Last Lecture: Network Layer

1. *Design goals and issues*
2. *Basic Routing Algorithms & Protocols*
3. *Addressing, Fragmentation and reassembly*
4. *Internet Routing Protocols and Inter-networking*
5. *Router design*
   1. *Short History + Router architectures* ✔
   2. *Switching fabrics*
   3. *Address lookup problem*
6. *Congestion Control, Quality of Service*
7. *More on the Internet's Network Layer*

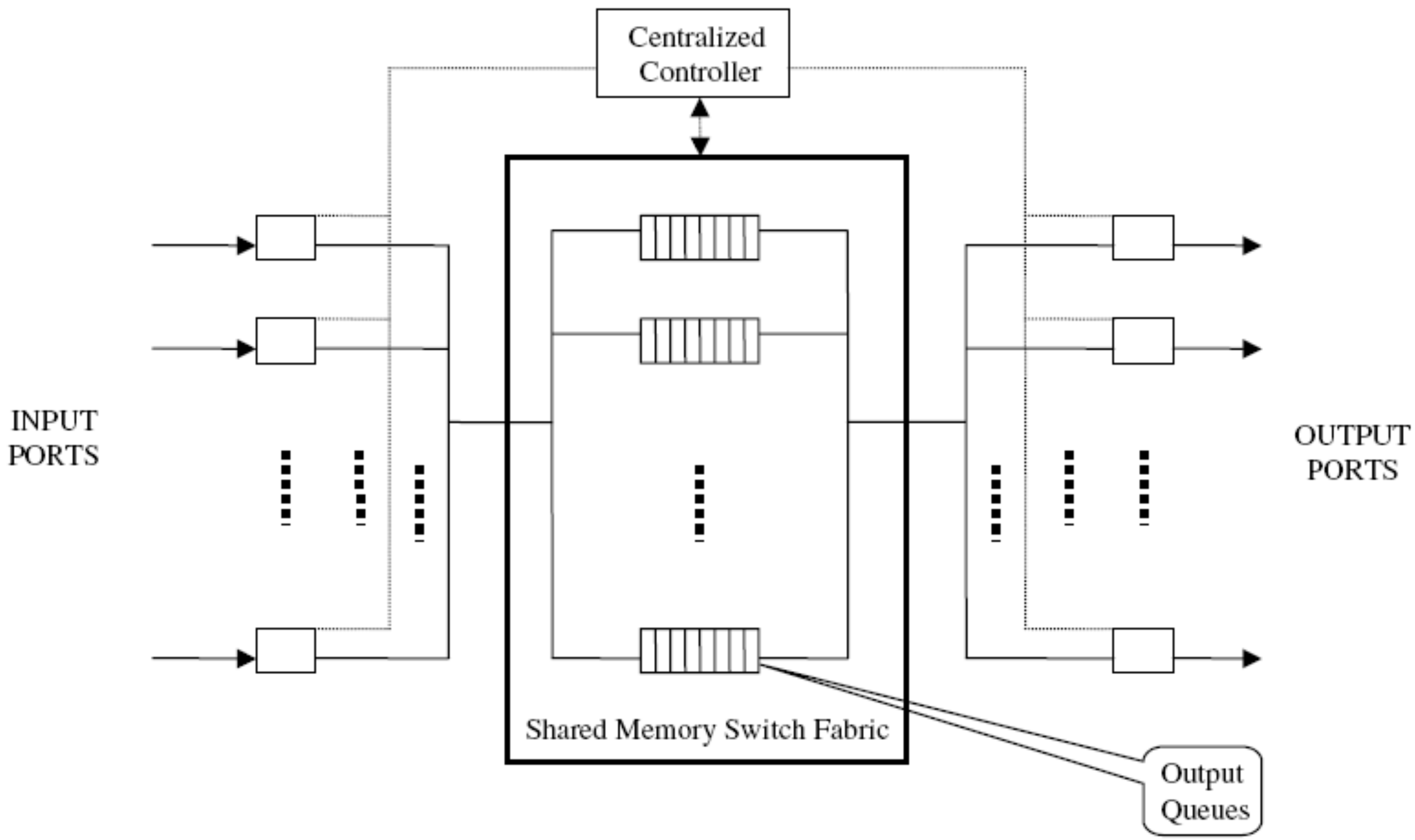
This Lecture: Network Layer

1. *Design goals and issues*
2. *Basic Routing Algorithms & Protocols*
3. *Addressing, Fragmentation and reassembly*
4. *Internet Routing Protocols and Inter-networking*
5. *Router design*
    1. *Short History + Router Archiectures*
    2. *Switching fabrics* ✔
    3. *Address lookup problem*
6. *Congestion Control, Quality of Service*
7. *More on the Internet's Network Layer*

Five Common Switch Fabric Designs

- Shared Memory
- Shared Medium
- Disjoint Paths
- Crossbar, Knockout Switch
- Multi-state Interconnection Network

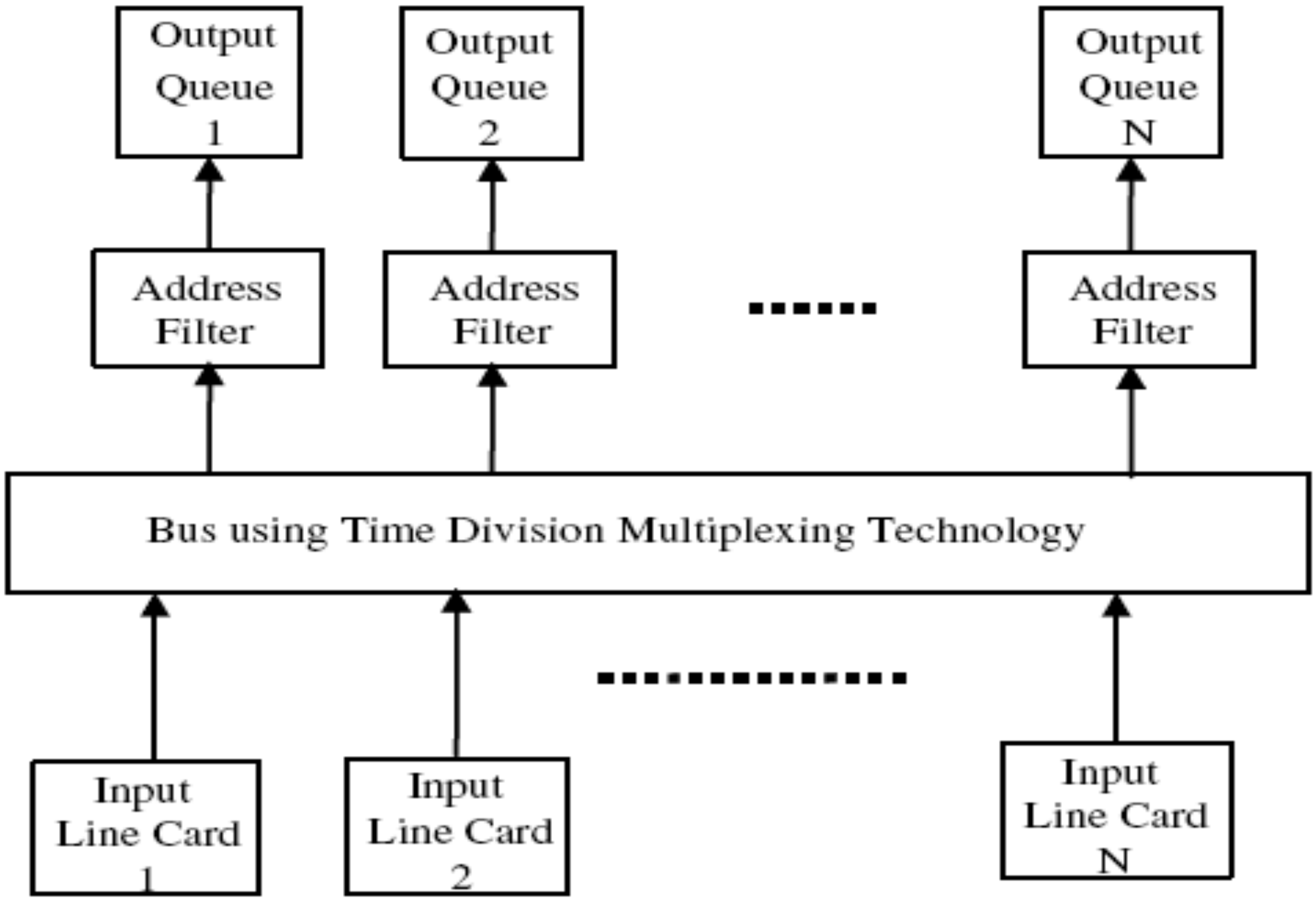
Shared Memory Switch



SMS: Pros and Cons

- *Pros*
  - Functionally an OQ switch, optimal throughput & delay
  - Can reduce total amount of memory needed
  - Broadcast/multicast ready
- *Cons*
  - Under "hot-spot" traffic, might be unfair
    - Can fix with separate memory segments per output
    - But then doesn't save as much memory
  - Need a controller & memory speedup of $2N$
  - Single point of failure
- *Commercial routers:*
  - Juniper Networks' E-series/ERX edge router
  - M-series/M20, M40, M160 core routers

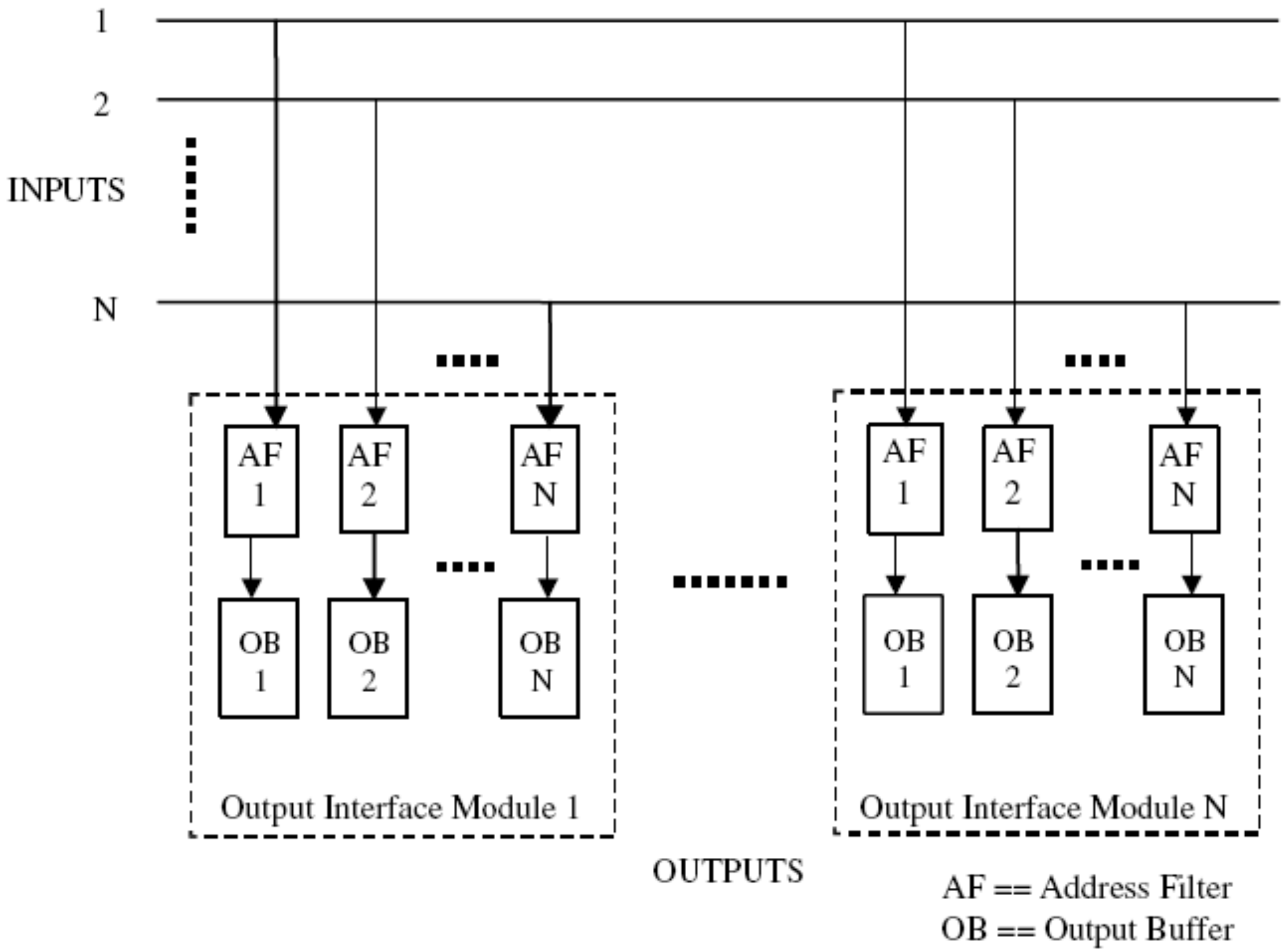
Shared Medium Switch



SMedS: Pros and Cons

- *Pros*
  - Functionally an OQ switch, optimal throughput & delay
  - TDM bus technology is well-understood & advanced
  - Broadcast/multicast ready
- *Cons*
  - Need speedup of *(N+1)* for output memory, *N* for filter
  - Can also be unfair under "hot-spot" traffic, need sophisticated scheduling/balancing algorithm
  - Single point of failure
- *Commercial routers:*
  - Cisco 7500 series

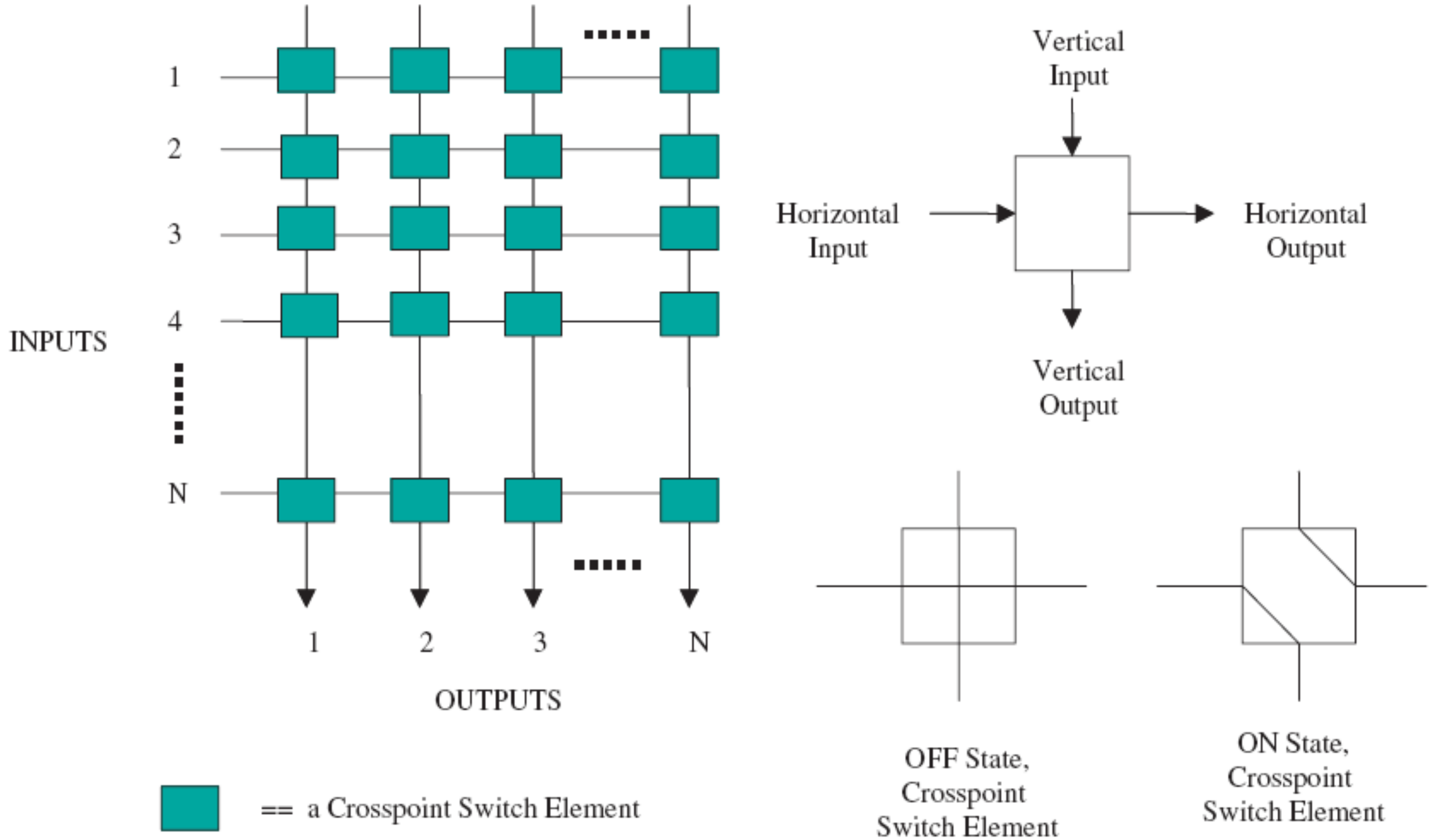
Disjoint Paths Switch



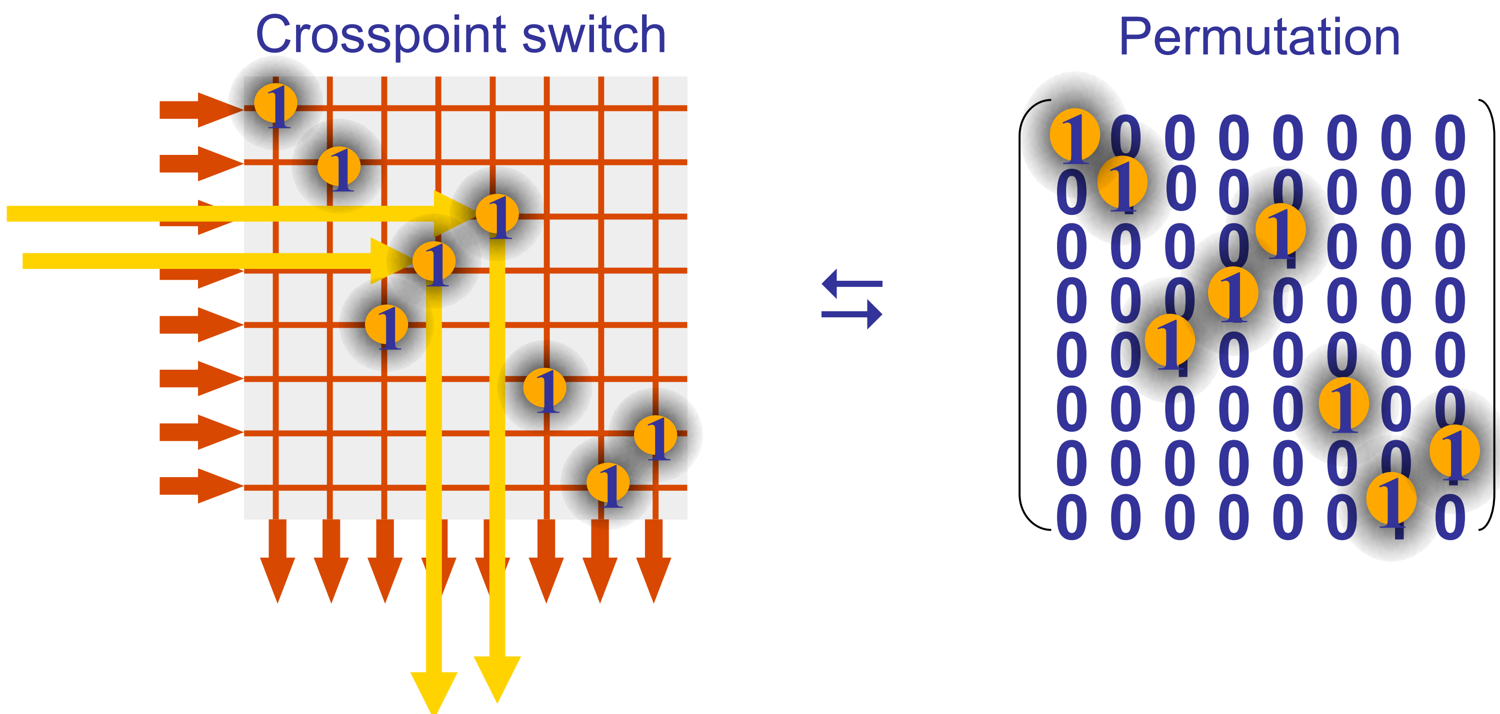
DPS: Pros and Cons

- *Pros*
  - Functionally an OQ switch, optimal throughput & delay
  - No contention of any kind (neither input nor output)
  - No “mechanical” speedup needed
  - Broadcast/multicast ready
  - Suited for both bursty & uniform traffics
  - Fault tolerant, Straightforward implementation
- *Cons*
  - Complexity scales as $O(N^2)$, can’t make large switches
  - (Too much memory)
- *Commercial routers:*
  - Cisco 7500 series

Crossbar/Crosspoint Switch



Crosspint Switch



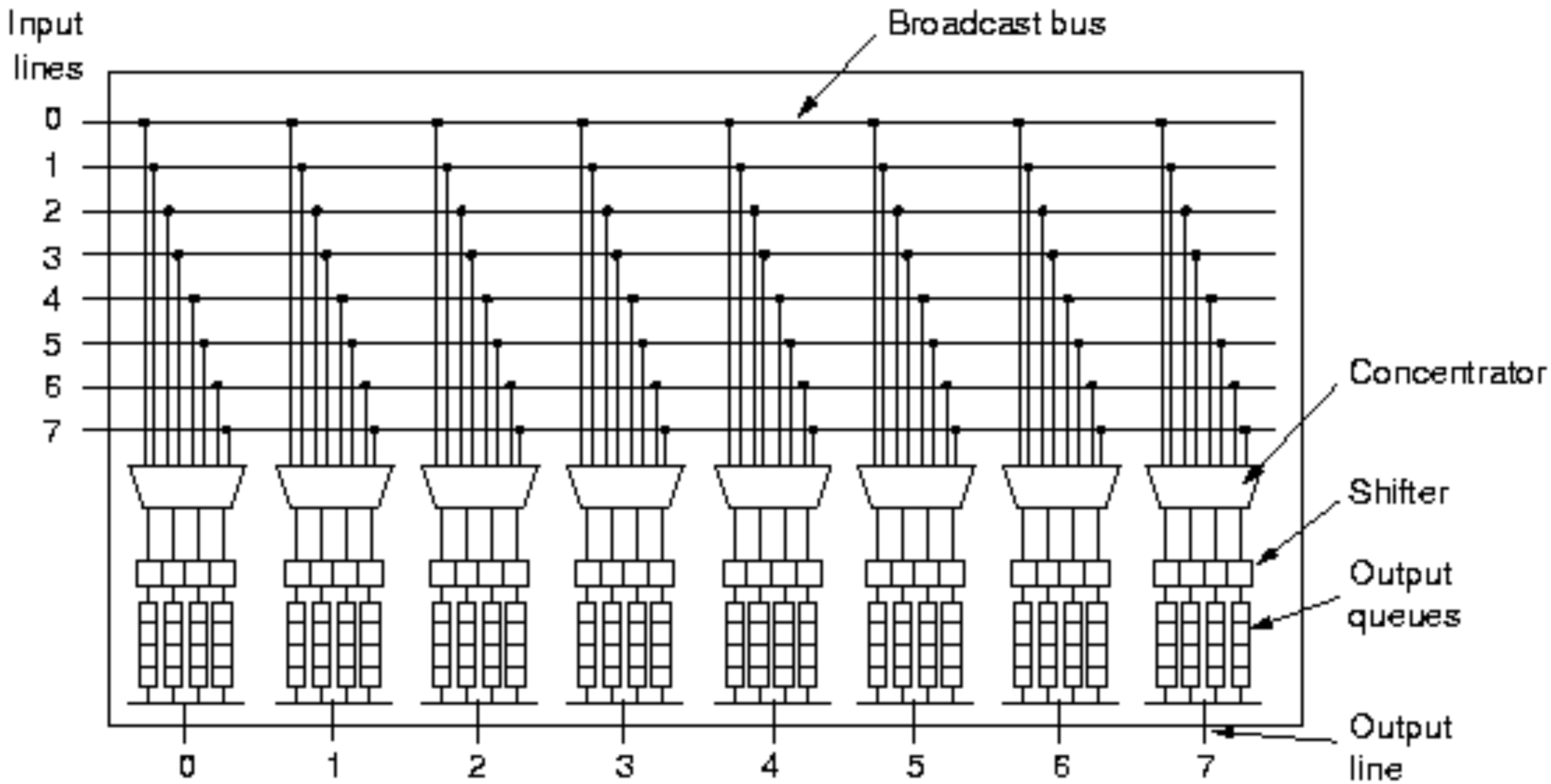
A crosspoint switch supports all permutations
So it is “non-blocking”
But it needs N^2 crosspoints

Crossbar: Pros and Cons

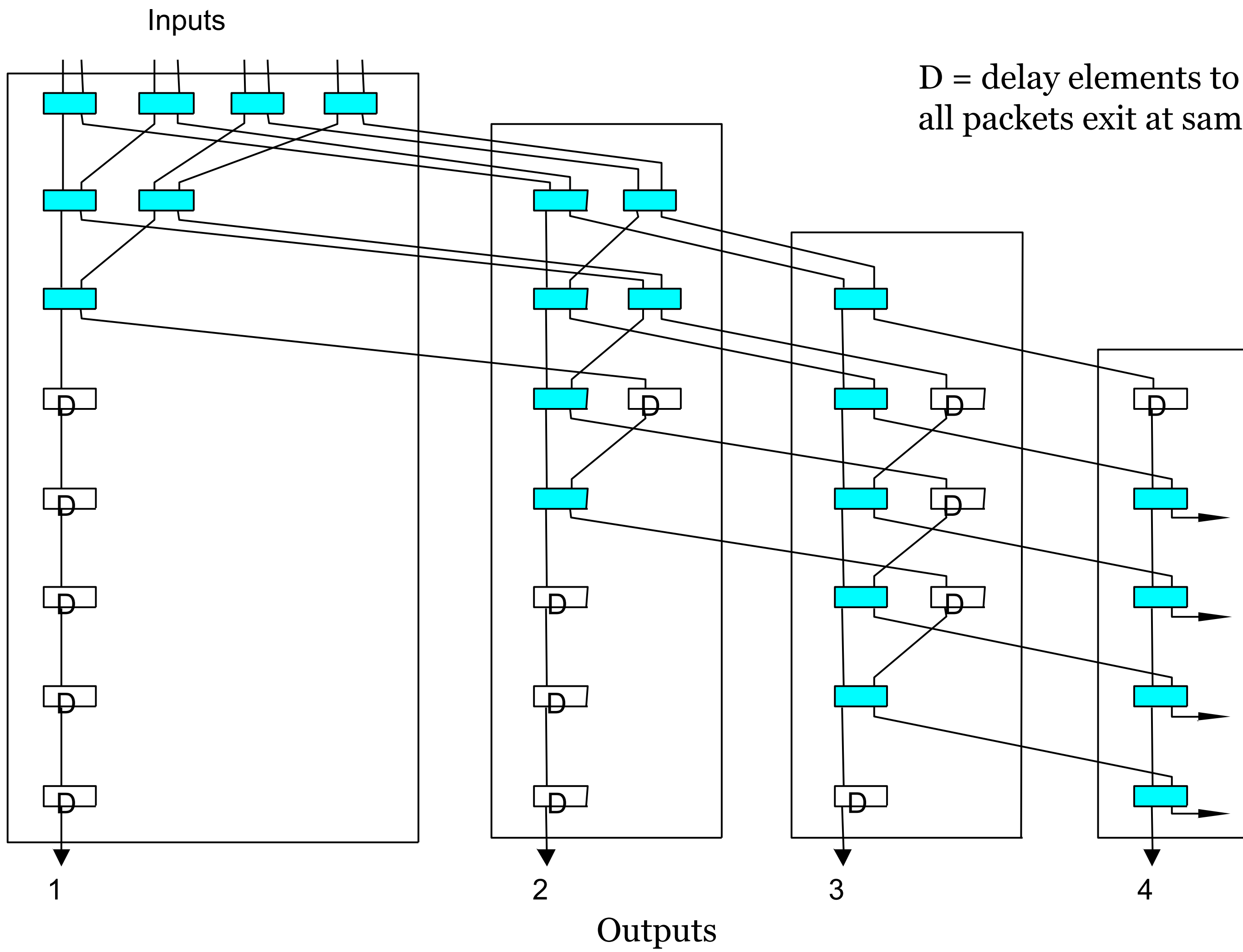
- *Pros*
  - Simple control, internally non-blocking
  - Can perform well, depending on how buffers are managed
  - Can be used to build larger switches
- *Cons*
  - Complexity scale as $O(N^2)$, can't make large switches
  - Can multicast, but require sophisticated scheduling
- *Commercial routers*
  - IQ-crossbar: Cisco 12416
  - CIOQ-crossbar: Lucent's PacketStar 6400 IP Switch
  - Lucent GRF 400 Multi-gigabit Router
  - Foundry Network's Big Iron

Knockout Switch

Basic idea: use a *concentrator* to reduce buffer size



Concentrator: Select a Few from Many

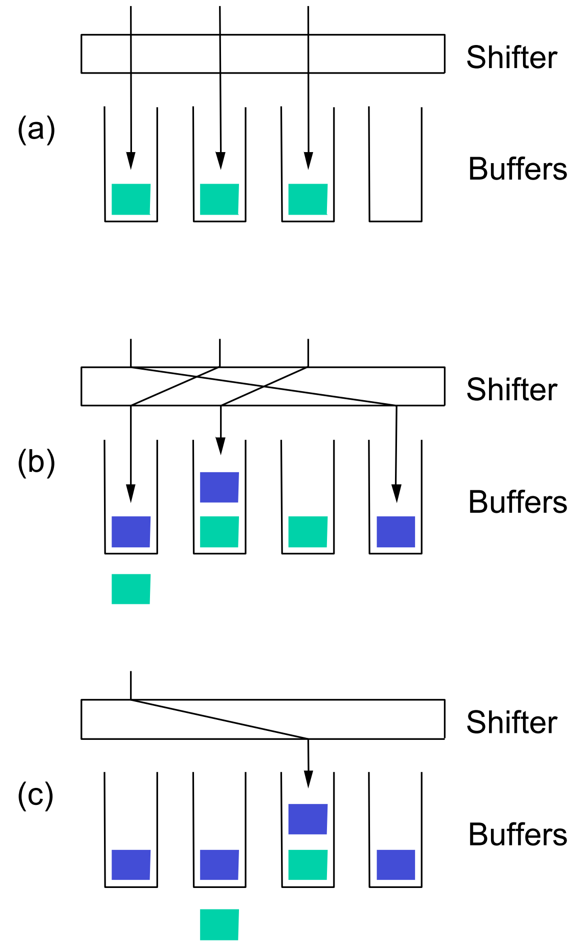


D = delay elements to ensure all packets exit at same time

Shifter: Balance Output Buffers

Physically, the shifter can be implemented with an L×L Banyan network

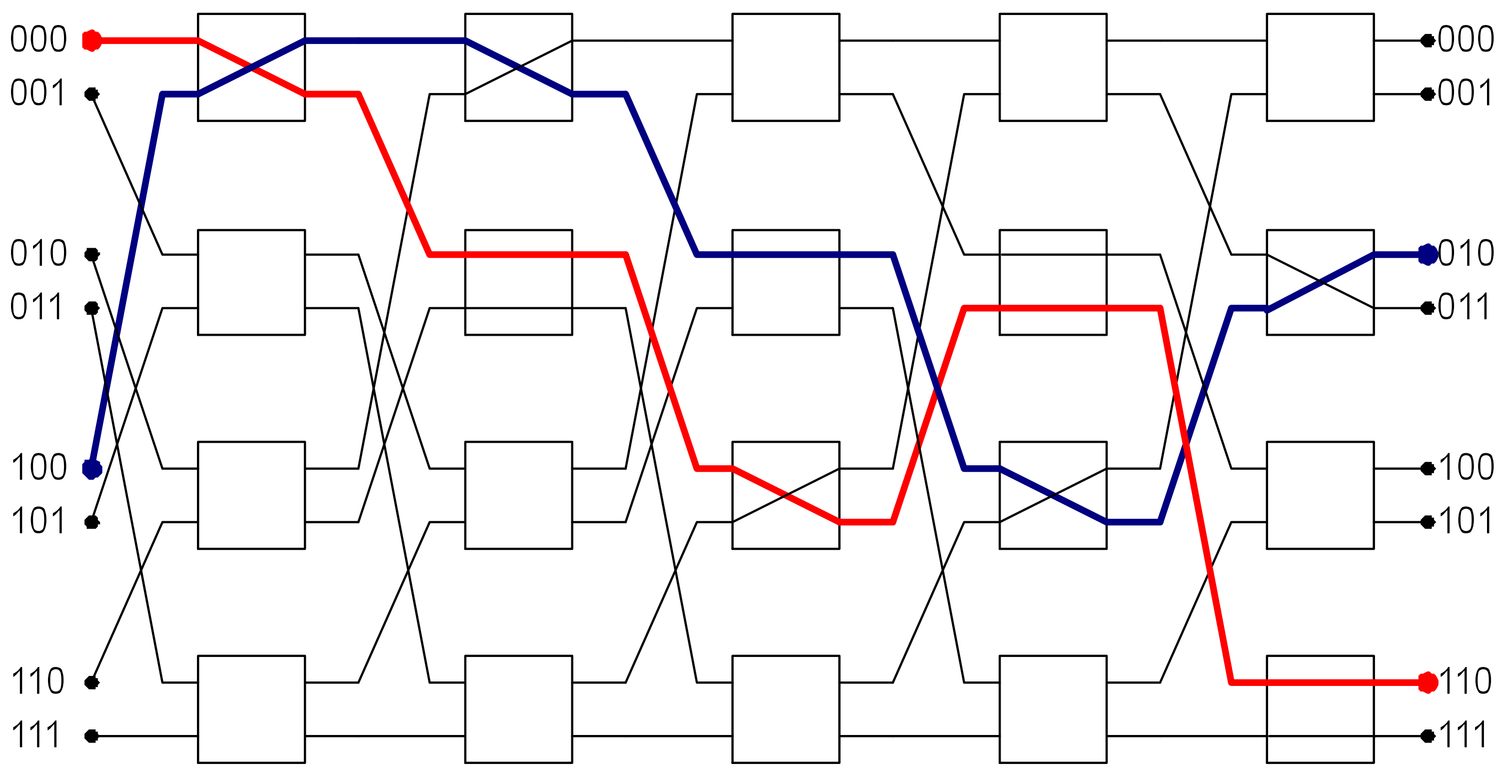
L is the number of outputs of the concentrator



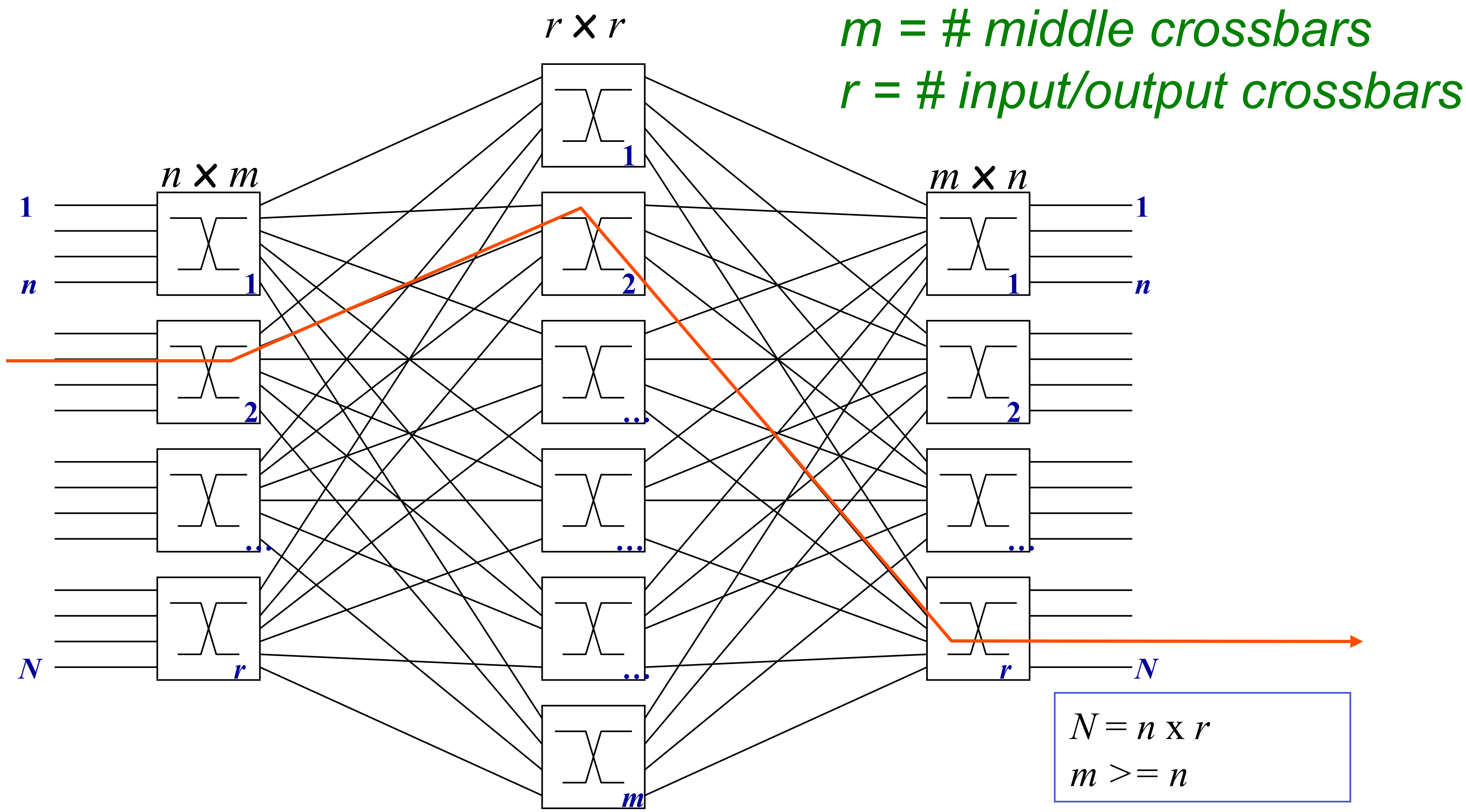
Multi-state Interconnection Networks

Can we build functionally crossbar-equivalent switch using significantly fewer than N^2 2x2 switching elements (or crosspoints?)

- Yes! Theoretically we can even achieve *$O(N \log N)$*
- Practically: a little worse – *$O(N \log^2 N)$* – with, e.g., Clos and Banyan types of topologies



3-Stage Clos Network – C(n,m,r)

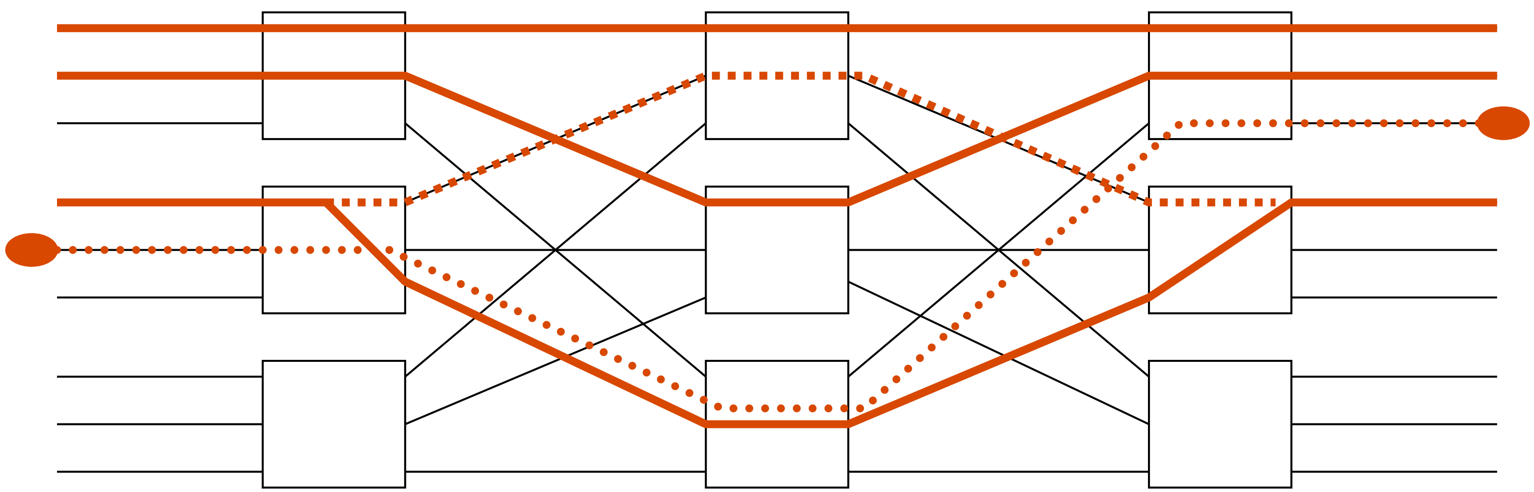


This is a class of networks, as we can vary n, m, k

Rearrangeably Nonblocking Condition

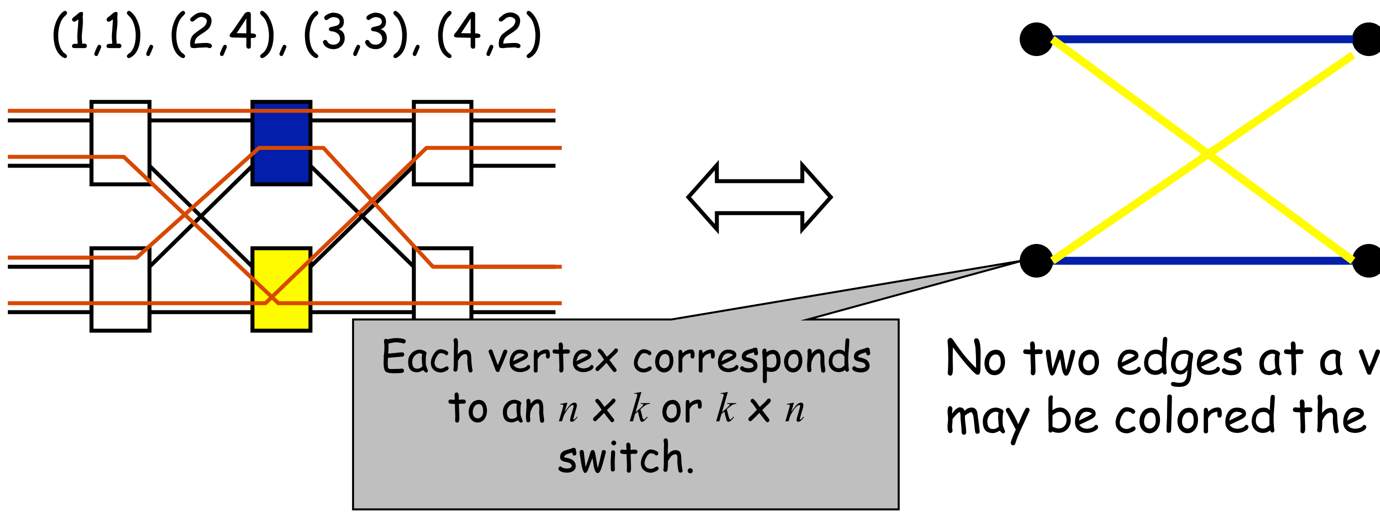
- A switch is rearrangeably nonblocking if it can route any (sub)-permutation of inputs to outputs simultaneously

Theorem: *C(n,m,r) is rearrangeably nonblocking if and only if $m \geq n$; In particular, C(n,n,r) is!*



Proof of Theorem

Routing matches is equivalent to edge-coloring in a bipartite multigraph. Colors correspond to middle-stage switches.



No two edges at a vertex may be colored the same.

<u>Konig 1931:</u> a D-degree bipartite graph can be colored in D colors. Therefore, if $k = n$, a 3-stage Clos network is rearrangeably non-blocking (and can therefore perform any permutation).

Is C(n,n,r) better than a crossbar?

- Given N inputs, how to choose *n* and *r*?
- Total # of crosspoints is

$$2rn^2 + nr^2 = N(2N/r + r) \geq 2\sqrt{2}N^{3/2}$$

- Can be achieved if we choose

$$r \approx \sqrt{2N}, n \approx \sqrt{N/2}$$

- So the answer is YES

The Price: Rearrangement Running Time

- *Method 1*: Find a maximum size bipartite matching for each of D colors in turn:

$$O(DM\sqrt{N}) = O(DN^{2.5}) \text{ worst case}$$

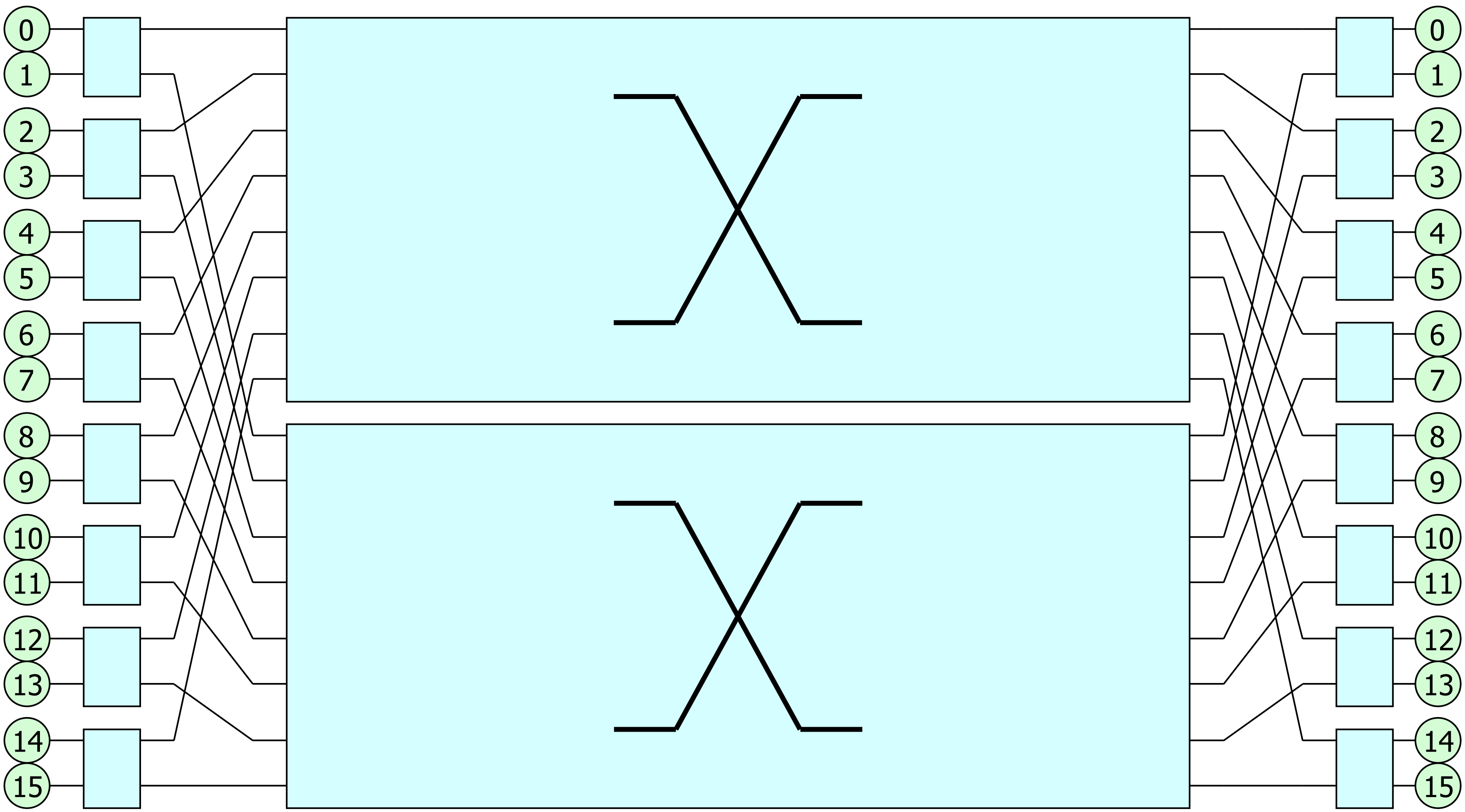
- *Method 2*: Partition graph into Euler sets [Cole et al. '00]

$$O(M \log D) = O(N^2 \log D) \text{ worst case}$$

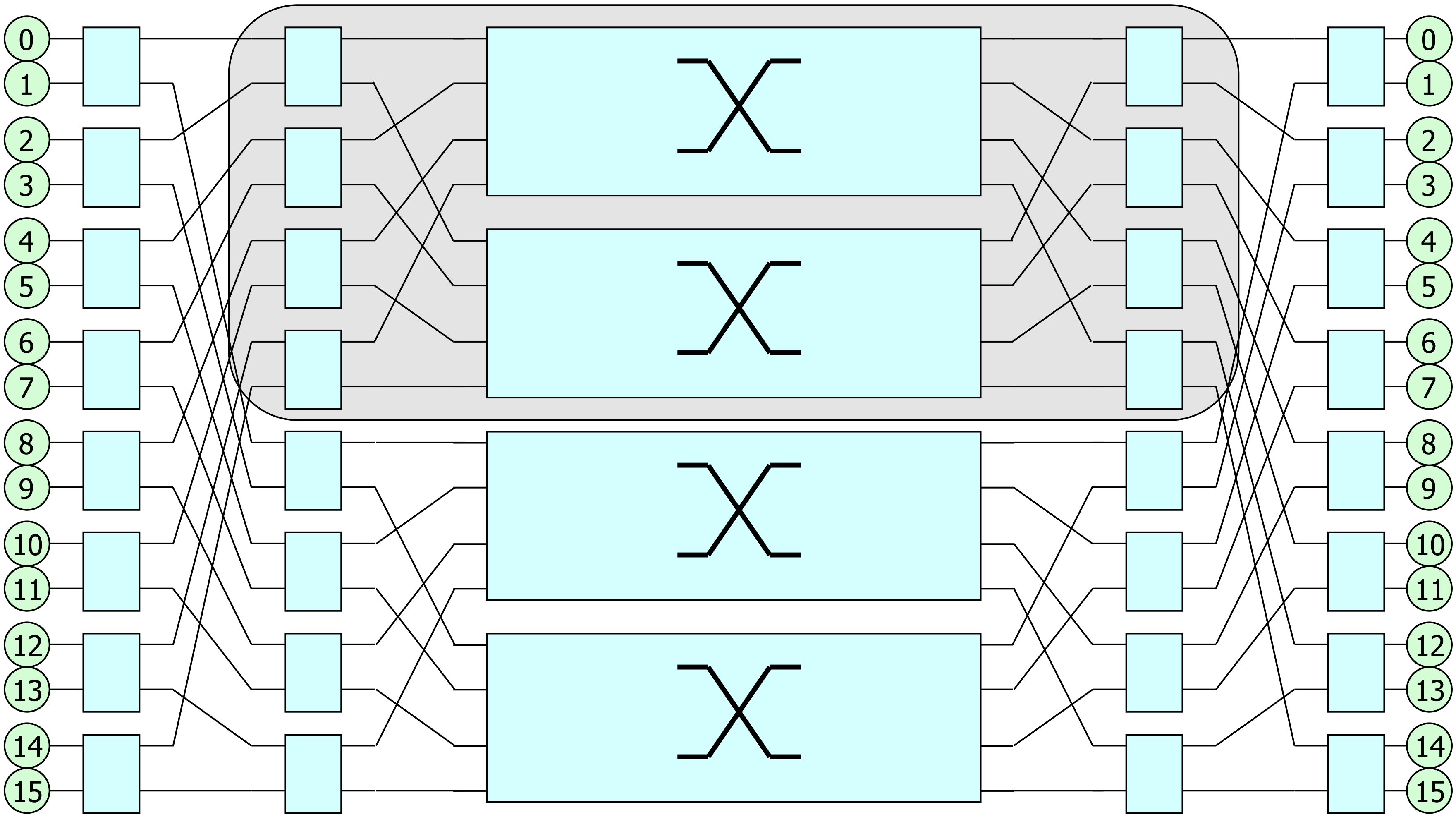
- Both are slow and complex

Benes Network

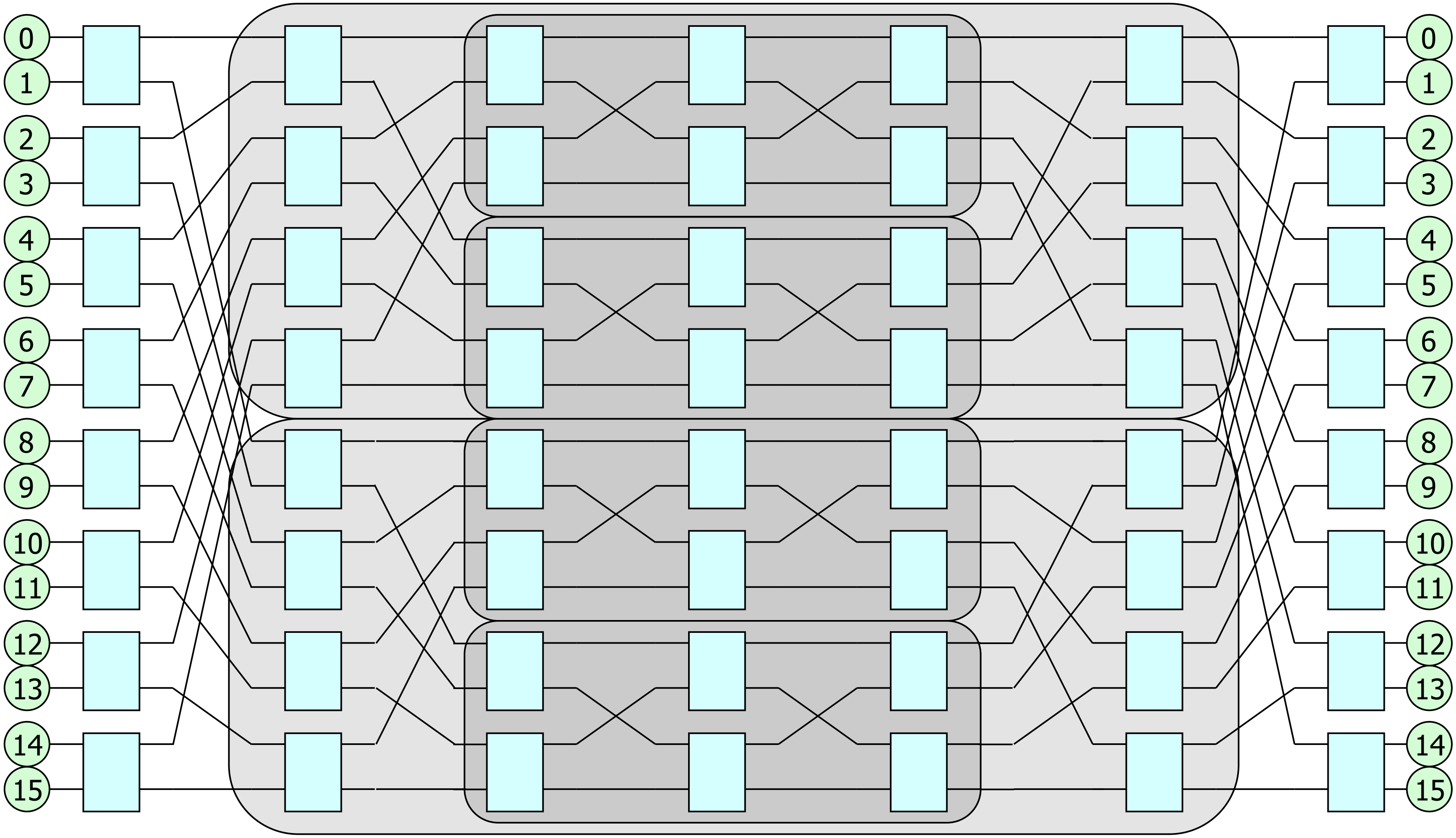
- Can we do better than $O(N^{3/2})$ for rearrangeability?
- Yes: use Clos recursively



Benes Network – Recursive Construction



Benes Network – Recursive Construction



16 port, ***7 stage Clos*** network = ***Benes topology***

Benes Network Complexity

- Symmetric
- Size:
  - $F(N) = 2(N/2) + 2F(N/2) = O(N \log N)$
- It is rearrangable
  - Clos network with m=n=2

Rearrangeable Clos: Pros & Cons

Pros

- A rearrangeably non-blocking switch can perform any permutation
- A cell switch is time-slotted, so all connections are rearranged every time slot anyway

Cons

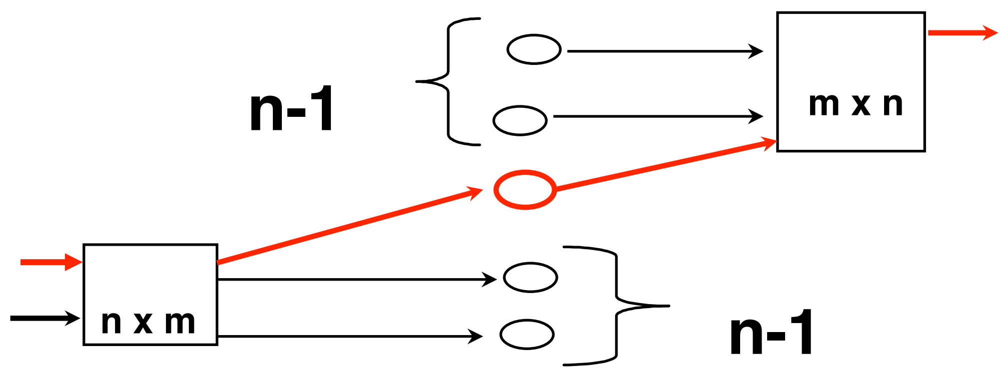
- Rearrangement algorithms are complex (in addition to the scheduler)

Can we eliminate the need to rearrange?

Strictly Non-blocking Clos Network

- A switch is *strictly non-blocking* if a new request from a free input to a free output can be accommodated without disturbing existing connections

Theorem: *C(n,m,r) is strictly nonblocking if and only if* $m \geq 2n-1$



Strictly Non-blocking Clos: Complexity

- Given N inputs, how to choose n and r?
- Total # of crosspoints is (set $m = 2n$ for simplicity)

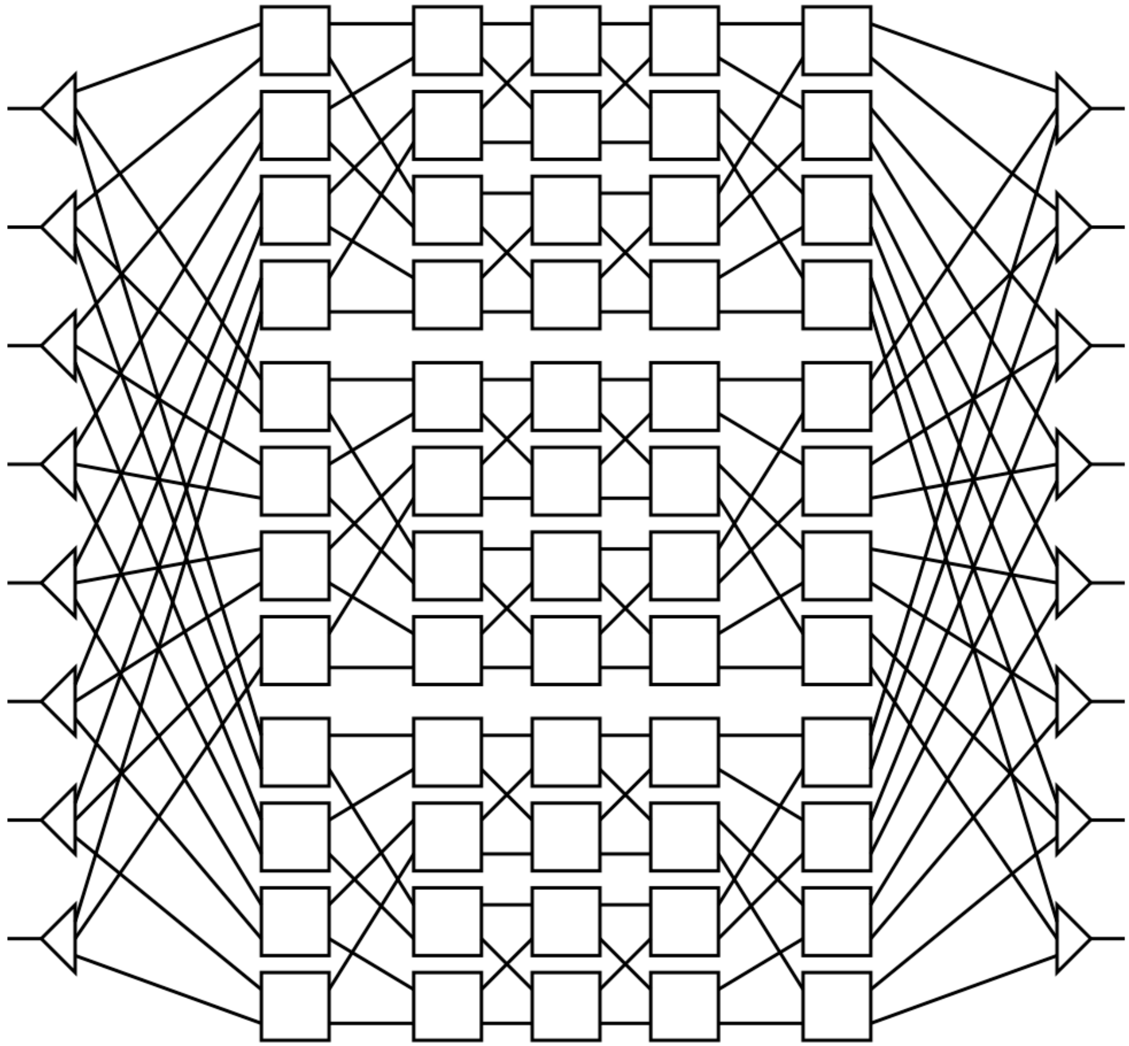
$$4n^2r + 2nr^2 = 2N(2N/r + r) \geq 4\sqrt{2}N^{3/2}$$

- Can be achieved if we choose

$$r \approx \sqrt{2N}, n \approx \sqrt{N/2}$$

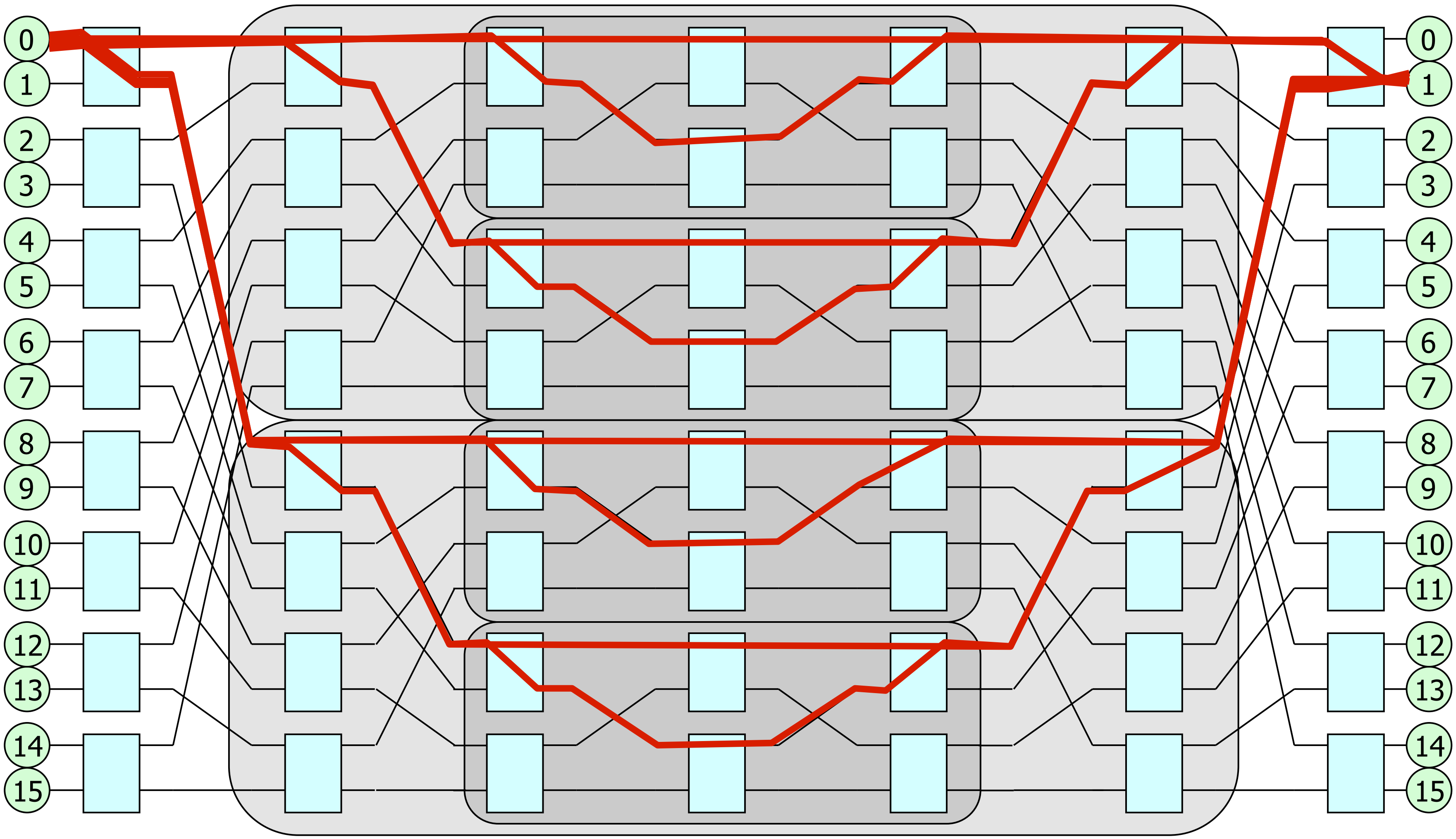
- Seem a little high. Can we do better?

Cantor Network – Strictly nonblocking

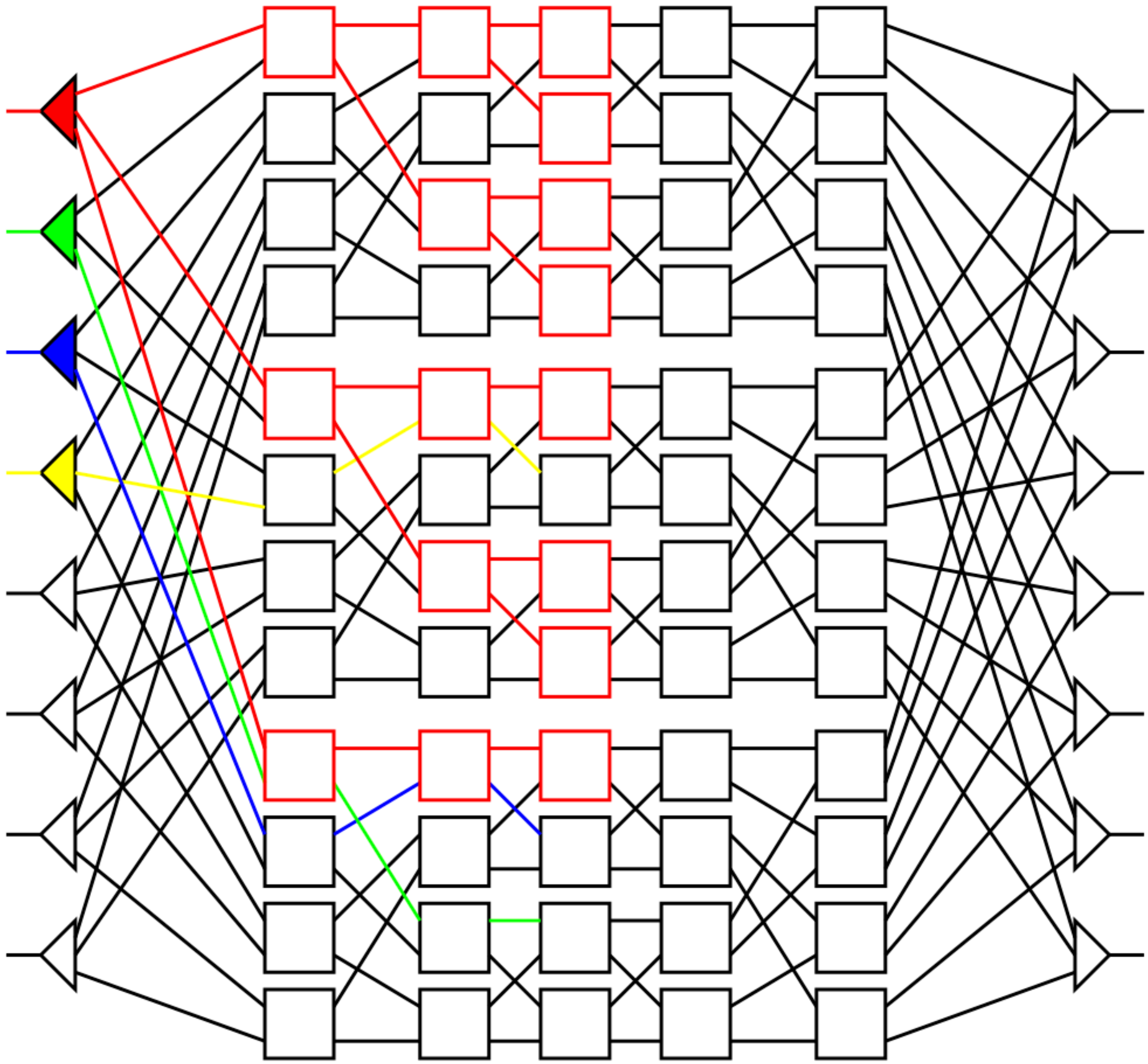


Log N copies of Benes, complexity $O(N \log^2 N)$

Proof Sketch



Proof Sketch



Proof Sketch

- Benes network:
  - 2 log N -1 layers,
  - N/2 nodes in layer.
  - Middle layer= layer log N -1
- Consider the middle layer of the Benes Networks.
- There are Nm/2 nodes in in all of them combined.
- Bound (from below) the number of nodes reachable from an input and output.
- If the sum is more than Nm/2:
  - There is an intersection
  - there has to be a route.

Proof Sketch

- Let A(k) = number of nodes reachable at level k.
- $A(0)=m$
- $A(1)= 2A(0)-1$
- $A(2)=2A(1)-2$
- $A(k)=2A(k-1) - 2^{k-1} = 2^k A(0) - k\, 2^{k-1}$
- $A(\log N -1) = Nm/2 - (\log N -1) N/4$
- Need that: $2A(\log N -1) > Nm/2.$
  - $2[Nm/2 - (\log N -1) N/4] > Nm/2.$
- Hold for $m > \log N-1$.