ***Electronic buffers***



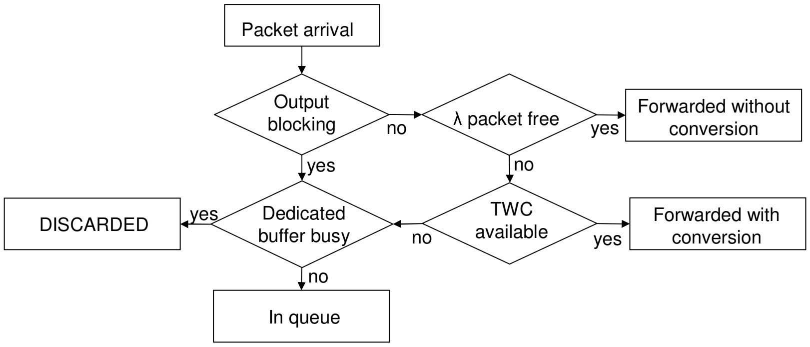
Queuing policies

- Hp: In each time slot, packets in the queues are served before packets from input fibers
- Two policies for buffer management:
  - FIFO: only the Head Of Line (HOL) packets (up to M) are served
  - Windowing: w packets in each queue can be forwarded, if directed to different output fibers
    - Window size: w=b, up to M*max(b, N) packets can be forwarded





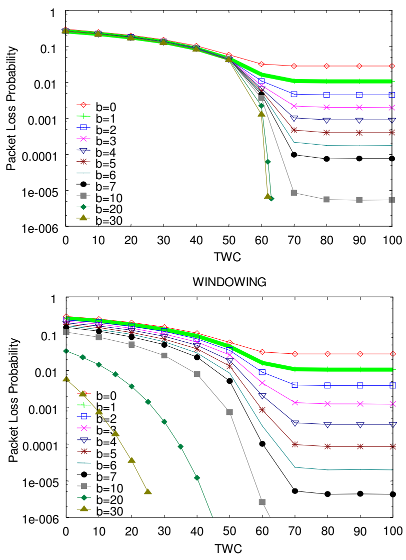
Packet forwarding


Packet arrival
Output blocking
no
λ packet free
yes
Forwarded without conversion
yes
no
DISCARDED
yes
Dedicated buffer busy
no
TWC available
yes
Forwarded with conversion
no
In queue

- The packet is sent without conversion, if possible
- Otherwise the packet is sent with conversion
- If output fiber is congested or no TWC is available, the packet is stored in the queue
- The packet is lost when its queue is busy

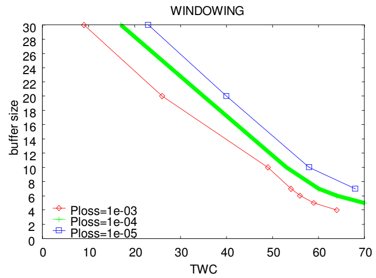
Packet loss probability

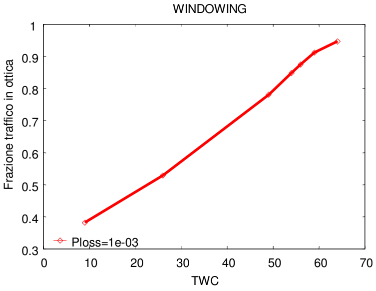
- Ploss = *f*(TWC), varying buffer size b,
- N=16, M=16, L=L'=256, p=0.8
- FIFO: improvement when buffer size increases only if enough TWCs are available
  - Due to the saturation of the queues
- Windowing: relevant improvement of the throughput in all regions
  - Small number of TWCs can be enough to assure low packet loss



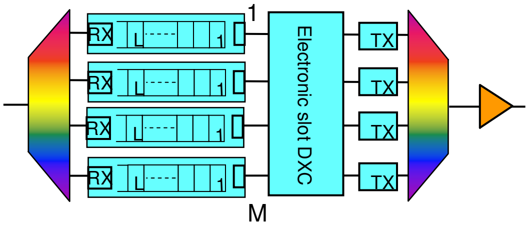
Switch dimensioning

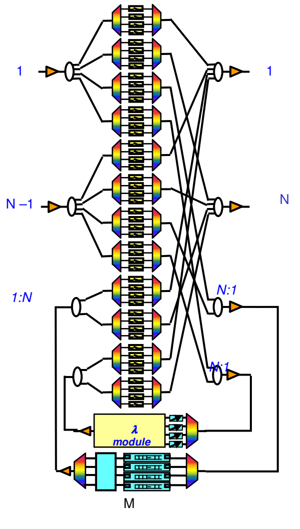
- b=*f*(TWC), to obtain a target PLP
- N=16, M=16, L=256, p=0.8
- Different couples (b, TWC) can be chosen to obtain the same packet loss
- The percentage of packets forwarded without O/E conversion decreases as the number of TWCs decreases
- If a certain percentage of packets must be forwarded in optics, a certain number of TWCs must be employed





Practical implementation of the hybrid switch

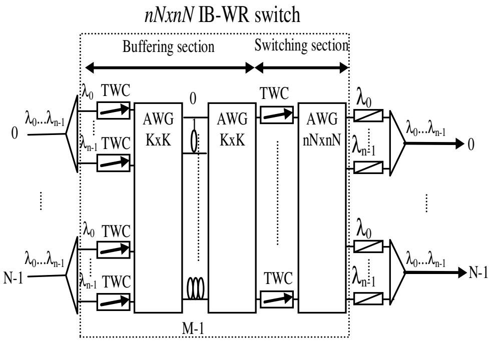




- Each buffer is used by the same wavelength

All optical AWG based buffered switch

- The first TWC stage is used to choose the FDL delay
- The second TWC stage is used to choose the output interface
- Good performance
- Low scalability in relation to TWC tuning range



Esercitazione di laboratorio
Architetture di commutazione con buffer ottici e architetture ibride opto-elettroniche

Ing. Michele Savi
DEIS - Universita'di Bologna

michele.savi@unibo.it

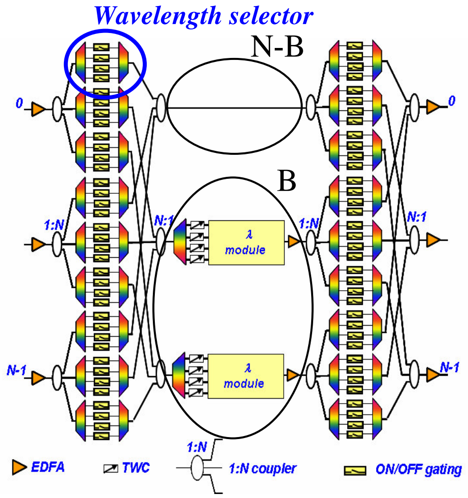
Architetture di commutazione con bufferizzazione

- Permettono di risolvere la contesa nel dominio temporale, in aggiunta al dominio della lunghezza d'onda
- Buffer completamente ottici
  - Fiber Delay Lines (FDLs)
    - Spezzoni di fibra che consentono la memorizzazione di un pacchetto per un tempo prefissato
    - Il pacchetto (payload) rimane nel dominio ottico per tutto il percorso all'interno dell'architettura
- Buffer elettronici (architettura ibrida opto-elettronica)
  - Random Access Memories (RAMs)
    - Consentono la memorizzazione di un pacchetto per un tempo a piacere
    - Richiedono la conversione da dominio ottico a elettronico e viceversa

---

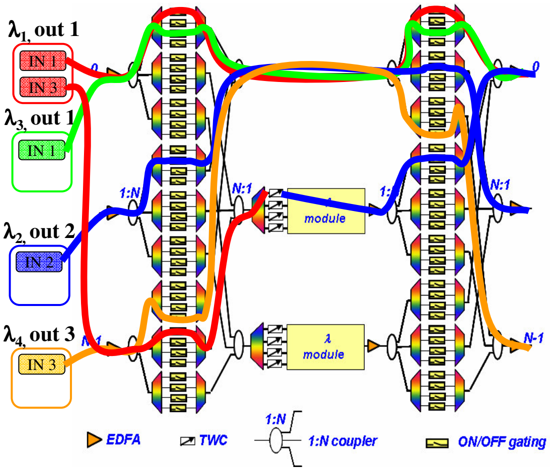
Implementazione dello schema SPW: architettura Space-Lambda-Space (S-λ-S) multi-stadio

- Architettura che implementa lo schema SPW con ridotta complessita' in termini di interruttori ottici (SOA)
- Il primo e terzo stadio sono broadcast & select
- I convertitori sono divisi in B gruppi
  - In un gruppo ci sono M TWCs, ognuno serve una differente lunghezza d'onda
  - B TWCs per lunghezza d'onda
- Complessita': $2N^2M$, non dipende dal numero di convertitori

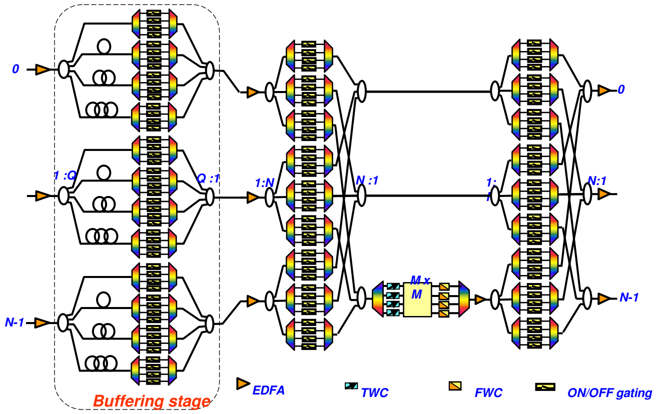


Algoritmi di controllo S-λ-S: esempio

- Algoritmo di scheduling euristico
- Fase 1: raggruppo i pacchetti
- Fase 2: spedisco i pacchetti senza conversione
- Fase 3: spedisco i pacchetti con conversione



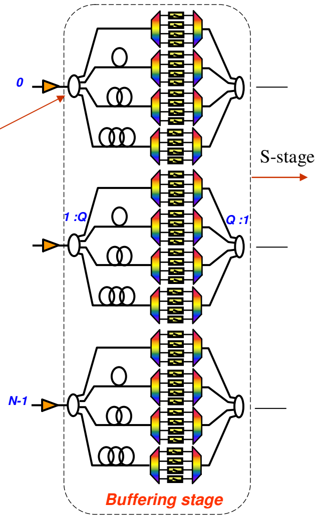
Architettura con accodamento ottico in ingresso



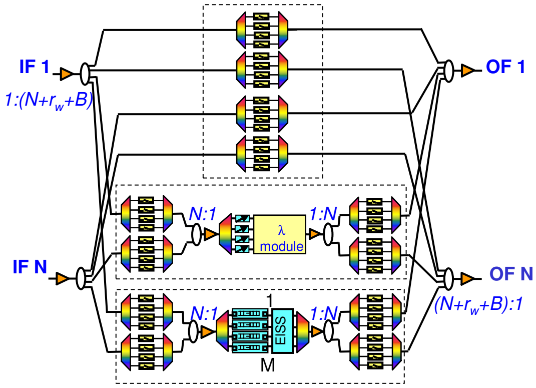
- Uno stadio di accodamento ottico basato su FDLs aggiunto in ingresso
- Rappresenta uns SPW con accodamento in ingresso

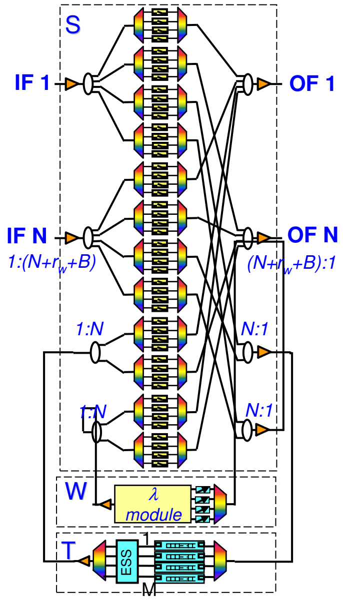
Stadio di accodamento ottico in ingresso

- Q copie del segnale WDM in ingresso
  - Possibili ritardi, D = kT, k=0..Q-1; T è la durata del pacchetto
- WSs permettono di selezionare quale pacchetto voglio spedire
  - Scheduling per input queueing possono essere applicati
- Pacchetti nello stesso canale di ingresso sono organizzati in una lista
  - FIFO
    - Liste servite in maniera FIFO
    - Servizio round-robin fra le liste
  - Window
    - Permette di superare il blocco dovuto a HOL
    - La lista viene scorsa per una finestra W e se un pacchetto è diretto a una fibra non bloccata viene spedito



Architettura ibrida opto-elettronica

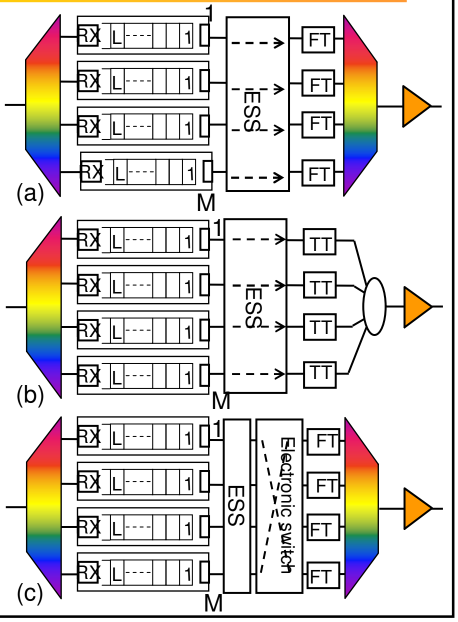




- Ho diversi blocchi di convertitori (r_w) e di buffer ***elettronici*** (B)
  - Buffer con M line card ognuno
    - Dedicate per lunghezza d'onda

Tipologie di blocchi di buffer

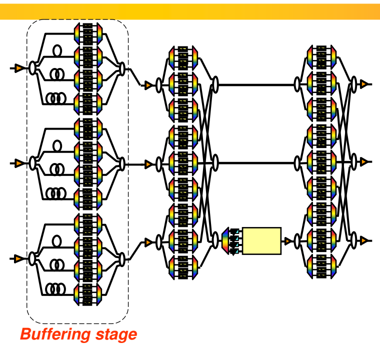
- Line cards per la memorizzazione del pacchetto elettronico con L posti in coda
- Electronic slot synchronizer per la sincronizzazione dei pacchetti da spedire nel dominio ottico
- Laser per trasmissione
  - Fixed transmitters (FT), a
  - Tunable transmitters (TT), b
- Lo schema con TT si può anche implementare con FT se si utilizza uno switch elettronico

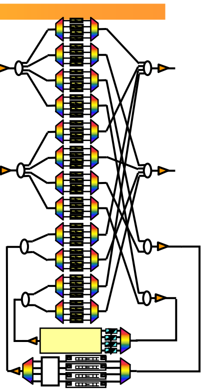


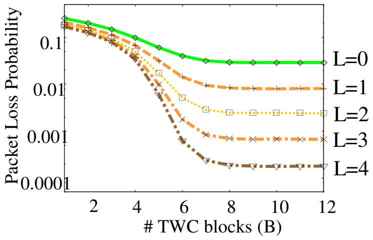
Architettura ibrida: differenti alternative

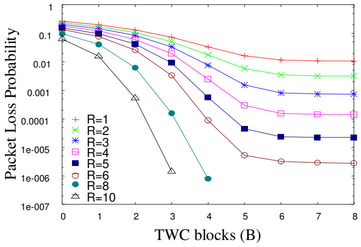
- In ogni time slot bisogna scegliere se considerare prima i pacchetti presenti nei buffer o i pacchetti provenienti dalle fibre in ingresso (diverse politiche)
  - Buffer First (BF)
  - Input Fiber First (IFF)
- A seconda delle scelte hardware e delle politiche ottengo 4 differenti possibilità
  - FT-BF
  - TT-BF
  - FT-IFF
  - FT-IFF

Risoluzione della contesa nel dominio temporale









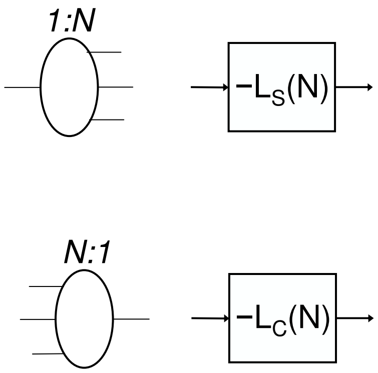
Physical evaluations

Optical devices

- Passive devices
  - Splitters/couplers, multiplexers/de-multiplexers
  - Power loss
    - Insertion loss
    - Non-uniformity
    - Polarization dependence
- Active devices
  - ON/OFF gates (SOA), wavelength converters, amplifiers (EDFA, SOA)
  - Amplify optical signals
  - Introduce noise

Splitter/couplers

- N ports Splitters/Couplers
- Three different effects are considered to contribute to loss:
  - insertion loss
    $L_{S/C,\ IL}(N) = 1 + 11\log_{10}(N)$ dB
  - polarization dependence
    $L_{S/C,\ PD}(N) = 0.3$ dB
  - non-uniformity
    $L_{S/C,\ NU}(N) = 0.55N^{0.3} - 0.05$ dB
- formulas are obtained by interpolating data got from data sheets (e. g. www.enablence.com/, www.futurexusa.com/)

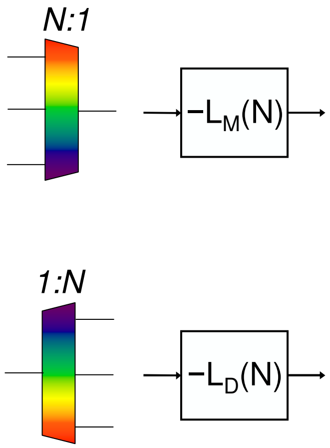


Multiplexers/de-multiplexers

- N wavelengths Multiplexers/Demultiplexers:
  - insertion loss
    $L_{M/D,\ IL}(N) = 1.2\log_2(N) - 0.4$ dB
  - polarization dependence
    $L_{M/D,\ PD}(N) = 0.05\log_2(N) + 0.05$ dB
  - non-uniformity
    $L_{M/D,\ NU}(N) = 1$ dB
  - in de-multiplexers, there is also incoherent crosstalk contribution from other wavelengths (non-ideal filters)

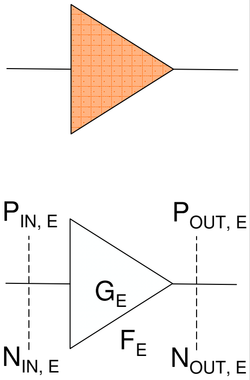
    $OX(N)|_{dB} = 10*\log10(1 + X(N))$,

    with $X(N) = 2X_A + (N-3)X_B$ (linear)
    X_A contribution from adjacent channels:
    $X_A|_{dB} = -25$ dB
    X_N contribution from non-adjacent channels:
    $X_N|_{dB} = -45$ dB
  - this contribution must be accounted at the receiver as an interference
- formulas obtained by interpolating data got from data sheets (ex http://usa.accelink.com)



EDFA

- Erbium Doped Fiber Amplifier (EDFA):
  - WDM signal with up to N wavelength channels
  - amplifies input signal (and noise)
  - characterized by gain G_E, noise figure F_E and sensitivity S_E
  - supposed to provide a constant output power equal to $P_{OUT,\ E,\ TOT}$
  - input power $P_{IN,\ E}$ must be higher than the sensitivity
    $P_{IN,\ E} \geq S_E$
    $G_E = P_{OUT,\ E,\ TOT} - P_{IN,\ E}$
  - the output power is equally divided among the wavelength channels
    $P_{OUT,\ E} = P_{OUT,\ E,\ TOT} - 10\log(N)$
  - introduces additional Amplified Spontaneous Emission (ASE) noise, output spectral noise density:
    $N_{OUT,\ E} = N_{OUT,\ E}G_E + hf(G_E - 1)F_E$
  - h Plank's constant, f frequency, $N_{IN,\ E}$, $N_{OUT,\ E}$ input and output spectral noise densities (white noise is assumed, no dependence from f)



SOA

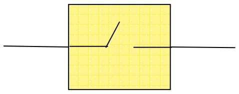
- Semiconductor Optical Amplifier (SOA):
  - often used to amplify one single wavelength channel
  - amplifies input signal (and noise)
  - characterized by gain G_S, noise figure F_S, sensitivity S_S and Extinction Ratio ER
  - supposed to provide a constant output power equal to $P_{OUT, S, TOT}$
  - the input power $P_{IN, S}$ must be higher than the sensitivity

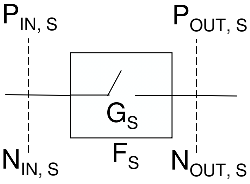
    $P_{IN, S} \geq S_S$

    $G_S = P_{OUT, S, TOT} - P_{IN, S}$
  - the output power is equally divided among the wavelength channels

    $P_{OUT, E} = P_{OUT, E, TOT} - 10 \log(N)$
  - introduces additional Amplified Spontaneous Emission (ASE) noise, output spectral noise density:

    $N_{OUT, S} = N_{OUT, S} G_S + hf(G_S - 1)F_S$
  - h Plank's constant, f frequency, $N_{IN, S}$, $N_{OUT, S}$ input and output spectral noise densities (white noise is supposed, no dependence from f)





SOA cross-talk

- Semiconductor Optical Amplifier (SOA):
  - Extinction Ratio ER: ratio between the output power in the ON and OFF state (ideally ∞)
  - due to the finite ER, a fraction of the signal traverses the SOA even in the OFF state
  - this originates coherent and incoherent crosstalk along the optical paths, which must be accounted at the receiver
  - coherent crosstalk is given by:

    $IX(N) = 10\log(1 - N/ER^*Q^2)$

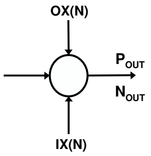
    - N is the number of interfering sources (SOAs with ER)
    - Q is a target quality factor (linear), equal to 6-7 for OOK systems with target Bit Error Rate 10^{-9} - 10^{-12}
  - incoherent crosstalk is given by:

    $OX(N) = 10\log(1 + X(N))$

    $X(N) = 2X_A + (N - 3)X_N$ (linear)

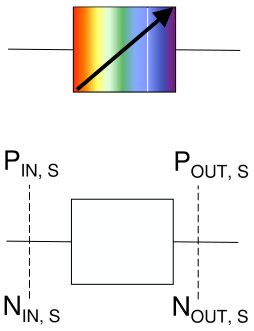
    $X_A = -25$ dB

    $X_N = -50$ dB
  - coherent and incoherent crosstalk must be accounted at the receiver as interference (additional noise sources)



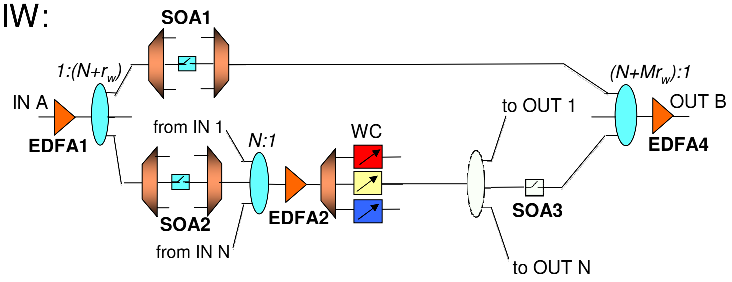
Wavelength converters

- Wavelength Converter (WC):
  - on output, a new wavelength is generated, the previous crosstalk effects are not further propagated
  - no further noise introduced
  - constant output power $P_{OUT, WC}$ = 3 dBm
  - sensitivity S_{WC} = 0 dBm (high value, an SOA must be saturated)

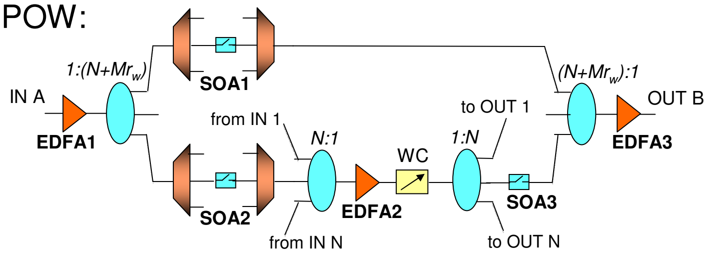


SPIW and SPOW paths

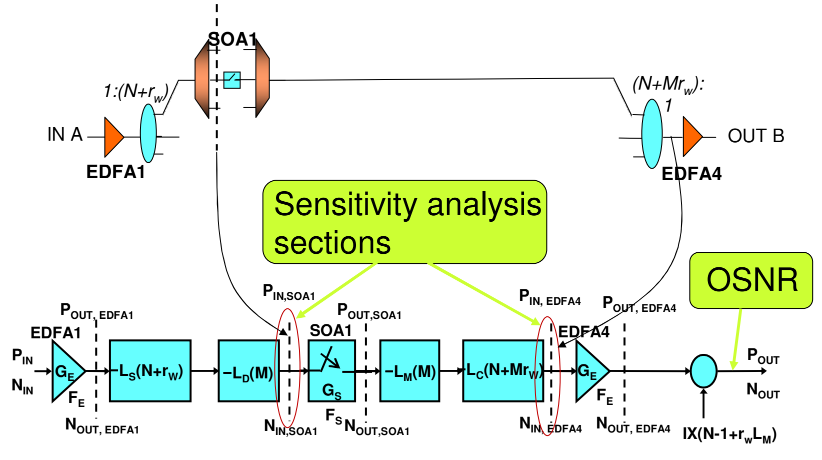
SPIW:



SPOW:



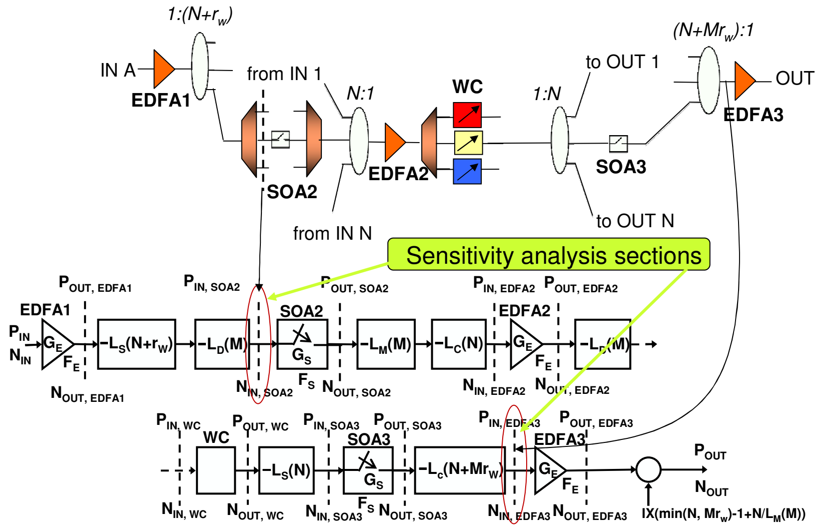
SPIW path without wavelength conversion



Sensitivity analysis sections

OSNR

SPIW path with wavelength conversion



Sensitivity analysis sections

Numerical results

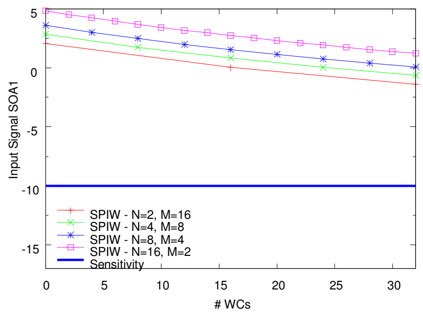
- Reference parameters:
  - EDFA: $\mathbf{P_{OUT,\,E}}$ = 25 dBm
  - SOA: $\mathbf{P_{OUT,\,S}}$ = 8 dBm, $\mathbf{S_S}$ = -10 dBm, $\mathbf{F_S}$ = 7 dB, $\mathbf{E_R}$ = 35 dB
  - WC: $\mathbf{P_{OUT,WC}}$ = 3dBm, $\mathbf{S_{WC}}$ = 0 dBm
  - $OSNR_T$ at the receiver: 20 dB
- Single channel analysis
- Amplifier requirement
  - input power > sensitivity
- Receiver requirement
  - OSNR>$OSNR_T$
- Interfering sources are accounted for as additional noise at the receiver input

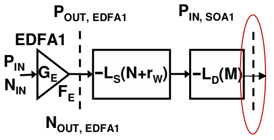
---

Sensitivity analisys: NM=32

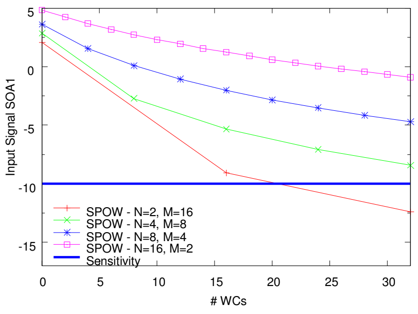
Power at SOA1 input decreases with the number of wc

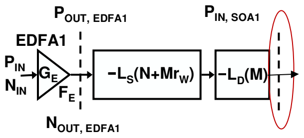
- SPIW





- SPOW

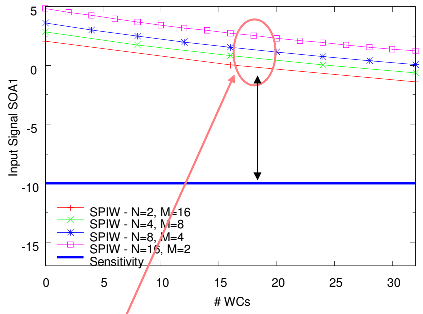


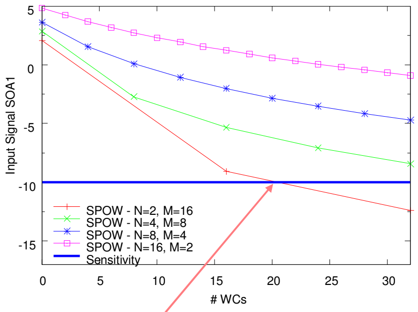


Sensitivity analisys: NM=32

Power at SOA1 input decreases with the number of wc

- SPIW
- SPOW





Input power is too low

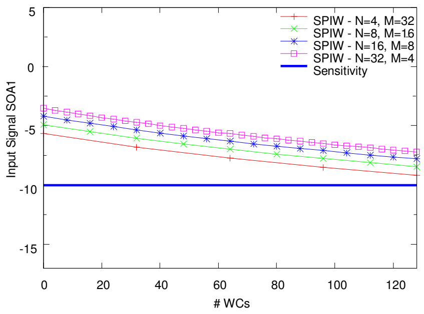
SPIW always match sensitivity constraints

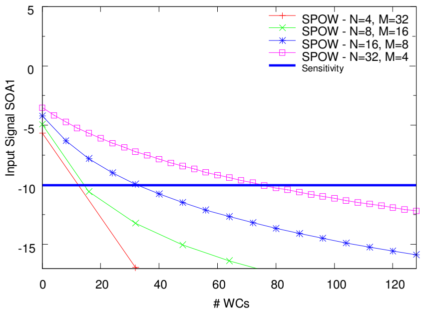
Sensitivity analysis: NM=128

-

- SPIW

SPOW

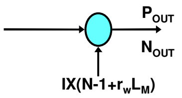




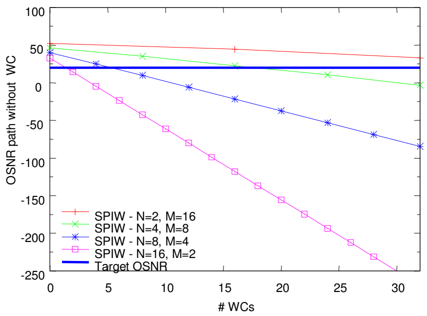
- Major limitations as the total number of channel increases
- Poor contention resolution capability in SPOW

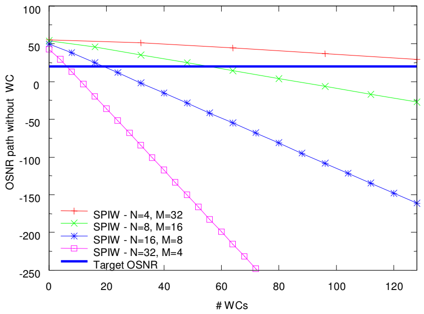
OSNR analysis: path without wc

N=32



N=128





- Strong dependence on the number of wavelength converters when N is high due to interference effects

---

Motivations for Power Consumption Evaluations

- Optical switching architectures are espected to provide high aggregate capacity with small power consumption in limited footprint
- Networking consumes more and more power as the overall capacity increases
  - power consumption must be considered as one of the main aspects in both control-plane and data-plane
- **Power consumption** of the node data-plane need to be evaluated according to the proposed switching architectures and traffic patterns
- Different switching architectures must be compared not only in terms of performance and complexity, but also in terms of feasibility/scalability and power consumption

Power Consumption in Optical Architectures

- Power consumption of the data-plane is related to active devices
  - SOAs: power consumption depends on physical parameters of the device
    - width, length, loss coefficient etc…
  - WCs: power consumption depends on the technology employed
    - cross-gain modulation (CGM), cross-phase modulation (CPM), four-wave mixing
- SOA power consumption also related to the **gain** of the amplifier
- SOA only consumes power when in ON state (allowing a packet to pass through)
  - power consumption to forward a packet depends on the number of SOAs involved
  - power consumption related to forwarding procedure and packet loss

---

Power Consumption in SOA Amplifiers and WCs

- Power consumption of SOAs and WCs has been evaluated in
  - *V. Eramo, M. Listanti, ”Power Consumtion in Bufferless Optical Packet Switches in SOA Technology”, Journal of Optical Communications and Networking (**JOCN**), Vol. 1, No. 3, August 2009, pp.15-29*

$$P \simeq V_b \left(1 + \frac{1}{\Gamma \alpha_{SOA} L} \ln(G)\right) i_t$$

$$i_t = \frac{qwdLN}{\tau} \simeq 100L$$

power consumption is related to the gain G provided by the amplifier

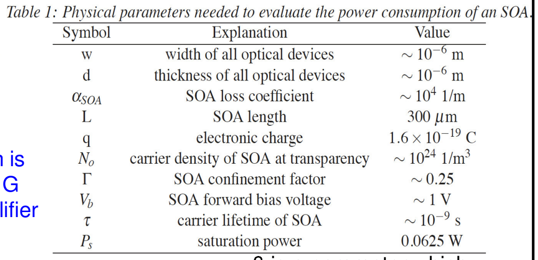


Table 1: Physical parameters needed to evaluate the power consumption of an SOA.

Symbol	Explanation	Value
w	width of all optical devices	$\sim 10^{-6}$ m
d	thickness of all optical devices	$\sim 10^{-6}$ m
α_{SOA}	SOA loss coefficient	$\sim 10^{4}$ 1/m
L	SOA length	300 μm
q	electronic charge	1.6×10^{-19} C
N_o	carrier density of SOA at transparency	$\sim 10^{24}$ 1/m^3
Γ	SOA confinement factor	~ 0.25
V_b	SOA forward bias voltage	~ 1 V
τ	carrier lifetime of SOA	$\sim 10^{-9}$ s
P_s	saturation power	0.0625 W

$$P_{WC} \simeq 100LV_b + \frac{100\beta V_b}{\Gamma \alpha_{SOA}} + 10V_b\sqrt{\frac{\beta P_s}{\Gamma \alpha_{SOA}}} + 10\sqrt{\frac{\beta P_s}{\Gamma \alpha_{SOA} V_b}}$$

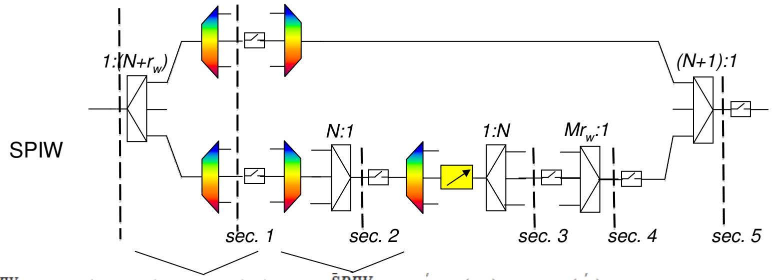
β is a parameter which depends on the technology

β = 4 CGM

β = 2 CFM

β = 0.3-0.52 FWM

SPIW Paths



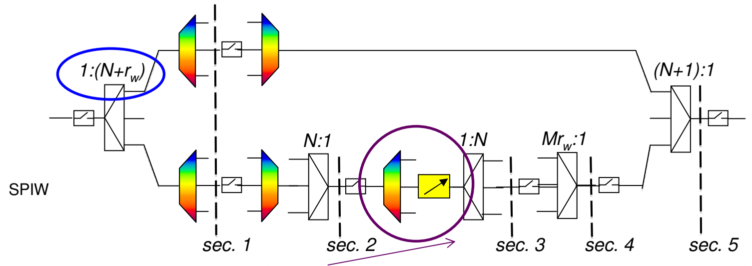
$$L_1^{SPIW} = L_{s/c}(N+r_w) + L_{D/M}(M) \qquad L_2^{SPIW} = L_{D/M}^{'}(M) + L_{s/c}(N^{'})$$

- To evaluate the gain of each SOA, the paths inside the architectures are considered and divided in different sections, according to the number of SOAs employed

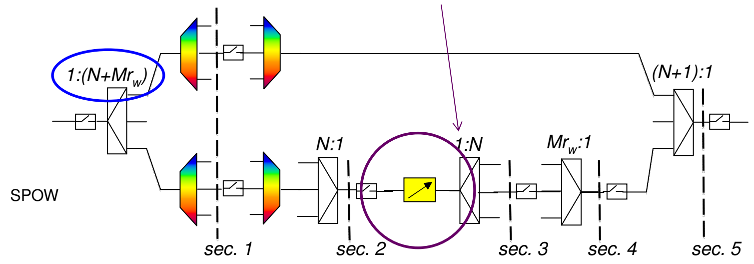
- the SOA after section **'*i*'** recovers the power losses introduced by passive optical devices in that section
  - ***$L_{s/c}$(N)*** indicates the loss introduced by a splitter/coupler with **N** outputs/inputs
  - ***$L_{D/M}$(M)*** indicates the loss introduced by a MUX/DEMUX which multipl/demultipl **M** wavelengths

---

SPOW vs. SPIW Paths



- The difference between SPIW and SPOW is the access to the WCs



Power Loss in Passive Devices

- Ideally, a splitter/coupler with N outputs/inputs introduces a loss equal to $10\log_{10}(N)$ dB, while an ideal MUX/DEMUX does not introduce any loss

- To obtain a more realistic evaluation, the actual loss of these devices has been evaluated following the methodology presented in:
  - *D. Cuda, R. Gaudino, G. A. Gavilanes, F. Neri, G. Maier, C. Raffaelli, M. Savi, "Capacity/Cost Tradeoffs in Optical Switching Fabrics for Terabit Packet Switches", ONDM 2009*

- In particular, loss expressions have been obtained by interpolating values found in data sheets of commercially available components:

- Loss of splitter/coupler:
  dB

$$L_{s/c}(N) = \underbrace{1 + 11\log_{10}(N)}_{IL} + \underbrace{0.55N^{0.3} - 0.05}_{NU} + \underbrace{0.3}_{PD}$$

INSERTION LOSS *NON UNIFORMITY* POLARIZATION DEPENDENT

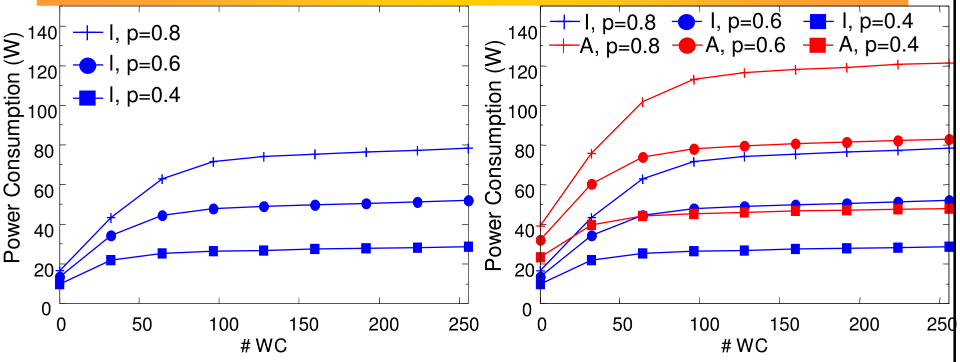
- Loss of multiplexer/demultiplexer:

$$L_{D/M}(N) = \underbrace{1.2log_2(N) - 0.4}_{IL} + \underbrace{1}_{NU} + \underbrace{0.05log_2(N) + 0.05}_{PD}$$

dB

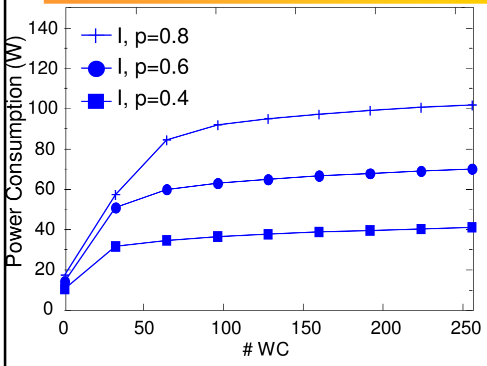
INSERTION LOSS *NON UNIFORMITY* POLARIZATION DEPENDENT

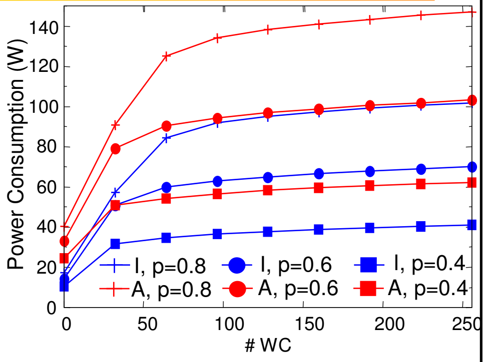
Comparison between ideal and actual power consumption: SPIW



- Power consumption of SPIW as a function of the number of WCs with **ideal** loss in passive devices
  - N=8, M=32, β=4
- When the number of WCs is low, power consumption rapidly increases with the number of WCs
- An asymptotic value is reached after a certain number of WCs (further WCs not needed)
- Power consumption significantly increases considering **actual** loss of passive devices

Comparison between ideal and actual power consumption: SPOW

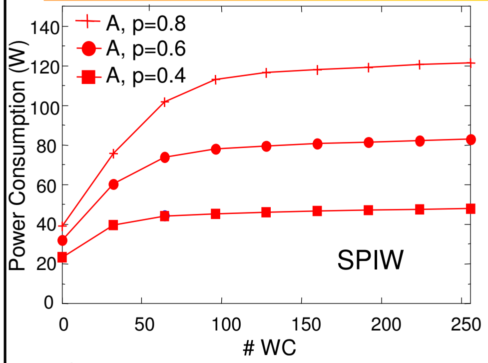


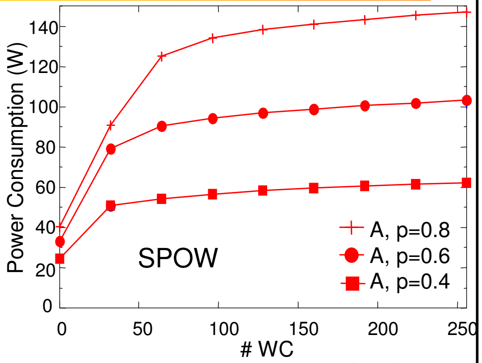


- Power consumption of SPIW as a function of the number of WCs with **ideal** loss in passive devices
  - N=8, M=32, β=4
- When the number of WCs is low, power consumption rapidly increases with the number of WCs
- An asymptotic value is reached after a certain number of WCs (further WCs not needed)
- Power consumption significantly increases considering **actual** loss of passive devices

---

Comparison between SPIW and SPOW





- Considering actual power loss in passive devices, the SPOW consumes more power than SPIW (when equipped with the same number of WCs and high load)
- The dimensioning process should take not only loss but also power consumption (and scalability analysis) into account

Considerations on Power Consumption

- Power consumption of active devices in architectures with shared wavelength converters evaluated in relation to the switching matrix organization
- To obtain realistic results, actual loss values for passive devices must be considered
- SPIW provides less power consumption than SPOW especially when the load is high
- Packet loss and, consequently, forwarding algorithm influence power consumption

---

Cost model

Optical packet switches: Requirements

- Performance
  - Minimize packet loss
- Fast scheduling algorithms
  - Reduce computational complexity
- Cost
  - Keep components as simple as possible
  - Limit the number of expensive components
- Different solutions can be compared in terms of performance, control and **cost**

Switch cost model

- Definitions:
  - N_{SOA}^{A} number of SOAs for a generic architecture A
  - N_{TWC}^{A} : number of TWCs for a generic architecture A
  - C_{SOA} : cost of an SOA
  - C_{TWC} : cost of a TWC
- Cost of architecture A:
  - $C_A = N_{SOA}^{A} C_{SOA} + N_{TWC}^{A} C_{TWC}$
  - Parametric evaluation of cost:
  - $C_{TWC} = \alpha C_{SOA}$ $C_A = (N_{SOA}^{A} + \alpha N_{TWC}^{A}) C_{SOA}$

Cost comparison

- Two architectures (A and B) relying on the same kind of TWCs are considered:
  - $C_A = (N_{SOA}^{A} + \alpha N_{TWC}^{A})\, C_{SOA}$
  - $C_B = (N_{SOA}^{B} + \alpha N_{TWC}^{B})\, C_{SOA}$
- Architecture ***A is less expensive than B*** ($C_A < C_B$) when:
  - $N_{SOA}^{A} + \alpha N_{TWC}^{A} < N_{SOA}^{B} + \alpha N_{TWC}^{B}$
- Find the limiting value of α_{th} satisfying the previous condition

$$\alpha_{th} = \frac{N_{SOA}^{A} - N_{SOA}^{B}}{N_{TWC}^{B} - N_{TWC}^{A}}$$

- This condition results in
  - $\alpha > \alpha_{th}$ when $N_{TWC}^{A} < N_{TWC}^{B}$
  - $\alpha < \alpha_{th}$ when $N_{TWC}^{A} > N_{TWC}^{B}$

SPN and MS-SPW comparison

- SPN is less costly than MS-SPW when:
  - $(N_{SOA}^{SPN} + \alpha N_{TWC}^{SPN}) < (N_{SOA}^{MS\text{-}SPW} + \alpha N_{TWC}^{MS\text{-}SPW})$
- *Here* $N_{TWC}^{MS\text{-}SPW} - N_{TWC}^{SPN} > 0$ and $N_{SOA}^{SPN} - N_{SOA}^{MS\text{-}SPW} > 0$

$$\Rightarrow \quad \alpha > \alpha_{th} = \frac{N_{SOA}^{SPN} - N_{SOA}^{MS\text{-}SPW}}{N_{TWC}^{MS\text{-}SPW} - N_{TWC}^{SPN}}$$

- $N_{TWC}^{MS\text{-}SPW}$ and N_{TWC}^{SPN} are the minimum number of TWCs to obtain the asympotic value of PLP
- N_{SOA}^{SPN} and $N_{SOA}^{MS\text{-}SPW}$ are given by:
  - $N_{SOA}^{SPN} = (NM)^2 + NR(2M-1)$
  - $N_{SOA}^{MS\text{-}SPW} = 2N^2M$

Value of α_{th} as a function of N and M

- Value of α_{th} varying:
  - the number of fibers in case M=16, p=0.3, 0.5, 0.7
  - The number of wavelengths per fiber in case N=16, p=0.3, 0.5, 0.7
- The number of wavelength converters is chosen as the minimum to obtain the asymptotic loss probability
- SPN less costly than MS-SPW when $\alpha > \alpha_{th}$
- The value of α_{th} are very high because SPN uses strictly non-blocking space matrixes
- α_{th} increases when N or/and M increases