

### 5.3 Towards Scalable Switches

- Buffer throughput limitation  $\Rightarrow$  use input queueing or CIOQ
- Input queued crossbar scalability limited primarily by:
  - quadratic cost growth rate,  $O(N^2)$ , of crossbar
  - scheduler complexity & efficiency, i.e. solving the output contention (congestion management) problem
- To solve the crossbar cost  $\Rightarrow$  use switching fabrics
- To solve the scheduler / contention / congestion problem:
  - (sorting / self-routing networks – bad solution)
  - Switching Fabrics with Small Internal Buffers, large input VOQ's, and Internal Backpressure (Flow Control)

Central Scheduler is Impractical for large N

Solution 1: **Sorting Networks w. Distributed Control** (see ch.5 for details)

- all incoming packets allowed in – no central scheduling
- conflicting packets appropriately steered – distributed control

$(N+k) \times (N+k)$   
Sorting Network

$N$  inputs

$N$  outputs

$K$  conflicting packets recirculated

$(N) \times (sN)$   
Sorting Network

$N$  inputs

$N$  outputs

- uses communication paths as buffer memory ... too expensive
- Knock-Out Style but different Sw. Fabric
- Sorting Networks are quite large... not too practical

CS-534 - Copyright University of Crete 3

Central Scheduler is Impractical for large N

Solution 2: **Switching Fabrics with Internal Buffering & Backpressure**

Switching Fabric  
Distributed Scheduling

req-grant backpressure

small internal buffers

the traffic here may have packets that are short-term-conflicting in the switching fabric, but are long-term-non-conflicting in the fabric

handled by queue

owing to backpressure and distributed scheduling

CS-534 - Copyright University of Crete 4