Using shared wavelength converters effectively in **OPTICAL SWITCHING**

Kurt Van Hautegem  
joint work with Wouter Rogiest and Herwig Bruneel

Faculty of Engineering and Architecture ♦ Department of Telecommunication and Information Processing ♦ SMACS research group ♦ Kurt.VanHautegem@Ugent.be

### OPTICAL BACKBONE

- technological developments & internet-based business models
- demand for bandwidth

nowadays:
- unlimited fiber capacity
- switching bottleneck

optical burst/packet switching:
- no dedicated communication channel
- use of available capacity
- contention

### CONTENTION RESOLUTION

electronic buffering(RAM): too slow for optical speeds

- optical contention resolution

wavelength converters (WCs)

- packets arrive on c ≠ wavelengths
- rWCs to schedule packets on same set of c wavelengths

Fiber Delay Lines (FDLs)

- set of fibers, # = N+1
- lengths j-D, j=0...N
- D = granularity

### SCHEDULING BASICS

- choose:
  - outgoing wavelength i (i=1...c)
  - delay line j (j=0...N)

constraints:
- availability of wavelength converter
- no overlap

- type of algorithm: NVF & VF

(N)VF = (non-)void-filling

satisfied: Scheduling Points (SPs)

- goal: minimize loss probability (LP)

choose 1 SP

- NVF: minimum gap
- VF: minimum delay

delay = FDL of SP

### 4 NEW ALGORITHMS

- assign cost to each SP
- choose SP with lowest cost
- v: # available WC (v=1...r)
- w: arriving wavelength (w=1...c)

- α, β & ε: algorithm parameters

optimised for minimal LP

2 cost functions:

#### C: cost of SP:

\[ C = α \cdot gap + (1 - α) \cdot delay \]

- algorithms C-NVF and C-VF

#### CW: cost of SP:

\[ CW = \left( \frac{1}{1 + \beta} \right)^{\epsilon} \cdot [α \cdot gap + (1 - α) \cdot delay]  
+ \left[ \frac{\beta}{1 + \beta} \cdot D \cdot [1 - \delta_{wl}] \right] \cdot e^{\epsilon-1} \]

extra summound to penalise use of WC

- algorithms CW-NVF and CW-VF

### RESULTS

- N=9, D=100, c=4, r=1...4
- inter-arrival time: Poisson (average E[T])
- packet size: exponential (E[B]=100)
- load= p = E[B] / (c\cdot E[T]) = 80 %
- Monte Carlo simulation

LP reduction with respect to currently best NVF and VF algorithms:

<table>
<thead>
<tr>
<th>r</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-NVF</td>
<td>2,5%</td>
<td>3,5%</td>
<td>4,1%</td>
<td>5,2%</td>
</tr>
<tr>
<td>C-VF</td>
<td>0,8%</td>
<td>3,4%</td>
<td>5,2%</td>
<td>8,9%</td>
</tr>
<tr>
<td>CW-NVF</td>
<td>8,8%</td>
<td>12,4%</td>
<td>12,7%</td>
<td>10,4%</td>
</tr>
<tr>
<td>CW-VF</td>
<td>15,0%</td>
<td>25,3%</td>
<td>29,3%</td>
<td>28,9%</td>
</tr>
</tbody>
</table>

### CONCLUSIONS

- C-NVF & C-VF: weighted average delay & gap
- CW-NVF & CW-VF: weighted average delay & gap + penalised use of WC (energy consumer)

- performance / + energy consumption \