



## Esercitazioni di Sistemi di commutazione LS

---

*Ing. Michele Savi*  
*DEIS - Università di Bologna*

*michele.savi@unibo.it*

## Analisi di architetture di commutazione

---

- Valutazione delle prestazioni di architetture di commutazione:
  - metodo analitico
  - metodo simulativo
- In questo contesto
  - metodo simulativo
    - simulatori ad hoc realizzati in linguaggio C

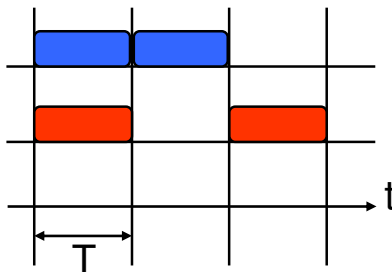
## Scenario

- Si considerano architetture di commutazione:
  - completamente ottiche (no conversione O/E/O)
  - a pacchetto:
    - in ambiente sincrono
      - Arrivi dei pacchetti a istanti discreti (slot)
      - Durata dei pacchetti pari alla durata di uno slot
    - in ambiente asincrono
      - Traffico di Poisson
      - Arrivi dei pacchetti distribuiti esponenzialmente
      - Durata dei pacchetti con durata distribuzione esponenziale
  - bufferless (no memorizzazione dei pacchetti)

## Processo degli arrivi

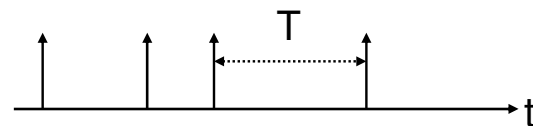
- Caso sincrono

- carico per canale  $p$  rappresenta la probabilità di presenza/assenza di un pacchetto



- Caso asincrono

- arrivi generati su un singolo canale secondo un processo di Poisson



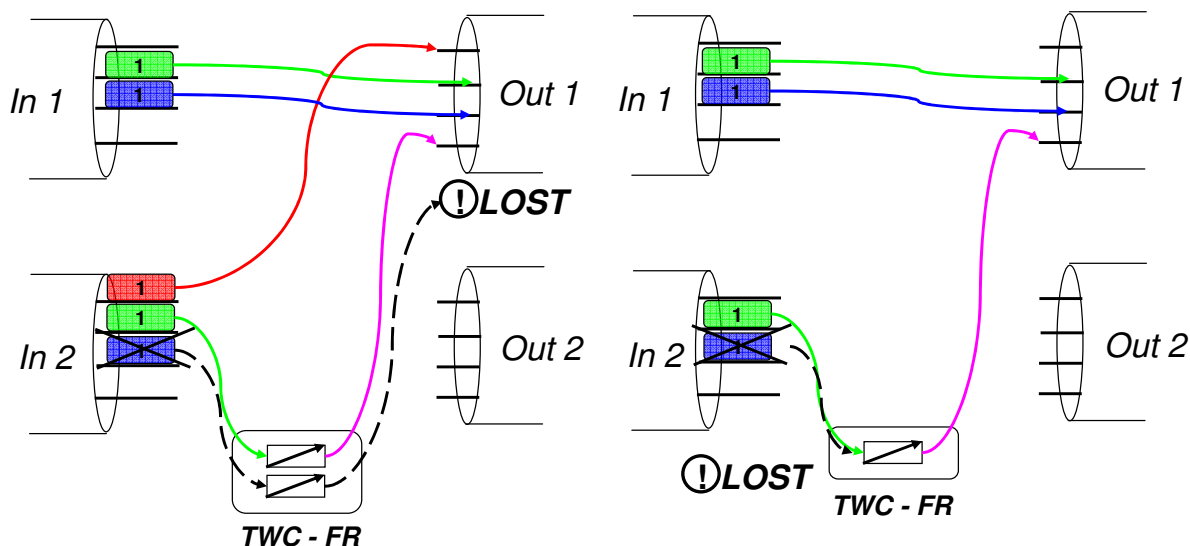
- tempo di interarrivo generato casualmente con distribuzione esponenziale e valor medio  $\mu=1$
- $p = \lambda \mu \quad \rightarrow \quad \lambda = p/\mu$

## Risoluzione della contesa

- Risoluzione della contesa non è possibile nel dominio temporale (assenza di buffer)
- Risoluzione della contesa nel dominio ottico utilizzando convertitori di lunghezza d'onda
  - FWC (fixed wavelength converter): convertono una lunghezza d'onda in un'altra con assegnamento statico
  - TWC (tunable wavelength converter) : convertono un set di lunghezze d'onda in un altro
    - TWC – FR (FULL RANGE), possono convertire una qualsiasi lunghezza d'onda verso ogni altra
    - TWC – LR (LIMITED RANGE), possono convertire una lunghezza d'onda del sub-set di ingresso verso ogni lunghezza d'onda del sub-set di uscita

## Conversione di lunghezza d'onda

- Risoluzione della contesa nel dominio ottico utilizzando convertitori di lunghezza d'onda

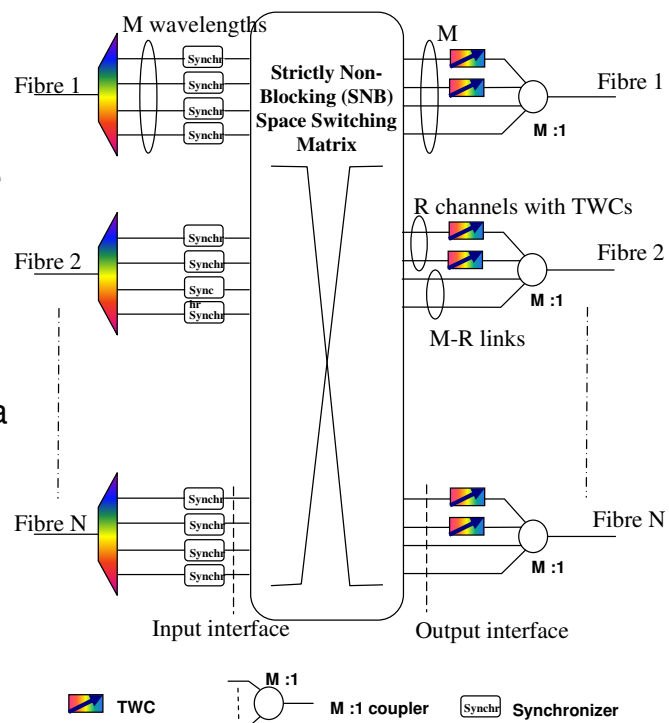


## Architetture considerate

- 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 SPW (shared per wavelength):
  - convertitori di lunghezza d'onda condivisi fra i pacchetti in ingresso sulla stessa lunghezza d'onda
  - simulatore: **SPW.c**

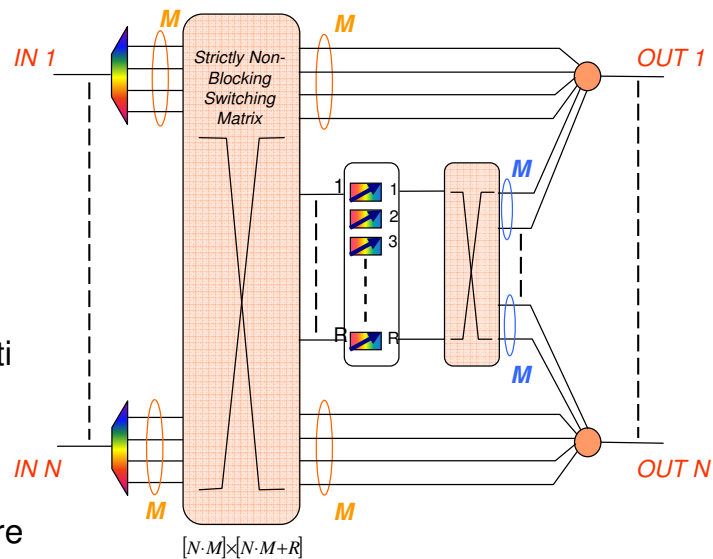
## SPL

- Pacchetti in ingresso vengono sincronizzati
- Matrice spaziale non bloccante permette l'inoltro dei pacchetti verso convertitori o semplici link ottici senza blocco interno
- Convertitori di lunghezza d'onda sulle interfacce di uscita permettono di risolvere situazioni di contesa
- Ogni fibra di uscita ha i propri convertitori dedicati



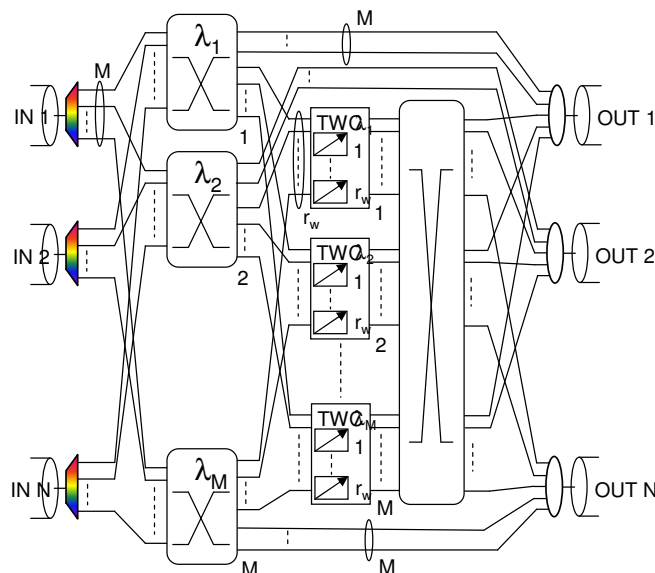
## SPN

- Pacchetti in ingresso vengono sincronizzati
- Matrice spaziale non bloccante permette di accedere direttamente alle fibre di uscita (no conversione) o di sfruttare i convertitori di lunghezza d'onda condivisi
- Altra matrice non bloccante permette di inviare i pacchetti convertiti alle rispettive fibre di destinazione
- Tutti i pacchetti possono sfruttare qualsiasi convertitore



## SPW

- Pacchetti in ingresso vengono sincronizzati
- M matrice spaziali non bloccanti permettono di accedere direttamente alle fibre di uscita (no conversione) o di sfruttare i convertitori di lunghezza d'onda condivisi per lunghezza d'onda
- Altra matrice non bloccante permette di inviare i pacchetti convertiti alle rispettive fibre di destinazione
- Tutti i pacchetti sulla stessa lunghezza d'onda sfruttano i convertitori nello stesso blocco



## Simulatori: considerazioni generali

---

- Simulatori permettono di calcolare il valor medio della probabilità 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

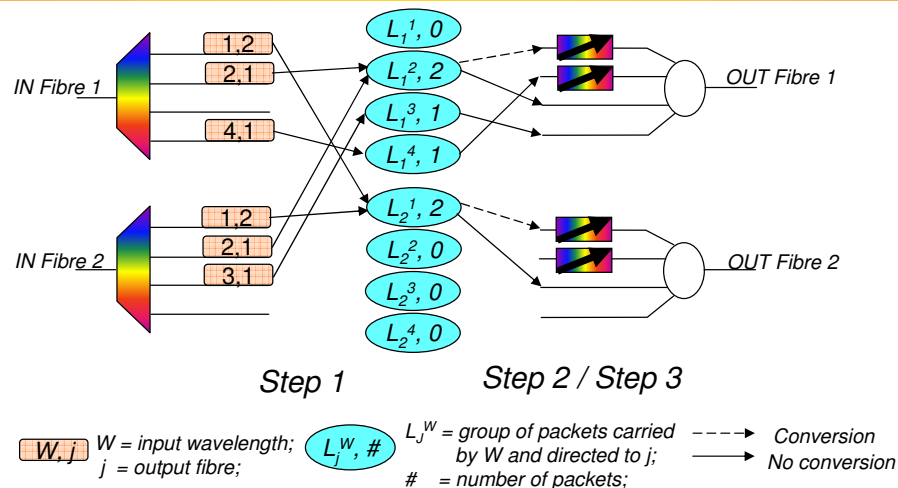
---

- 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'inizio 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( $\lambda_{in}$ , out)* (si differenzia da architettura ad architettura)
- Procedura *Sel\_lambda( $\lambda_{in}$ , 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( $\lambda_{in}$ , 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'è 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*( $\lambda_{in}$ , *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'è 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*( $\lambda_{in}$ , *out*) per MS-B&S.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)

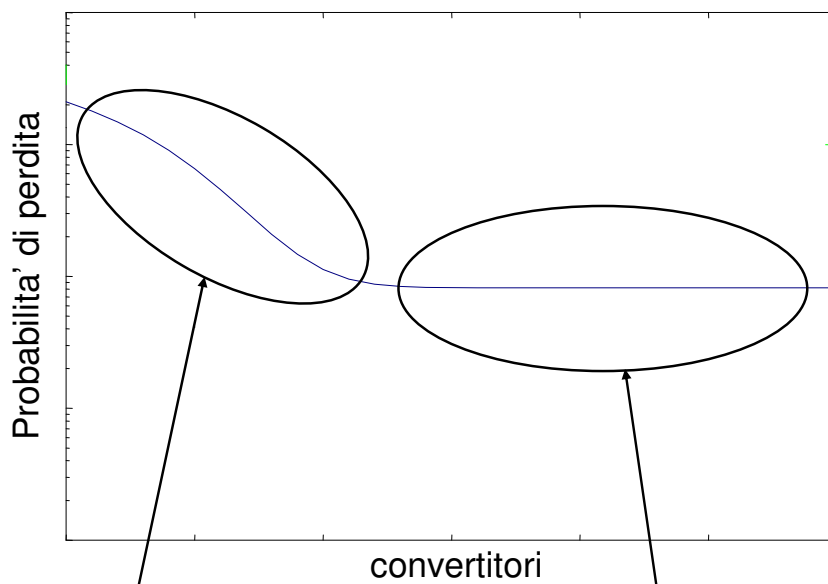
---

- Confronto fra i tempi di simulazione
  - In funzione del carico
  - In funzione delle dimensioni del commutatore
  - A parità di precisione.
- Confronto tra tempo di simulazione e tempo di esecuzione del modello
- Confronto analisi/simulazione

---

## Probabilità' di perdita in funzione dei convertitori

---



perdita aggiuntiva dovuta a  
mancanza di convertitori

Valore asintotico: dovuto a  
perdita per blocco in uscita