# TCP Window Estimation for Burst Assembly in OBS Networks

**Maurizio Casoni**
(maurizio.casoni@unimore.it)

Department of Information Engineering
University of Modena and Reggio Emilia
Italy

# Outline

- Introduction: Optical Burst Switching scenario
- Assembly algorithms
  - Time and volume based

  → TCP Window based
- How evaluate the current window ?
- Proposed multi-queue window-based assembly scheme
- Investigated scenario and numerical results
  - Ns2 simulation tool
  - TCP throughput and CWND behaviour
- Conclusions



# Optical Networks: Evolution

- DWDM technique
  - **Transmission rate in the range of Tbit/s**
- Architectural semplification
  - **From IP over ATM over SONET over WDM to IP over WDM**
- Need to exploit in an effective way the huge transmission bandwidth with IP traffic
  - **Wavelength Routing**
    - ✓ all-optical data network
    - ✓ Low flexibility for IP traffic
  - **Optical Packet Switching**
    - ✓ Ideal transfer mode for IP traffic
    - ✓ Severe technological constraints → not feasible in the short/middle term
      - Optical components immature
      - Optical buffers

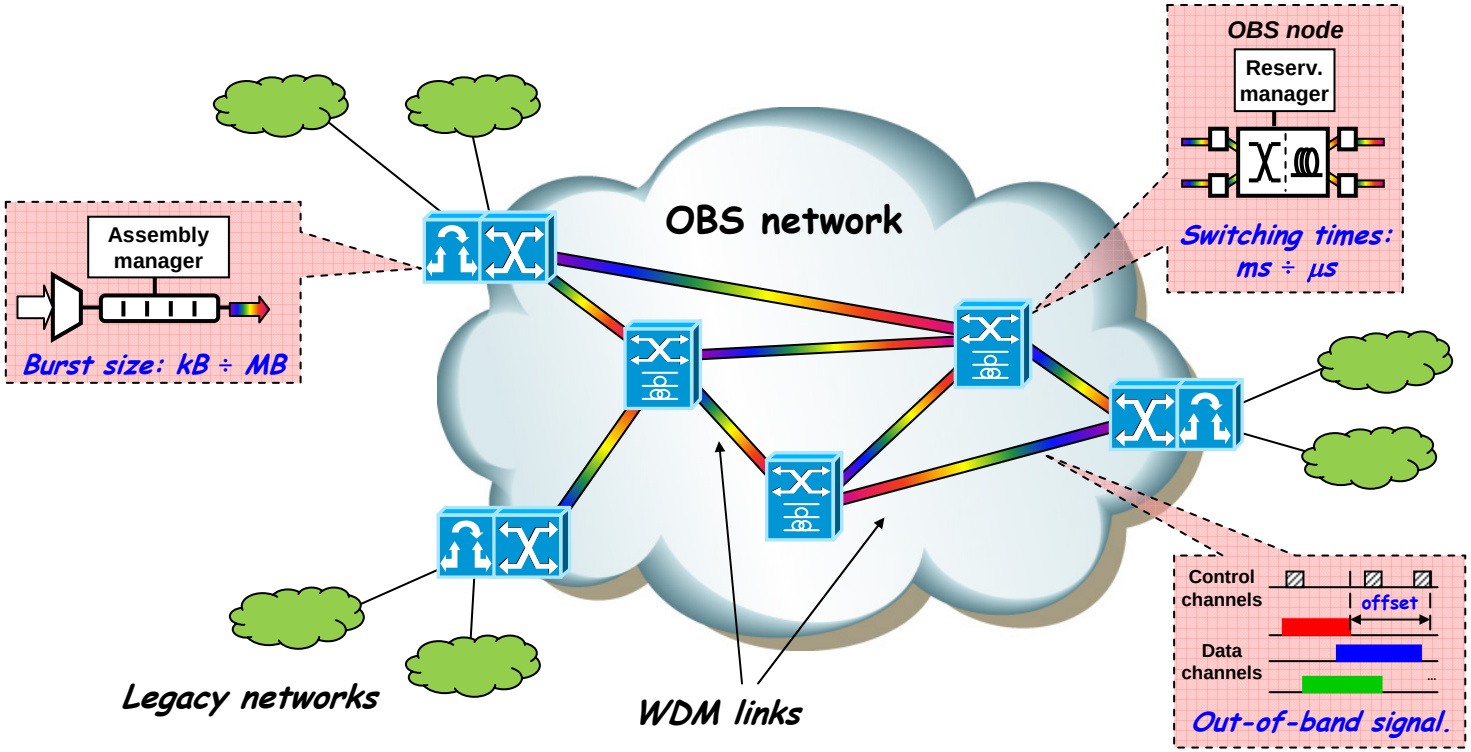
# Optical Burst Switching

**Goal**: better sinergy between the mature electronic technologies and the new optical tecnologies (mid-term solutions)

- **Switching granularity between WR and OPS**
  - *Burst concept:* aggregation of IP packets with common features (e.g. destination and QoS), considered as the basic optical unit
- **Time and space separation of data and control (header) fields**
  - Control packet employs dedicated channel and precedes the relative data burst
    - All-optical network, buffer-less and data trasparent
    - Hybrid opto-electronic network for control signals (*out-of-band signaling*)
  - Simplification of the electronic processing of the control packets at intermediate nodes
  - Reduction of the opto-electronic functionalities required to router

# OBS network

- Data coming from legacy networks are aggregated into a burst in edge nodes
- The control packet is sent first in order to reserve the resources in intermediate nodes
- The burst follows the control packet with some offset time, and it crosses the nodes remaining in the optical domain



# Window based burst assembly

- Past studies have shown that TCP performance are influenced by both assembly time and current TCP congestion window size
- **Basic idea**: Multi-queue assembly scheme, where TCP flows are assigned to queues according to current TCP window size
- Different assembly timers are assigned to each queue

- **Performance problem (and goal): how to properly evaluate the right TCP window size value for the current flow ?**

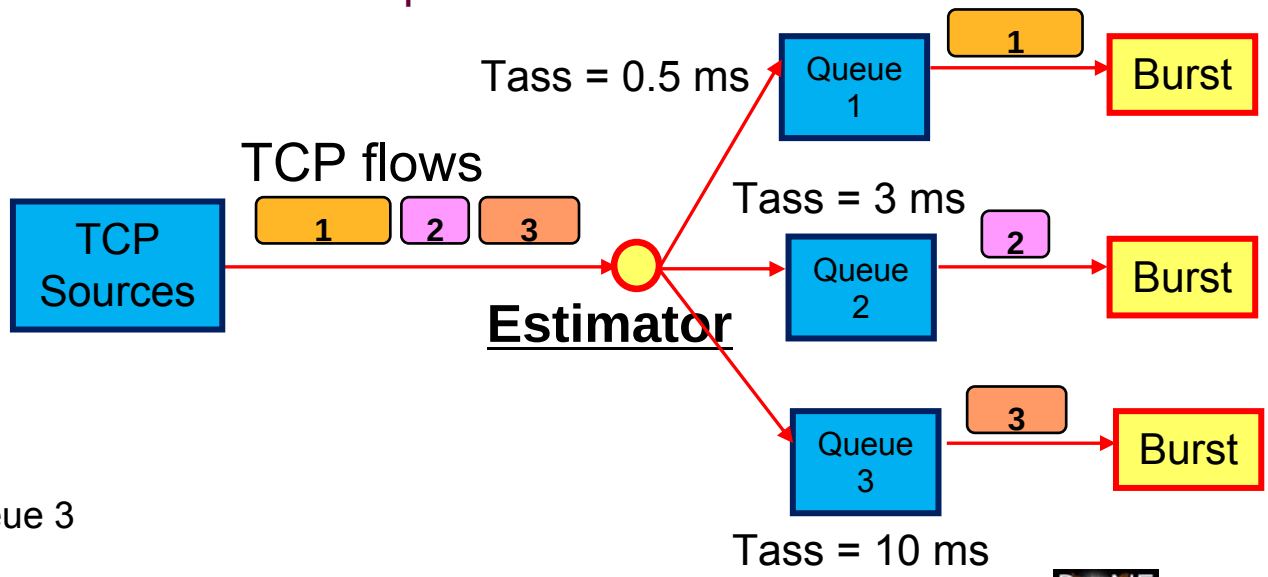
# Proposed Window based assembly

- Mixed flow multi-queue hybrid assembly algorithm working on the current estimated value of the congestion window W[t];
- Use of (simple) estimator of the congestion window:
  - ✓ Storing of the number of consecutive segments for each flow
  - ✓ Assumption of this value (+1) as current TCP window
  - ✓ Assumption of congestion avoidance operation

If W[t] < 64 then Queue 1;

Else if W[t] < 128 then Queue 2;

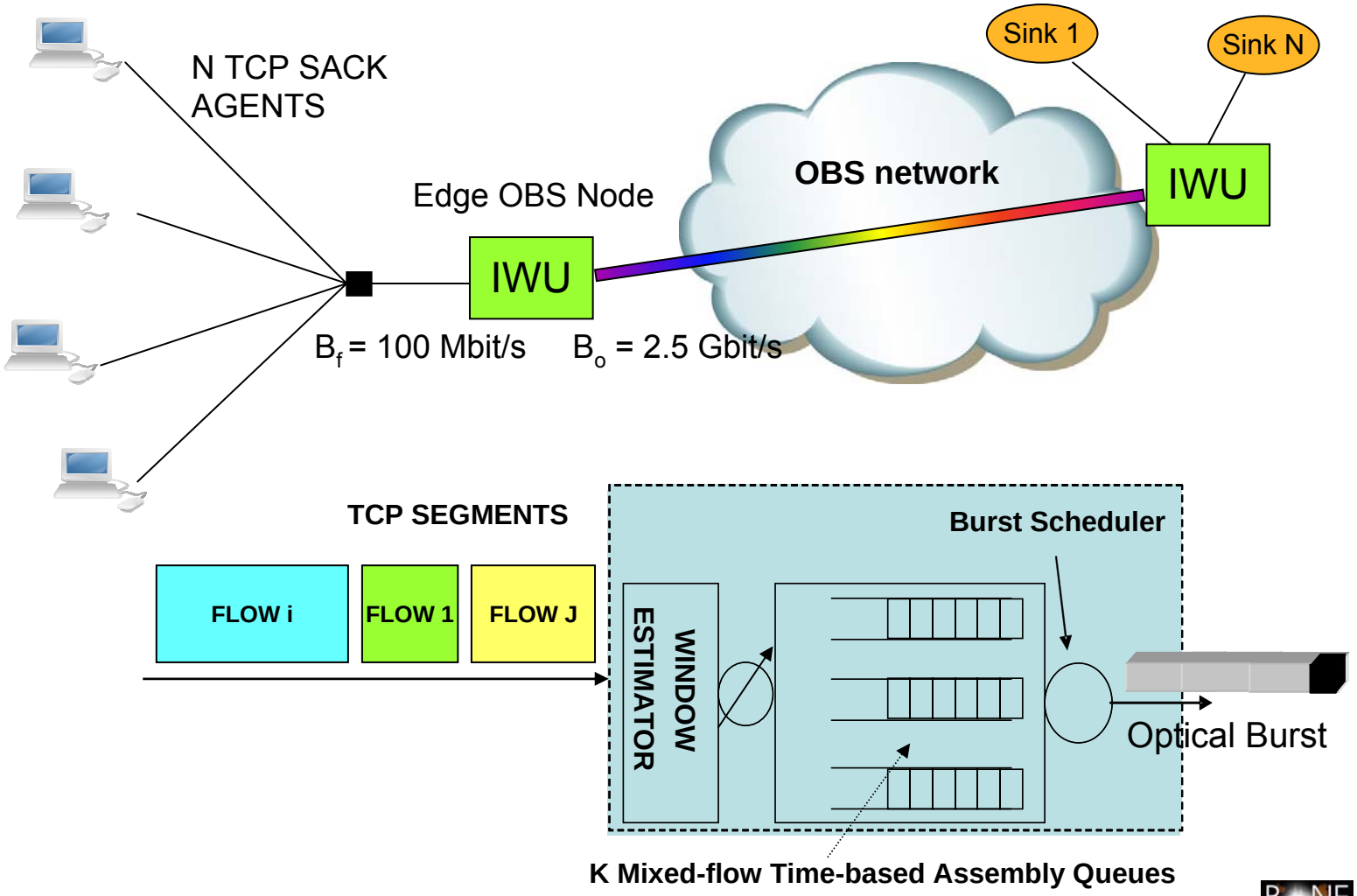
Else if 128< W[t] < Wmax then Queue 3



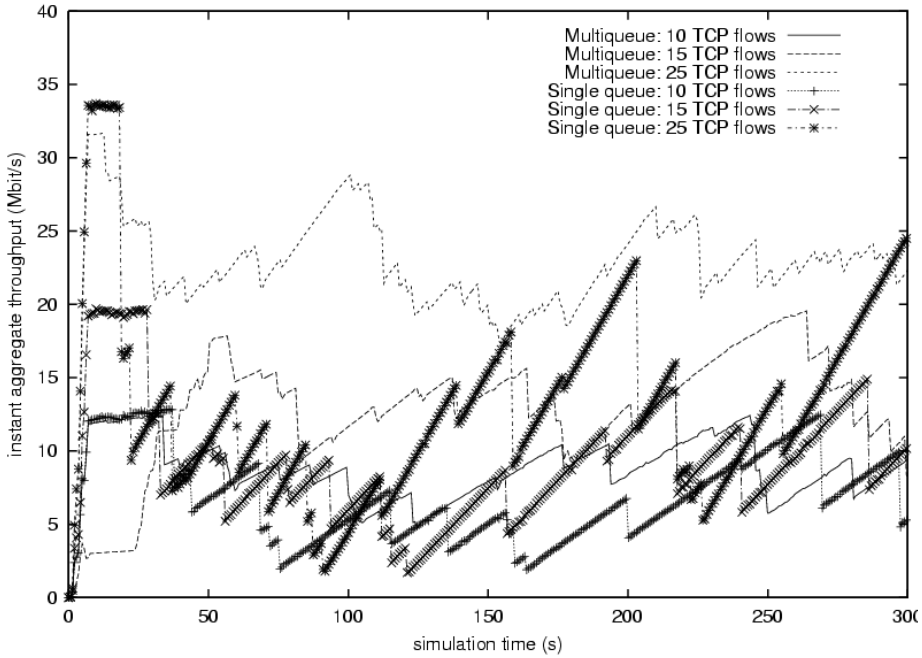
# PERFORMANCE EVALUATION

- **TCP SACK**
  - SACK option: the receiver informs the sender about the successfully r eceived segments and sender retransmits lost segments only
- **Throughput**
  - Measure of the variability of the bandwidth usage over a given time-scale
- **Instant throughput**
  - Measure of the variability of the bandwidth usage in a very short time interval and it is strictly related to the congestion window behaviour
- **Average throughput**
  - Amount of successfully transmitted bytes over a given time interval
- **Aggregated average throughput**
  - Average throughput over all active TCP flows

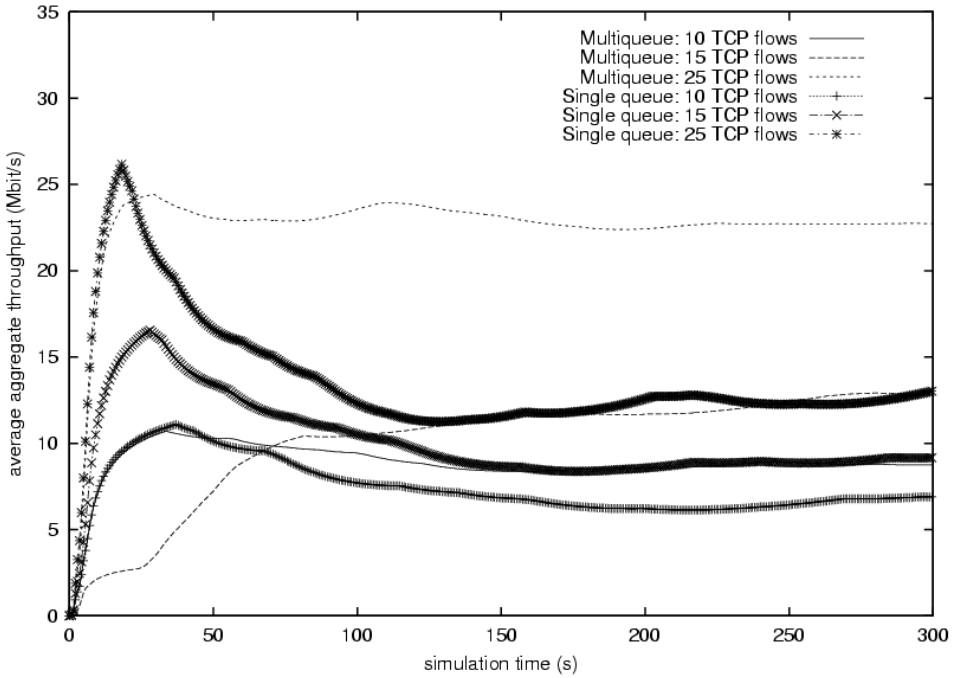
# Investigated Network Scenario



# TCP Throughput




Instant

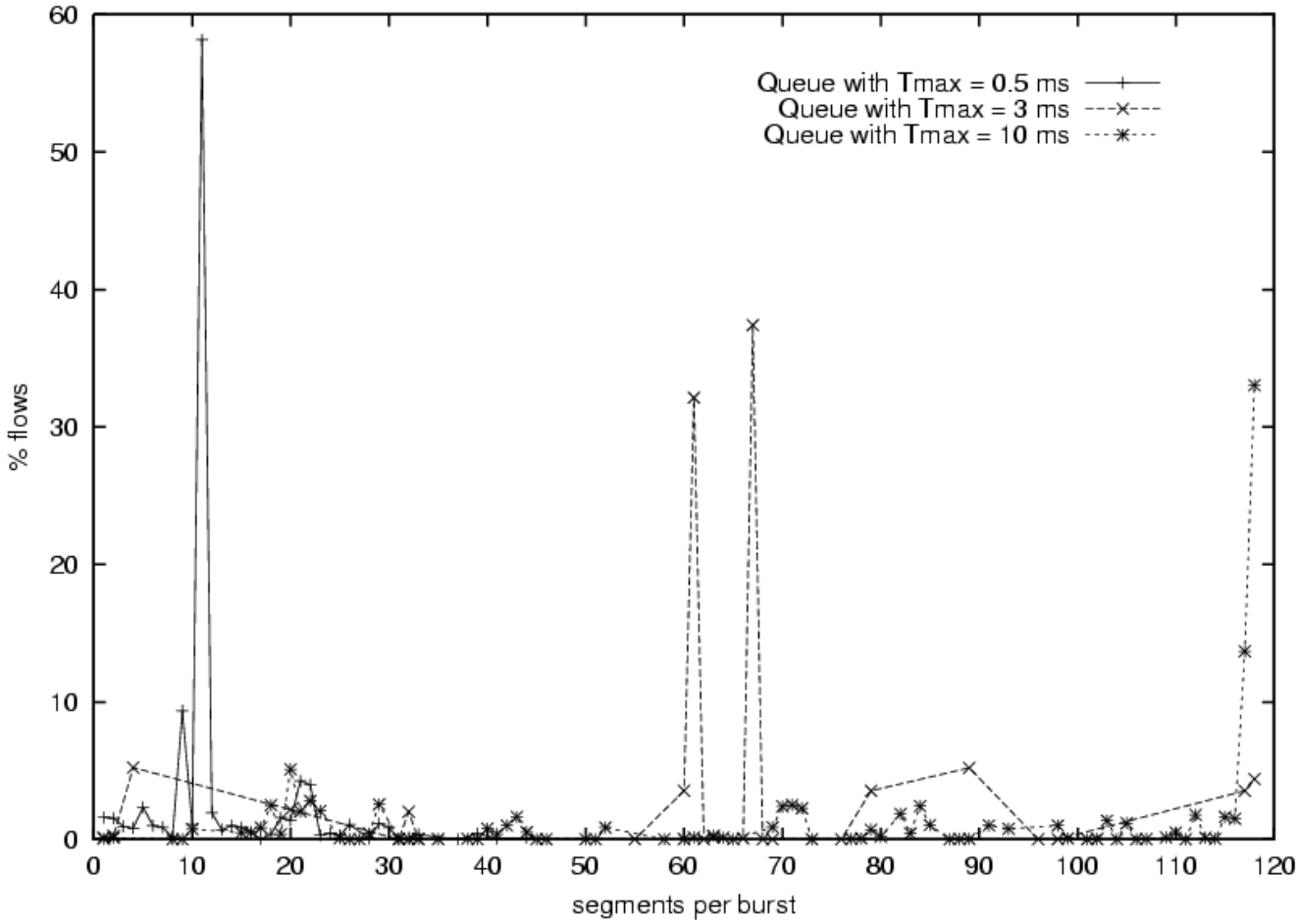



Average

- Multiqueue: three queues with Tmax = 0.5, 3 and 10 ms
- Single queue: Tmax = 3 ms
- Wmax = 256 segments
- Remarkable improvements, in particular for large numbers of TCP flows (almost 100% for N = 25)

# TCP flow probability distribution

- **N = 10**
- **Wmax = 256**



# TCP Intra Fairness

| $W_{max}$ / $T_{kmax}$ | 64 | 128 | 256 |
|---|---|---|---|
| $T_{1\_max}$= 0.5 ms | 0.9821 | 0.9627 | 0.9365 |
| $T_{2\_max}$= 3 ms | 0.9996 | 0.9691 | 0.9419 |
| $T_{3\_max}$=10 ms | 0.9999 | 0.9970 | 0.9721 |

**N = 10 flows**

# Conclusions

- TCP performance over Optical Burst Switching networks has been investigated by considering the burst assembly function
- A TCP window-based assembly algorithm has been proposed
  - works on multiple queues, where each queue is characterized by a given $T_{max}$
  - Incoming TCP segments are assigned to a given queue depending on the estimated current value of the congestion window of the flow they belong to
- A simple but effective estimator has then been added to make the assembly algorithm more realistic and feasible
- Results, obtained through simulations by means of the ns2 simulation tool, show that this algorithm can lead to TCP performance improvements of 100%, compared to a timer based assembly scheme operating on a single queue
- Next steps: improve the window estimation for, hopefully, even better results

*THANK YOU FOR YOUR ATTENTION*

**maurizio.casoni@unimore.it**
**casoni@ieee.org**

**http://www.dii.unimore.it/~mcasoni**

**… suggestions are very very welcome**