EE382C
Lecture 4

High-Radix and Non-Blocking Networks

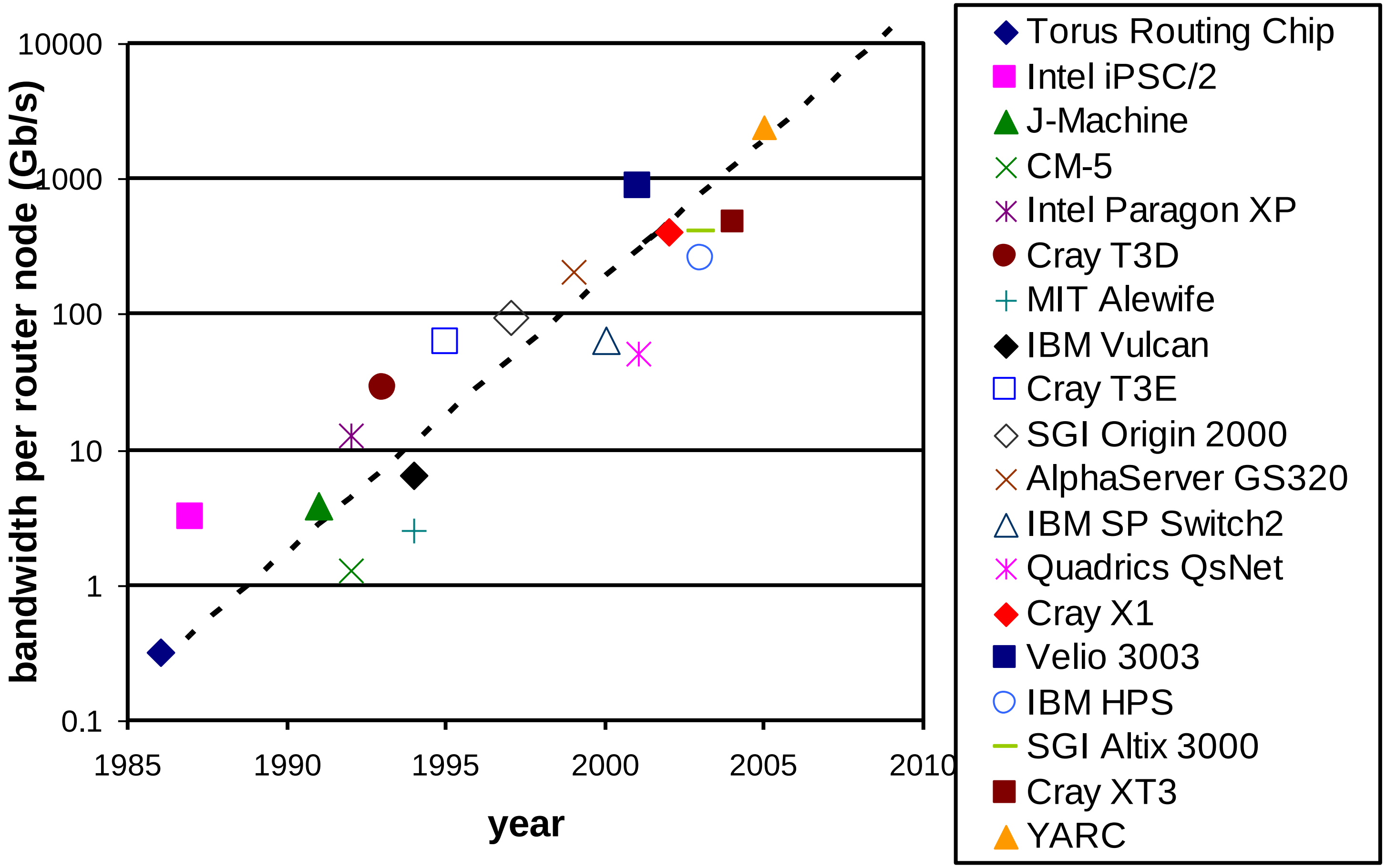
4/7/11

Question of the day

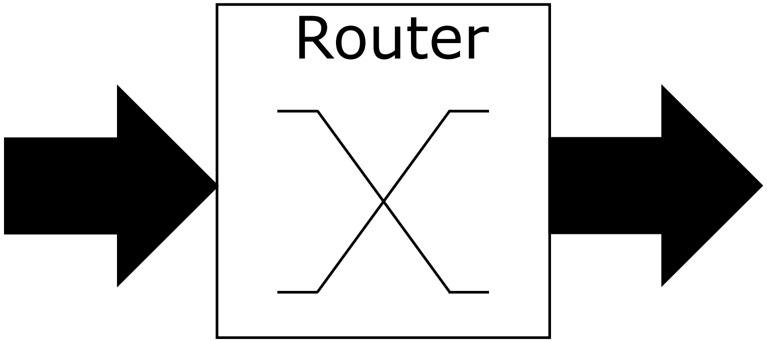
- What topology has an average hop count, H_{avg} for *load-balanced traffic* that is *close* to the $\log_{d/2}N$ bound and is able to route *arbitrary traffic* with an H_{max} of twice this amount?

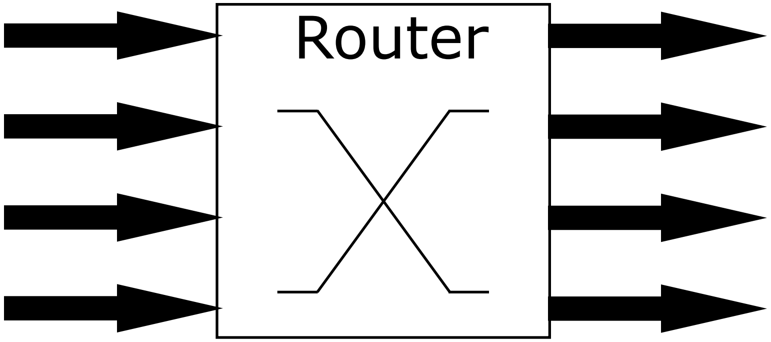
High-Radix Networks

Bandwidth Trend (ISCA '05)

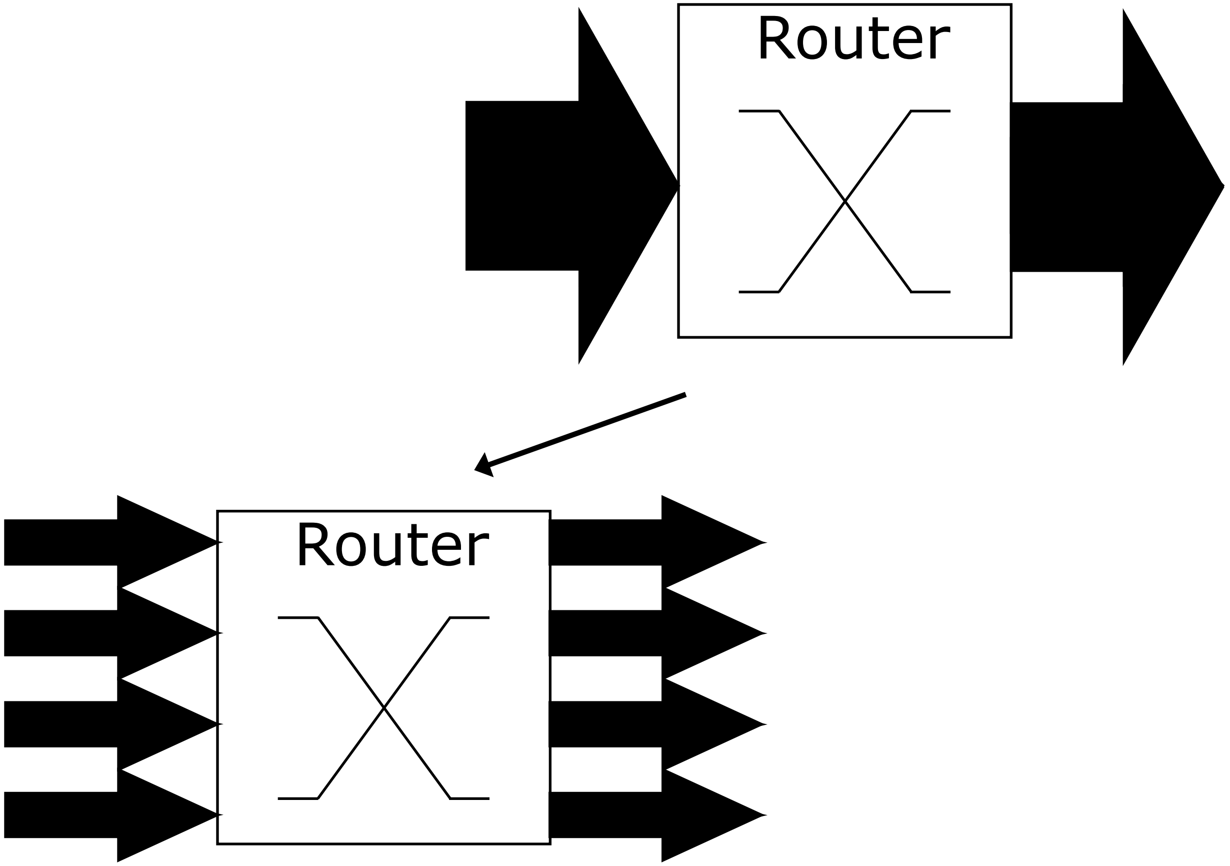


Router bandwidth

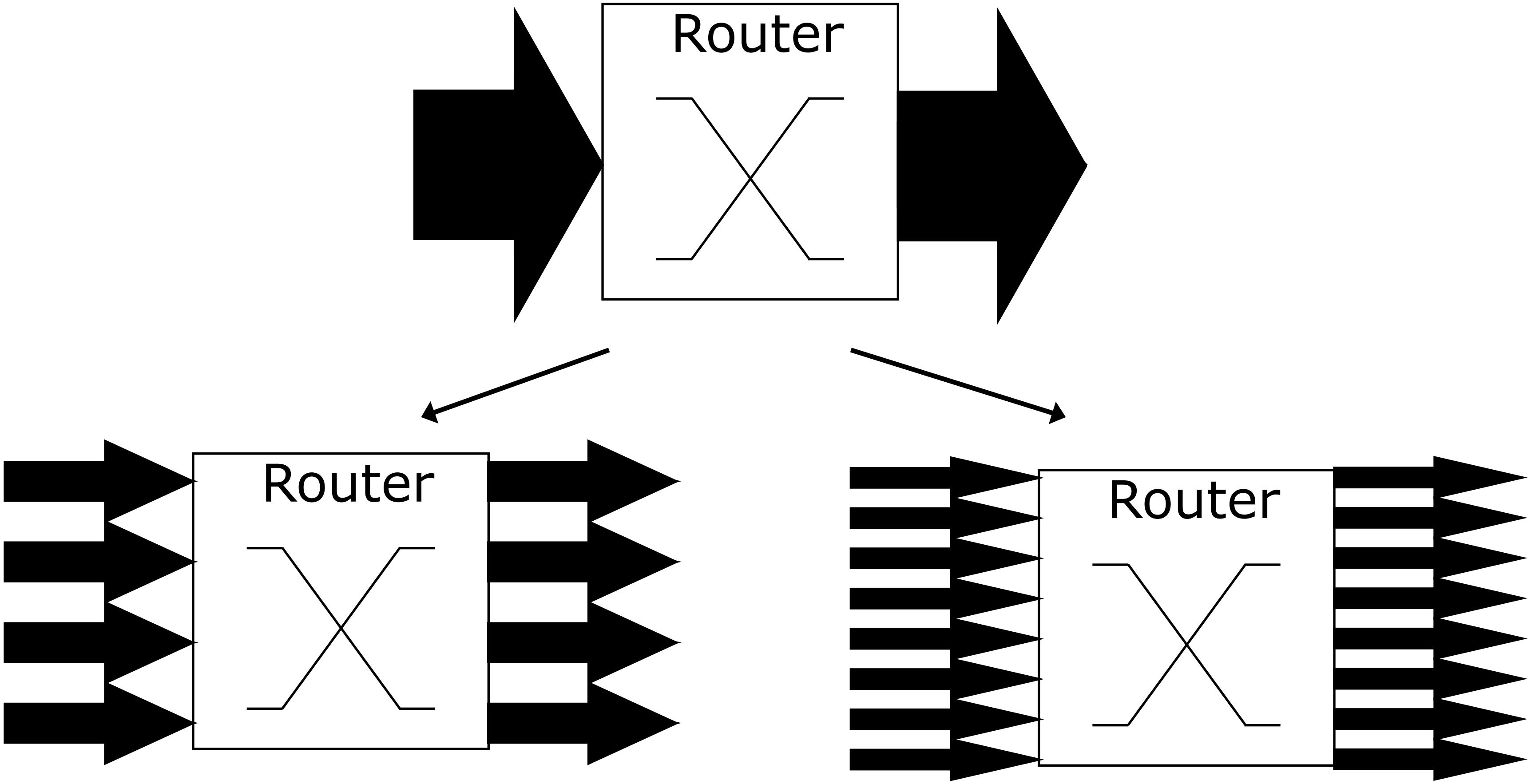




As bandwidth increases …

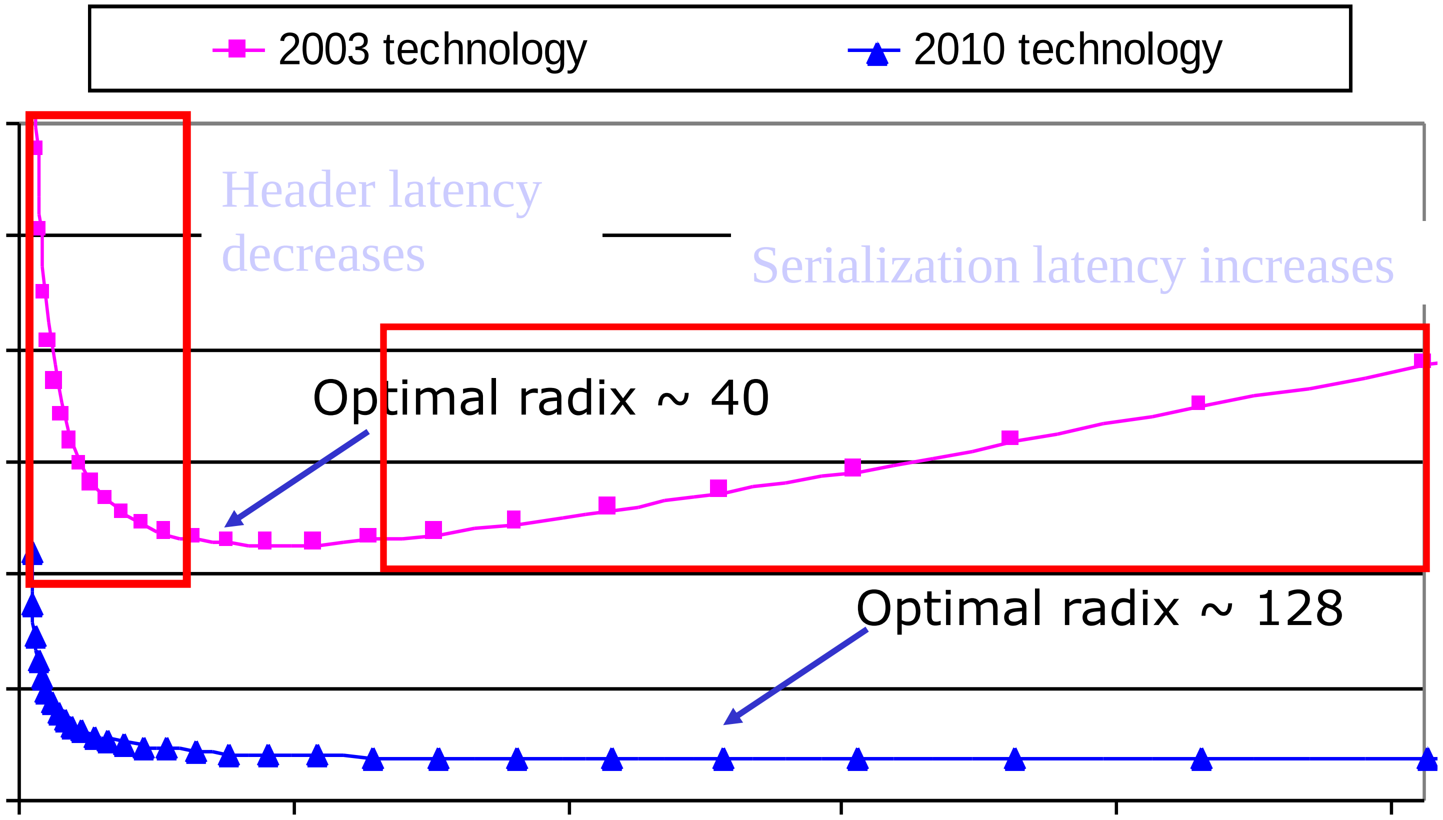


Low-Radix vs. High-Radix Router



Low-radix (small number of fat ports) High-radix (large number of skinny ports)

Latency vs. Radix



Determining Optimal Radix

Latency = Header Latency + Serialization Latency

= $H\, t_r + L / b$

= $2t_r \log_k N + 2kL / B$

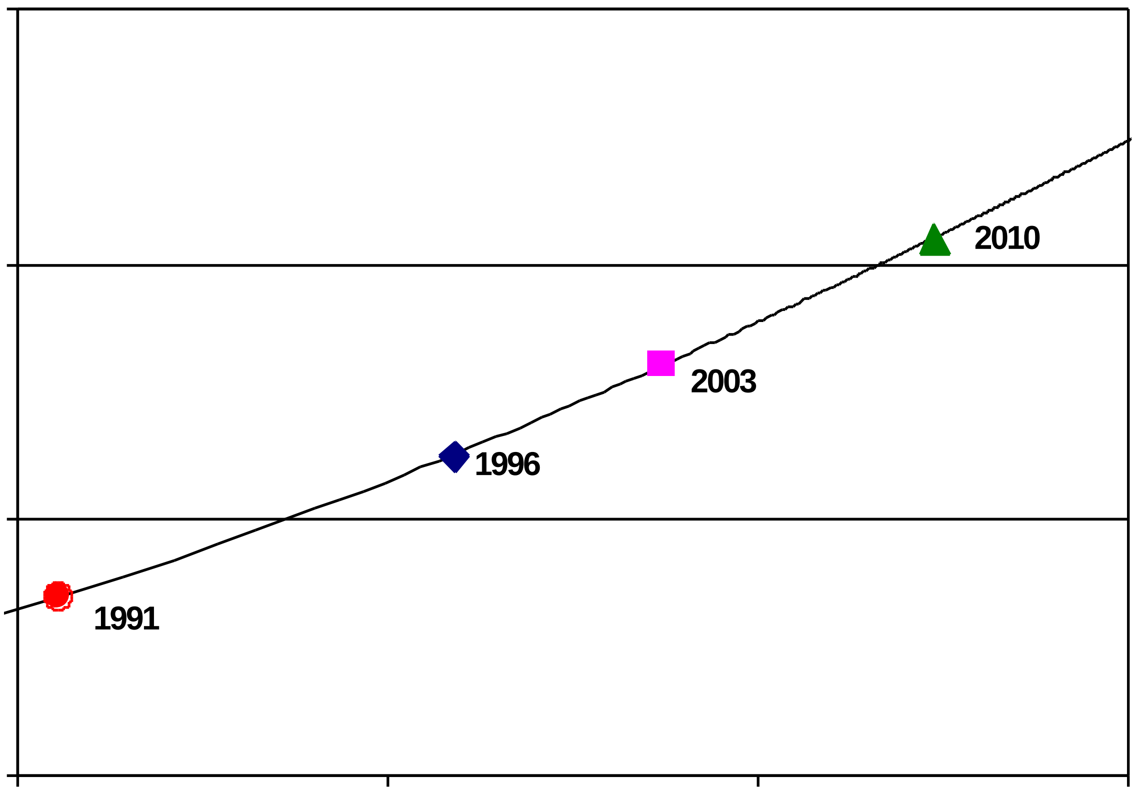
where k = radix
B = total Bandwidth
N = # of nodes
L = message size

Optimal radix

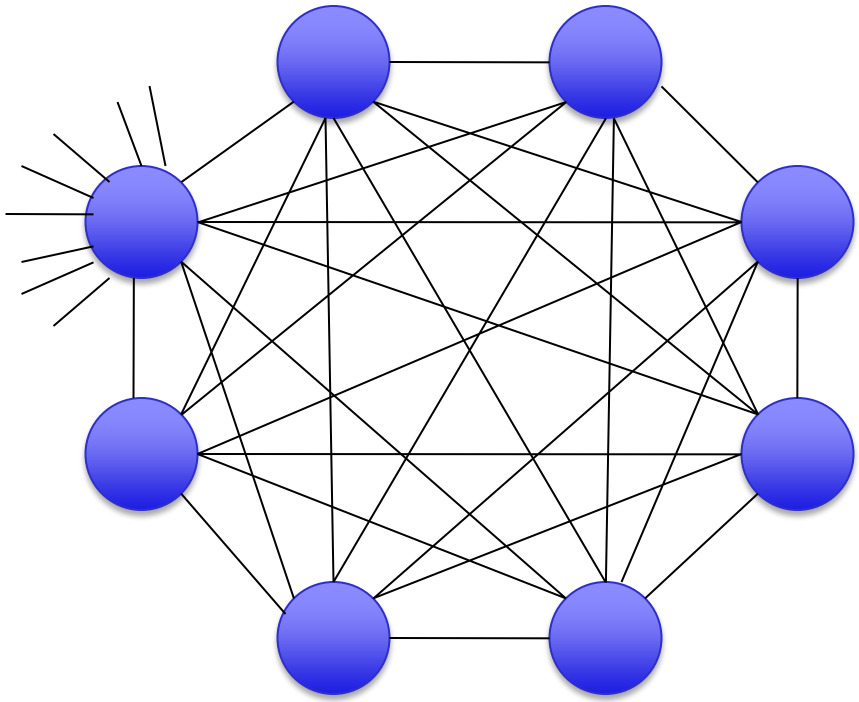
➔ $k \log_2 k = (B\, t_r \log N) / L$

Aspect Ratio

Higher Aspect Ratio, Higher Optimal Radix

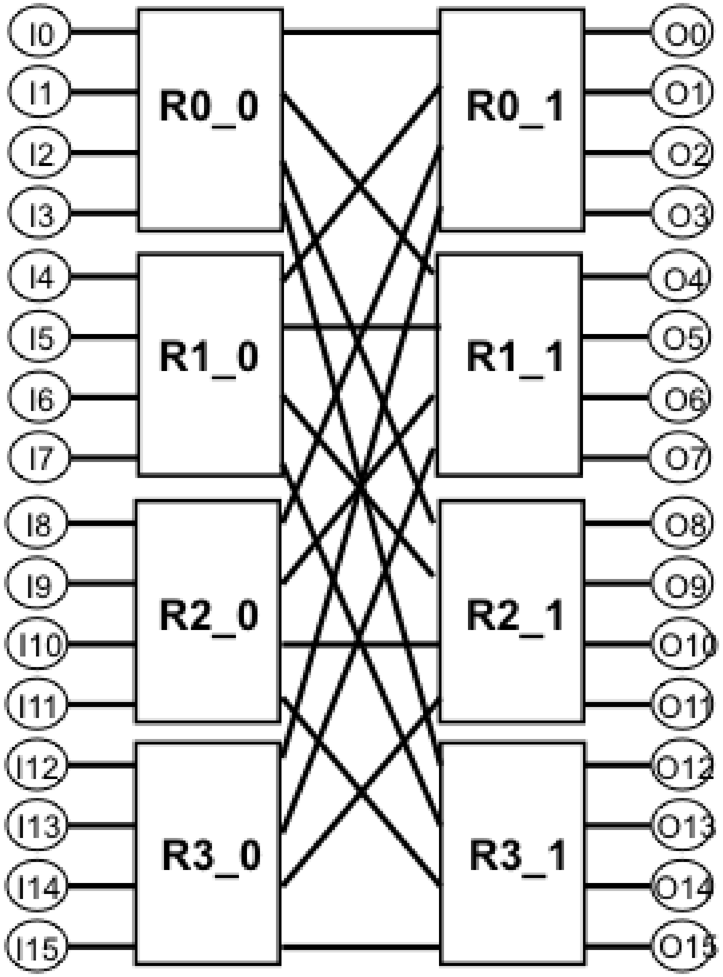


K_N – The Ultimate High-Radix Network



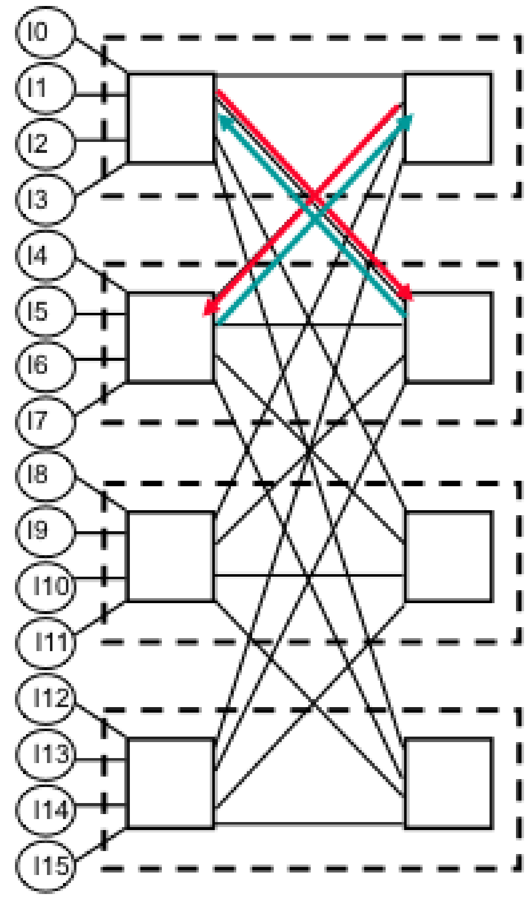
High-Radix Butterfly

- Just build a butterfly with large k
- For k = 128
  - 128 in 1 stage
  - 16K in 2 stages
  - 2M in 3 stages
- But – vulnerable to adversarial traffic

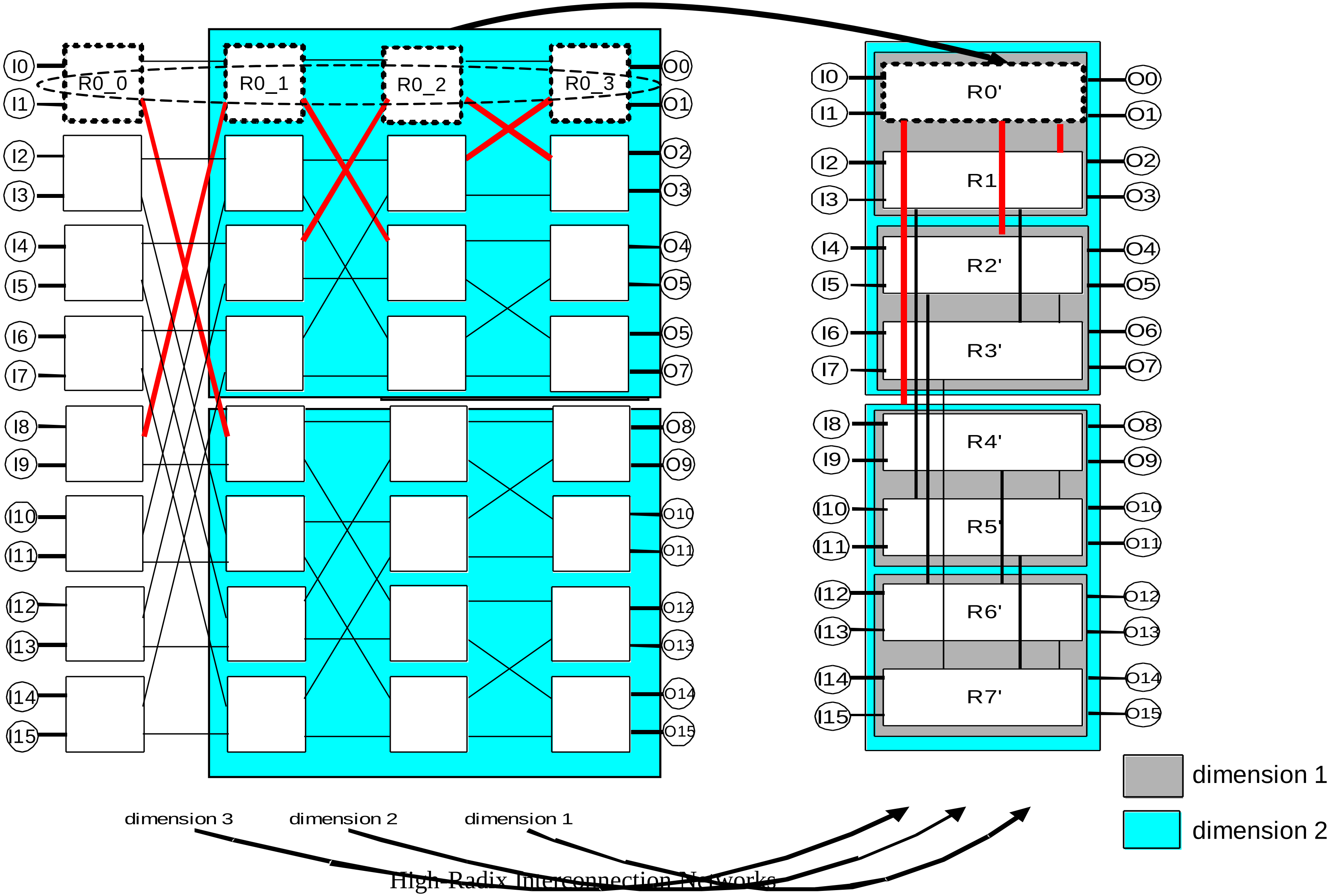


High-Radix Clos

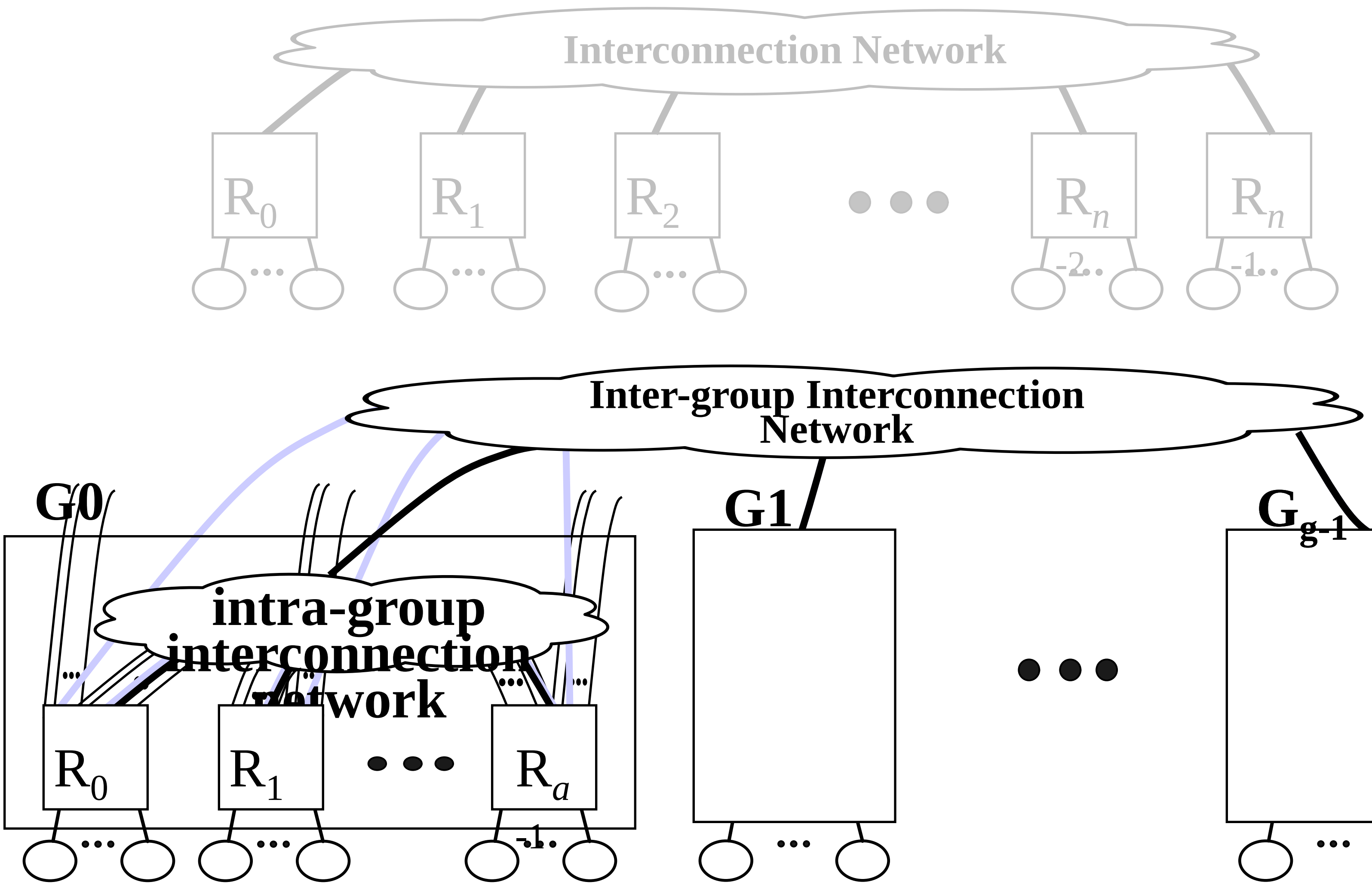
- Not vulnerable to adversarial traffic,
- But twice the number of stages – even on benign traffic.



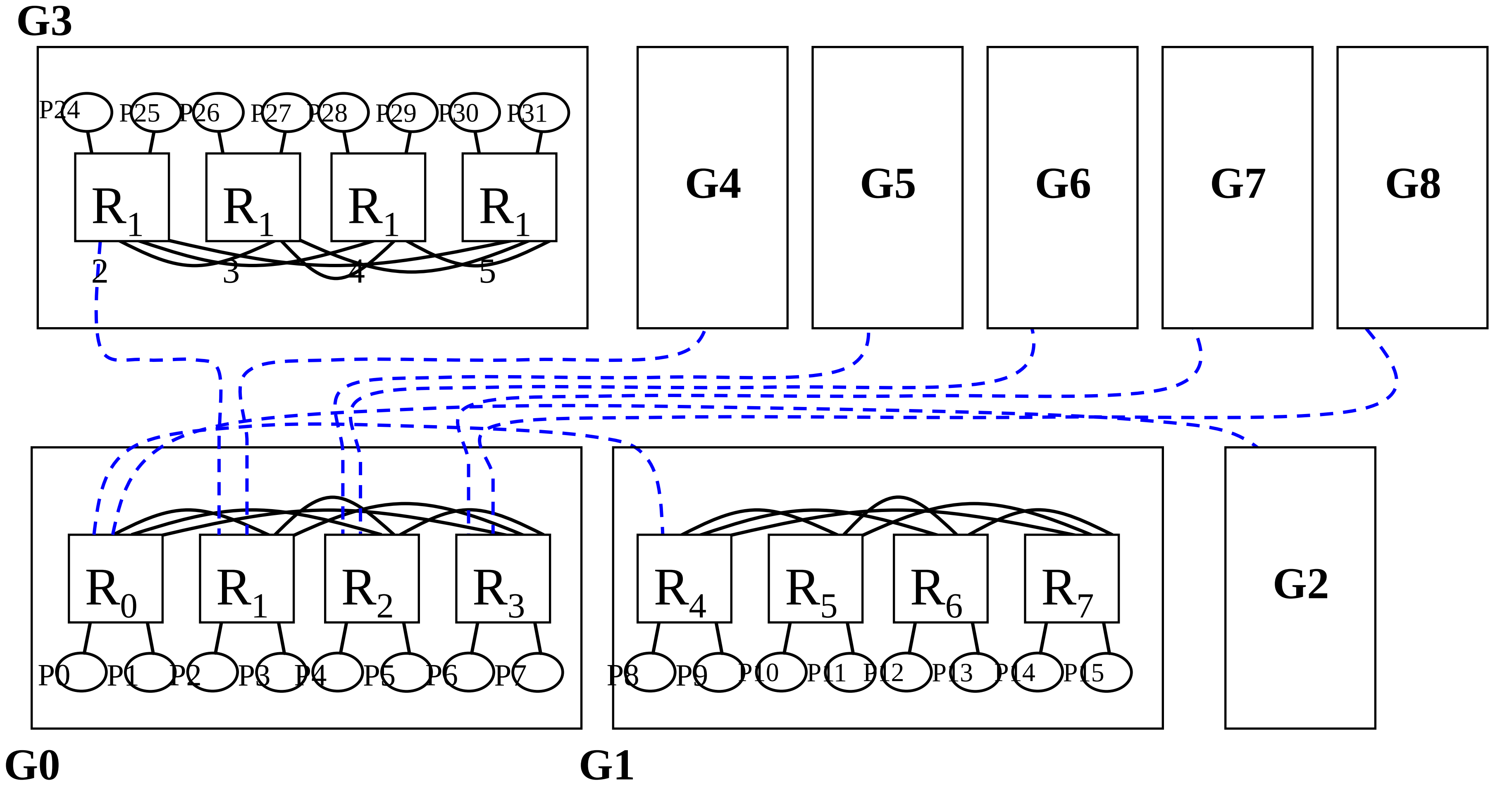
Flattened Butterfly



Dragonfly Topology



Dragonfly Topology Example



Dragonfly Topology

Dragonfly by the numbers

- Consider k=128
- Divide into g, l, p
  - p processors per router
  - l connections to other routers in same group
  - g global connections – to other groups
- Design method
  - Pick g
  - Let l = 2g-1
  - Let p = 128-2g
  - Compute concentration factor c = p/g

An Example

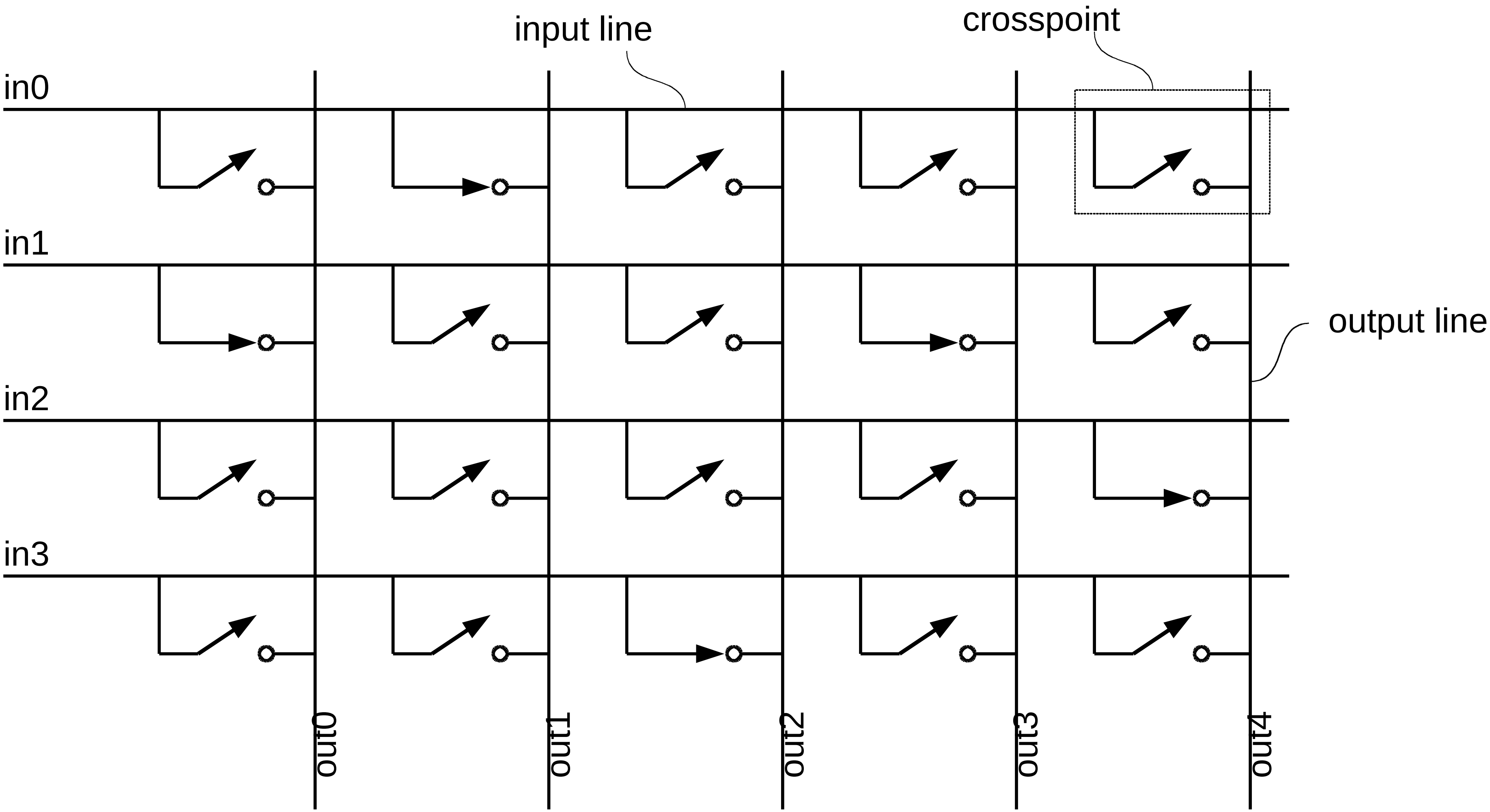
d	128	128	128	128
p	93	75	51	33
l	23	35	51	63
g	12	18	26	32
p/g	7.75	4.17	1.96	1.03
(l+1)g	288	648	1352	2048
group	2139	2625	2601	2079
N	618,171	1,703,625	3,519,153	4,259,871

Non-Blocking Networks

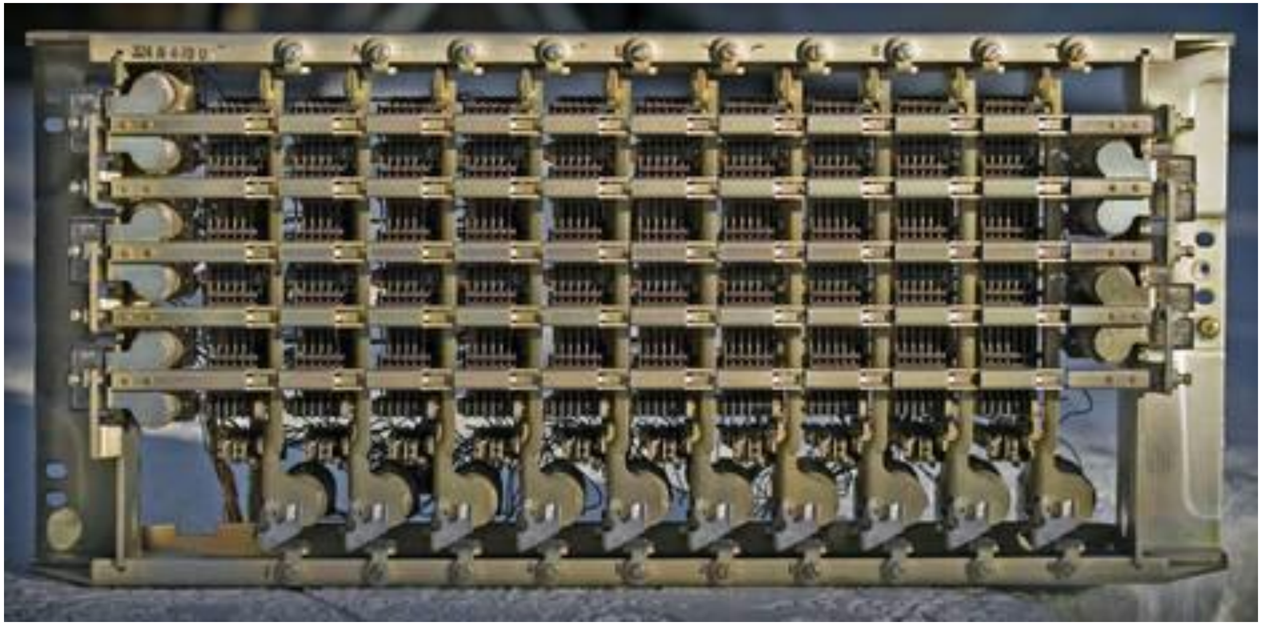
Non-blocking networks

- Non-blocking: able to connect any unconnected input to any unconnected output.
- Circuit switching vs. packet switching
- Strictly vs. rearrangeably non-blocking
- Non-interfering networks

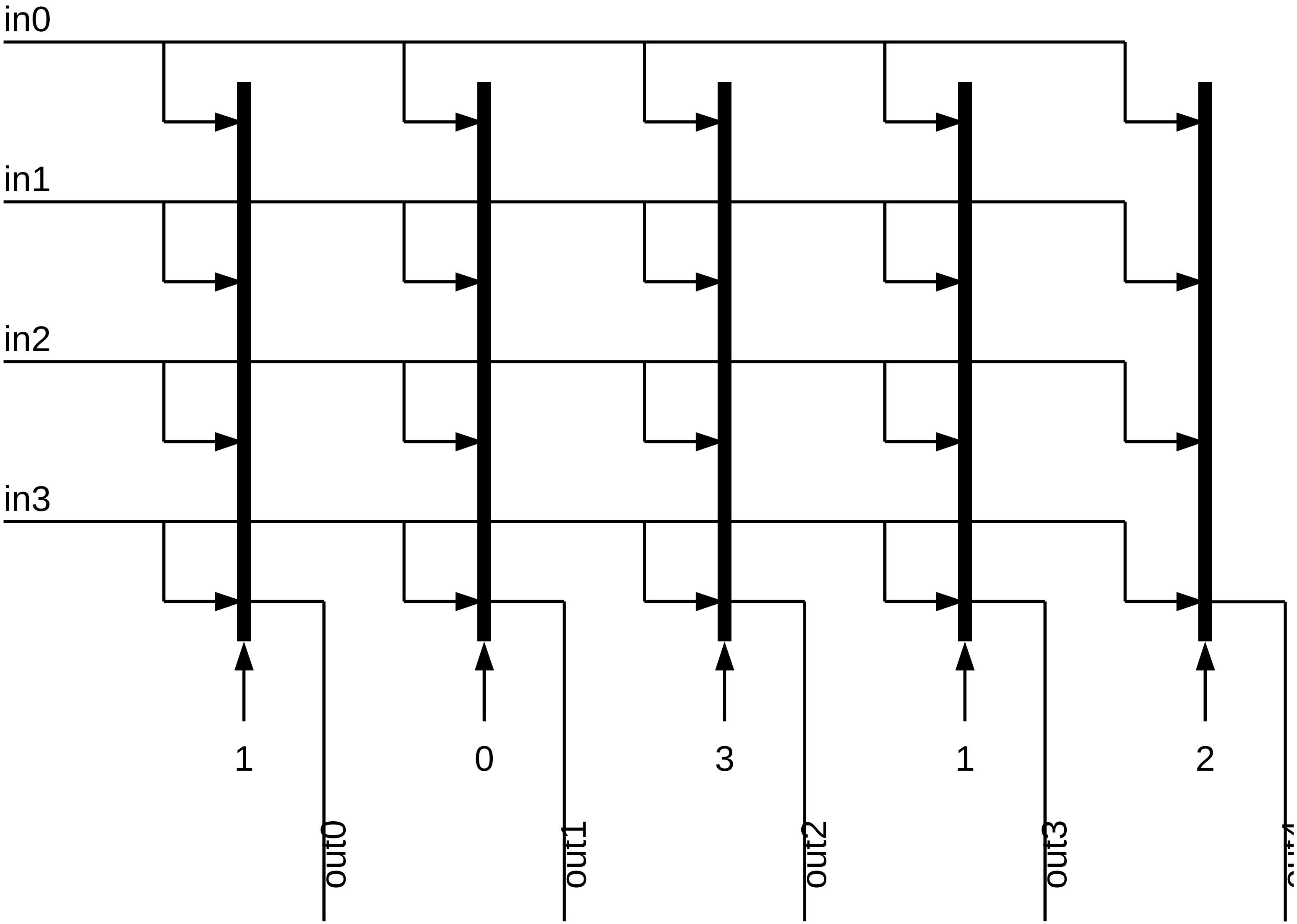
Crossbar Switch



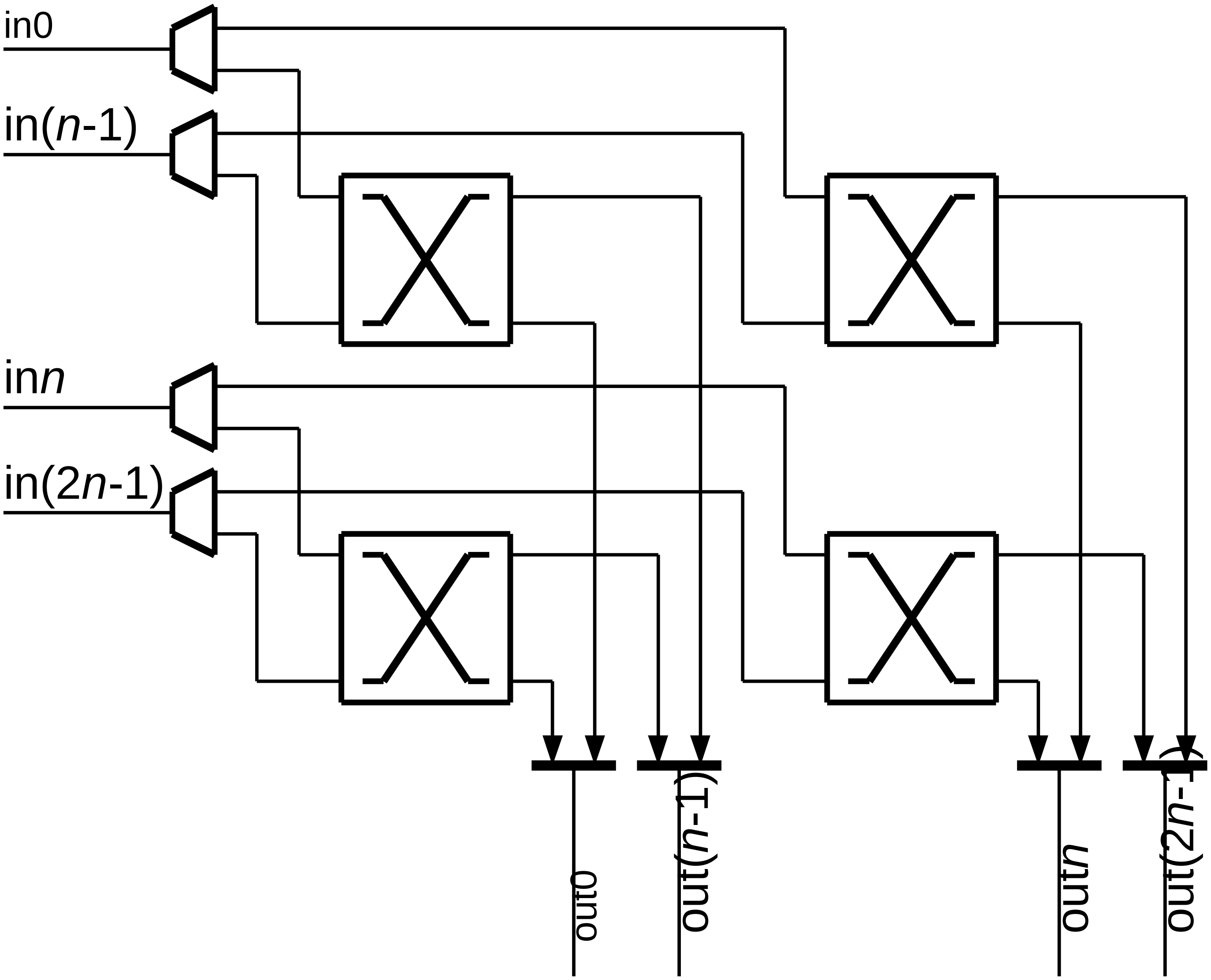
Old Mechanical Crossbar



Implemented with Multiplexers

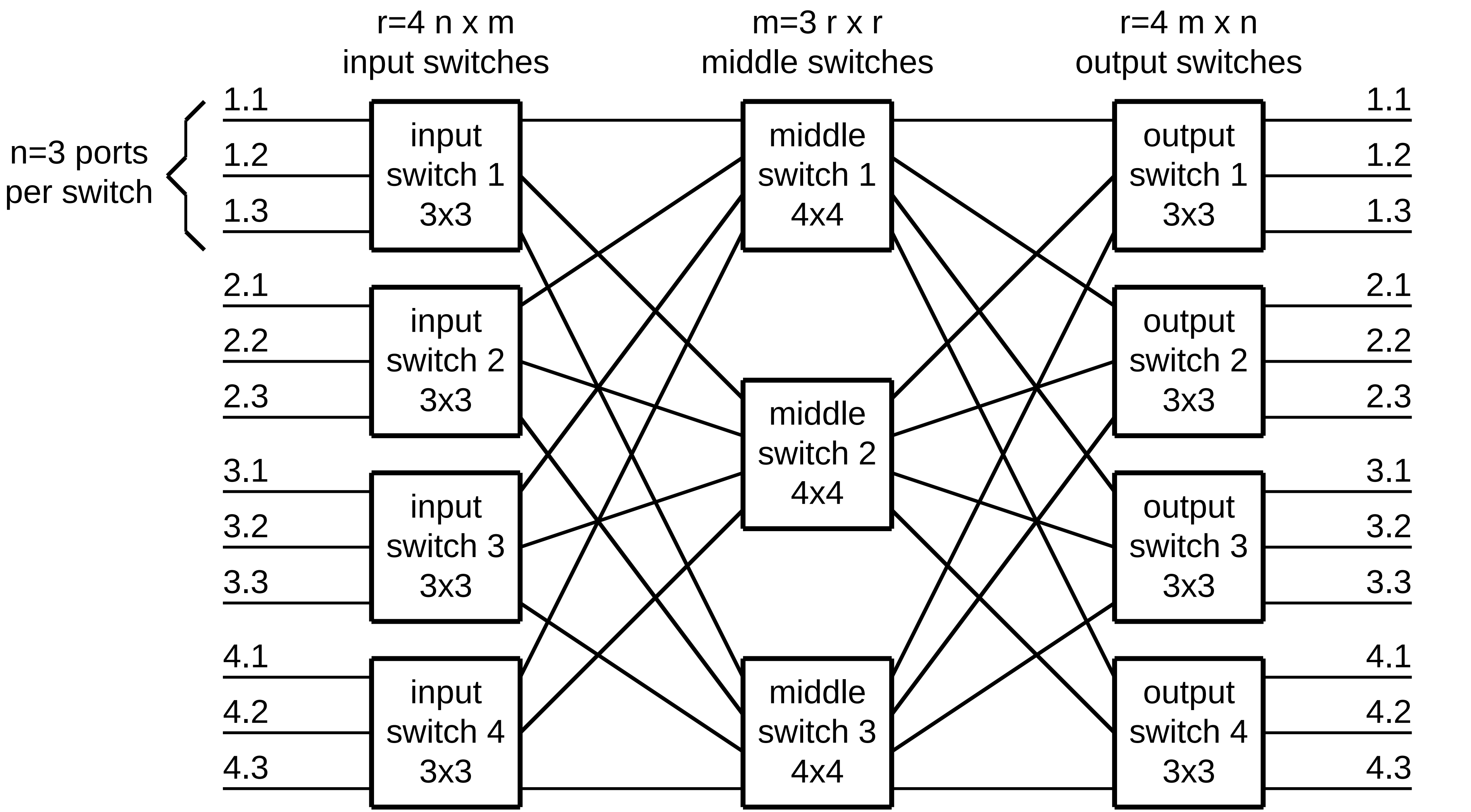


Crossbar Expansion

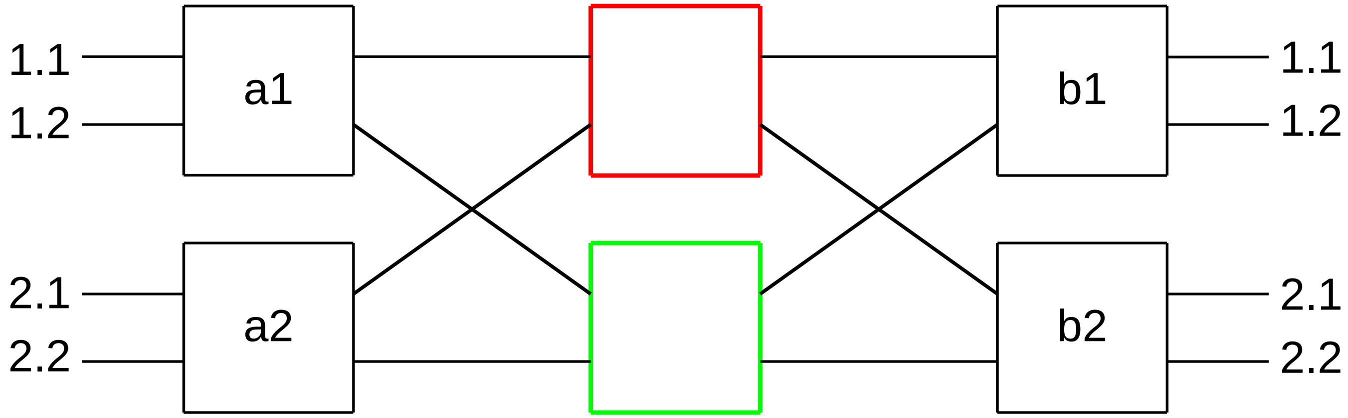


Clos Networks

Basic Clos Structure (3,3,4)

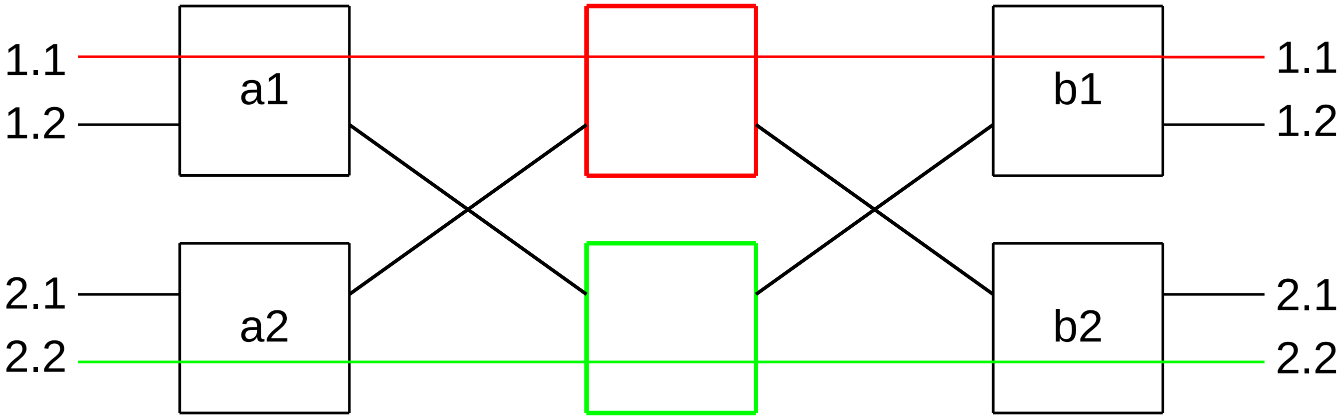


Simple Clos (2,2,2)



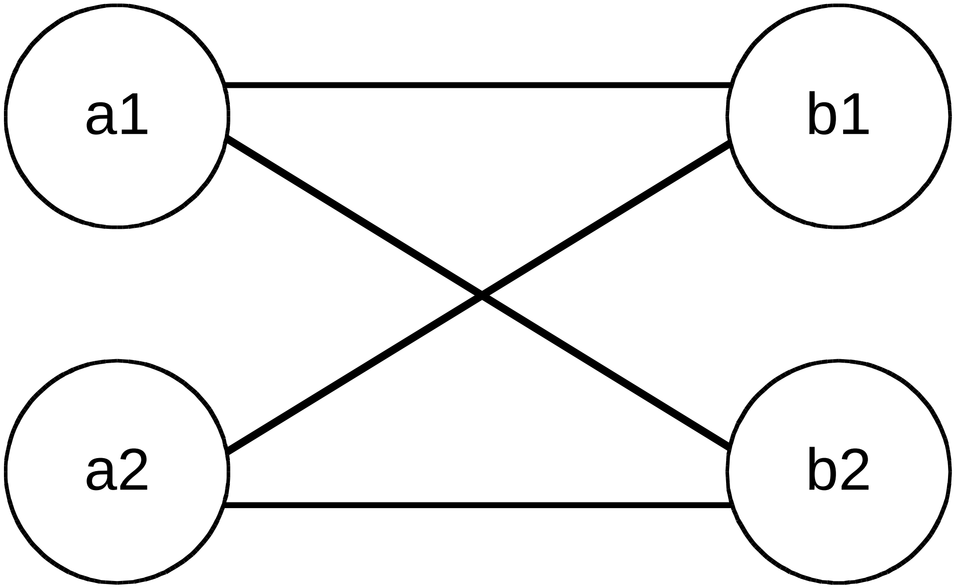
- Route
  - 1.1 to 1.1
  - 2.2 to 2.2
  - 1.2 to 2.1
  - 2.1 to 1.2

Routing example

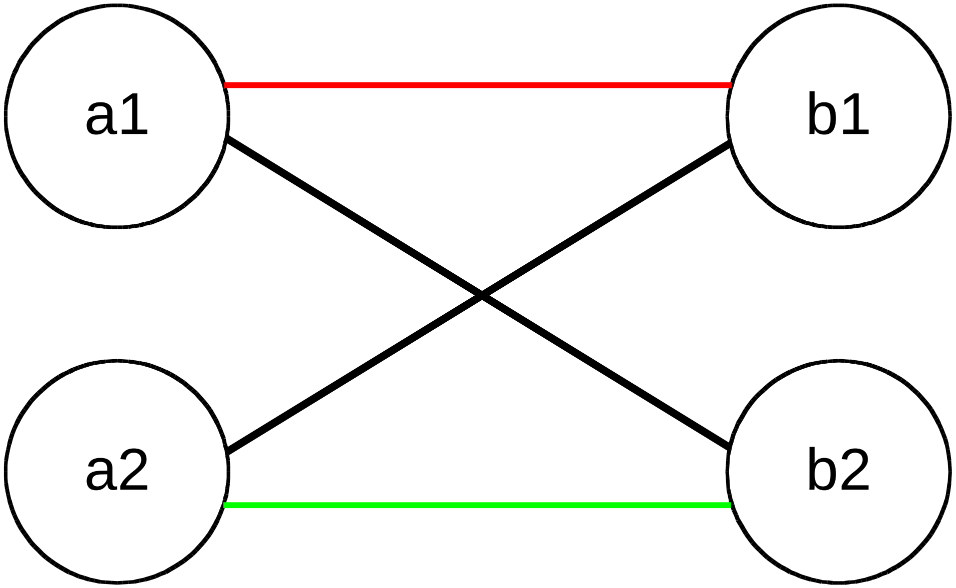


- Route
  - **1.1 to 1.1**
  - **2.2 to 2.2**
  - 1.2 to 2.1
  - 2.1 to 1.2

Edge Coloring

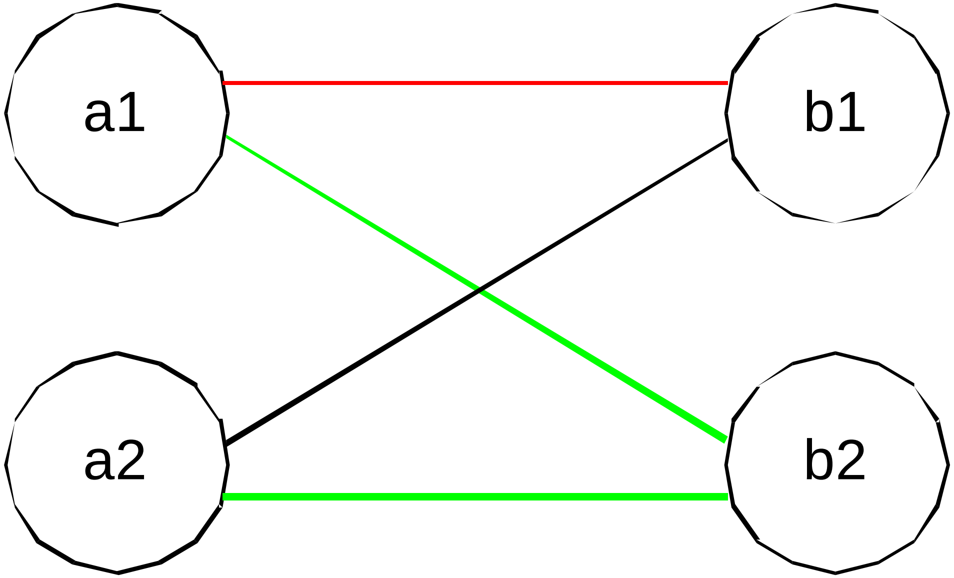


Edge Coloring



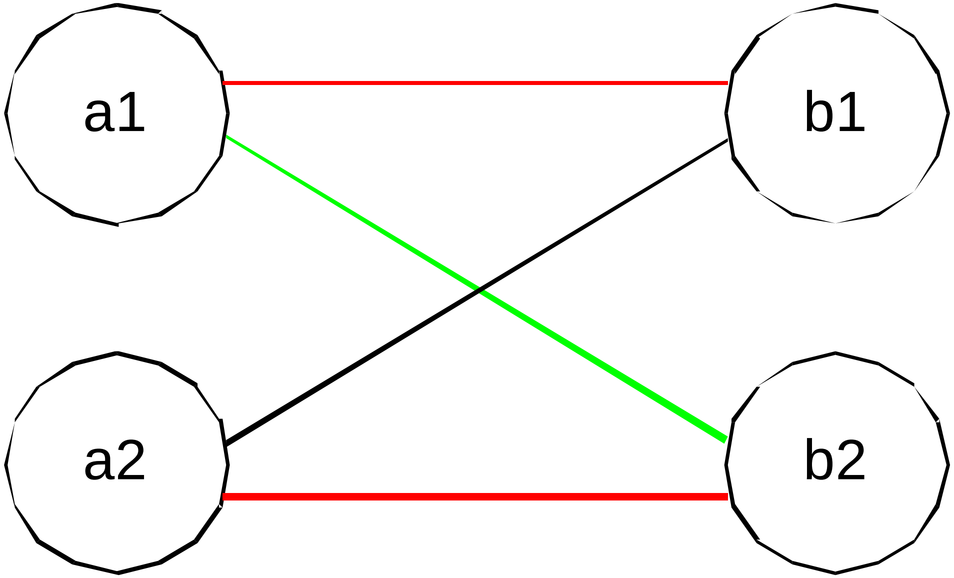
- Route
  - 1.1 to 1.1
  - 2.2 to 2.2
  - 1.2 to 2.1
  - 2.1 to 1.2

Rearrangement



- Route
  - 1.1 to 1.1
  - 2.2 to 2.2
  - 1.2 to 2.1
  - 2.1 to 1.2
- Route next call violating rule

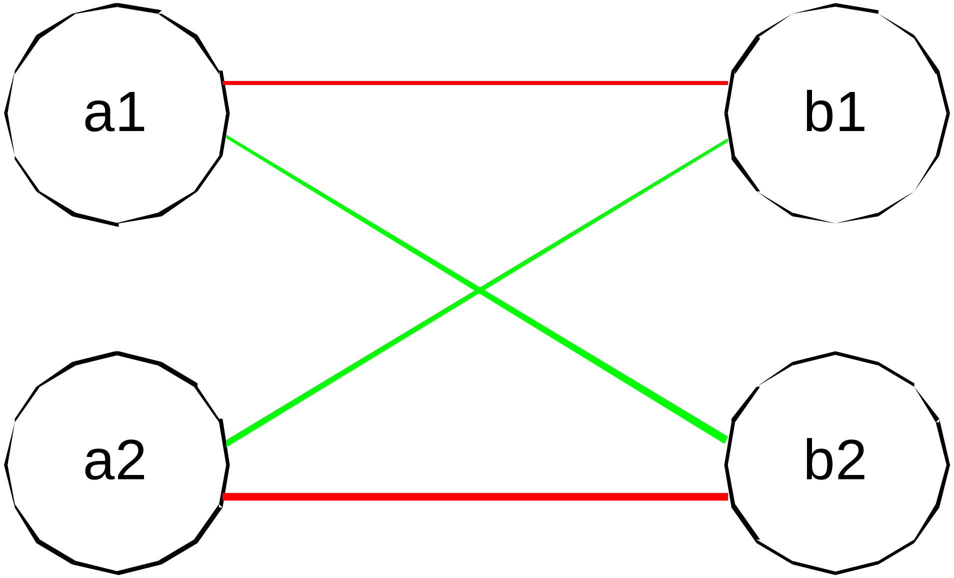
Rearrangement



- Route
  - 1.1 to 1.1
  - 2.2 to 2.2
  - 1.2 to 2.1
  - 2.1 to 1.2

- Route next call violating rule
- Fix color conflict at b2 by flipping other edge

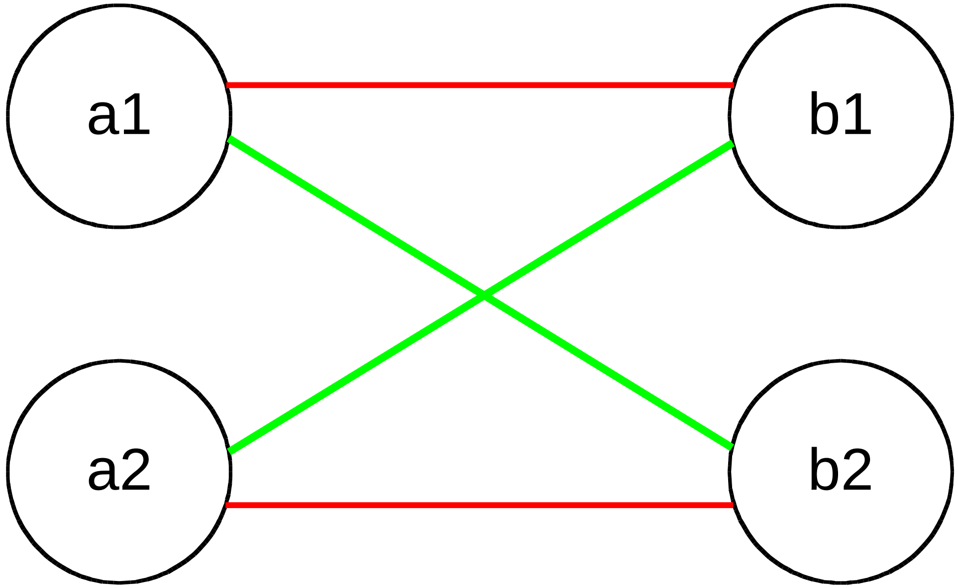
Rearrangement

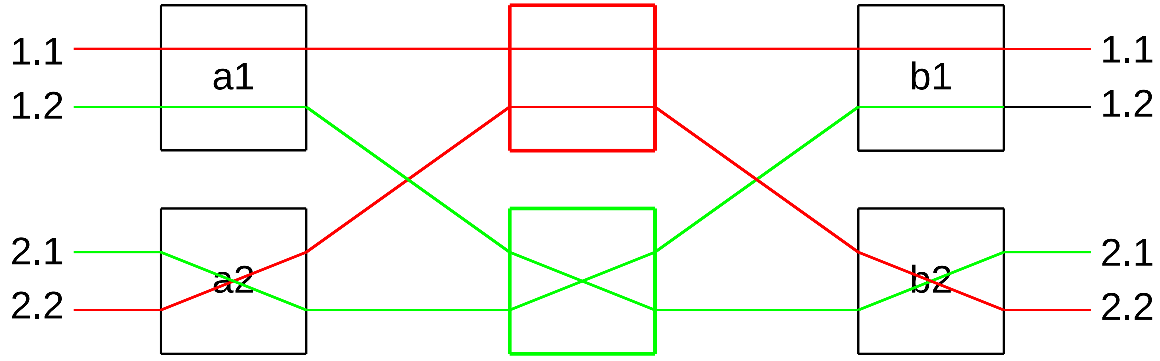


- Route
  - 1.1 to 1.1
  - 2.2 to 2.2
  - 1.2 to 2.1
  - 2.1 to 1.2

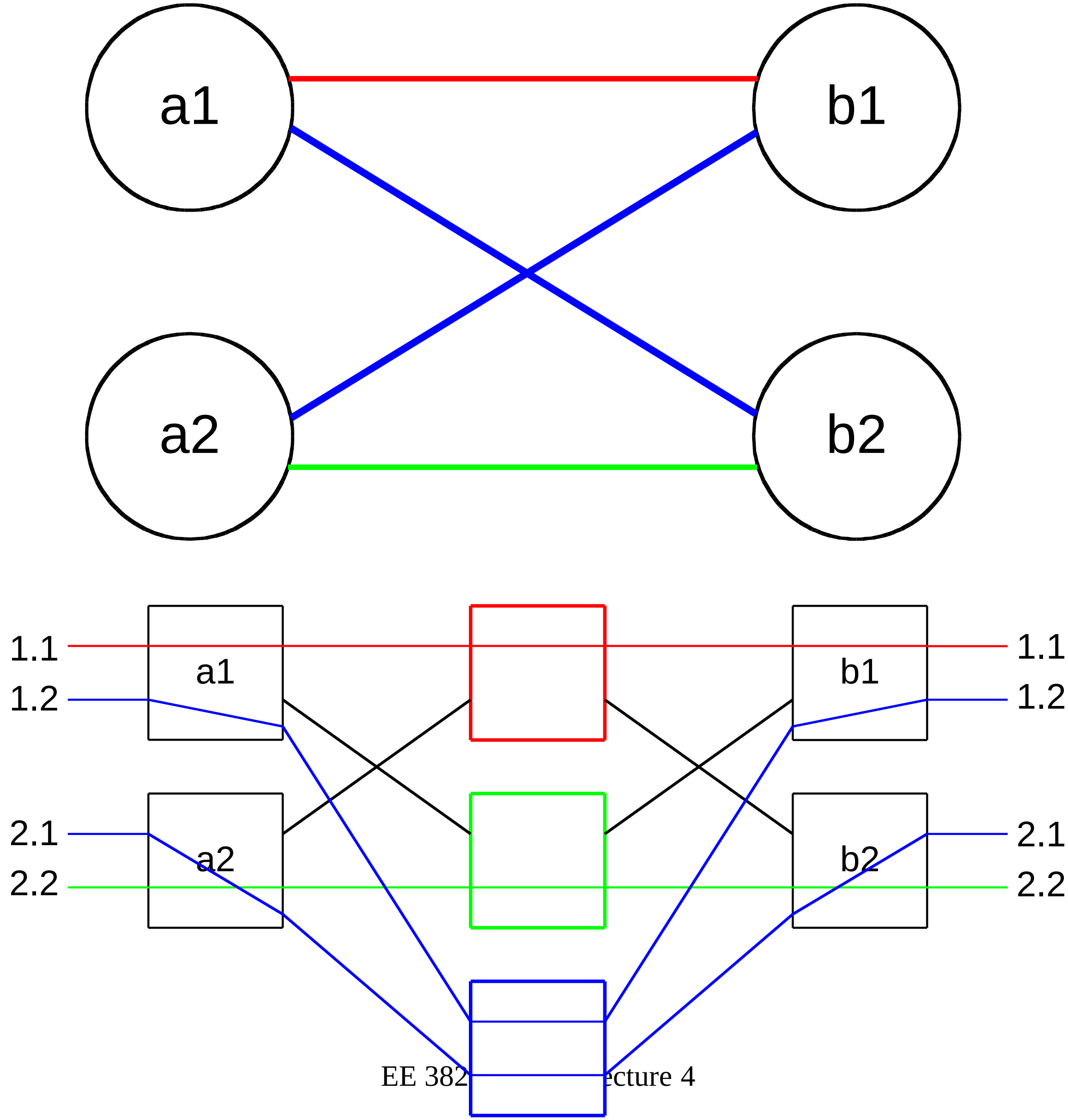
- Route next call violating rule
- Fix color conflict at b2 by flipping other edge
- Final circuit no longer conflicts

Rearrangement





Strictly non-blocking



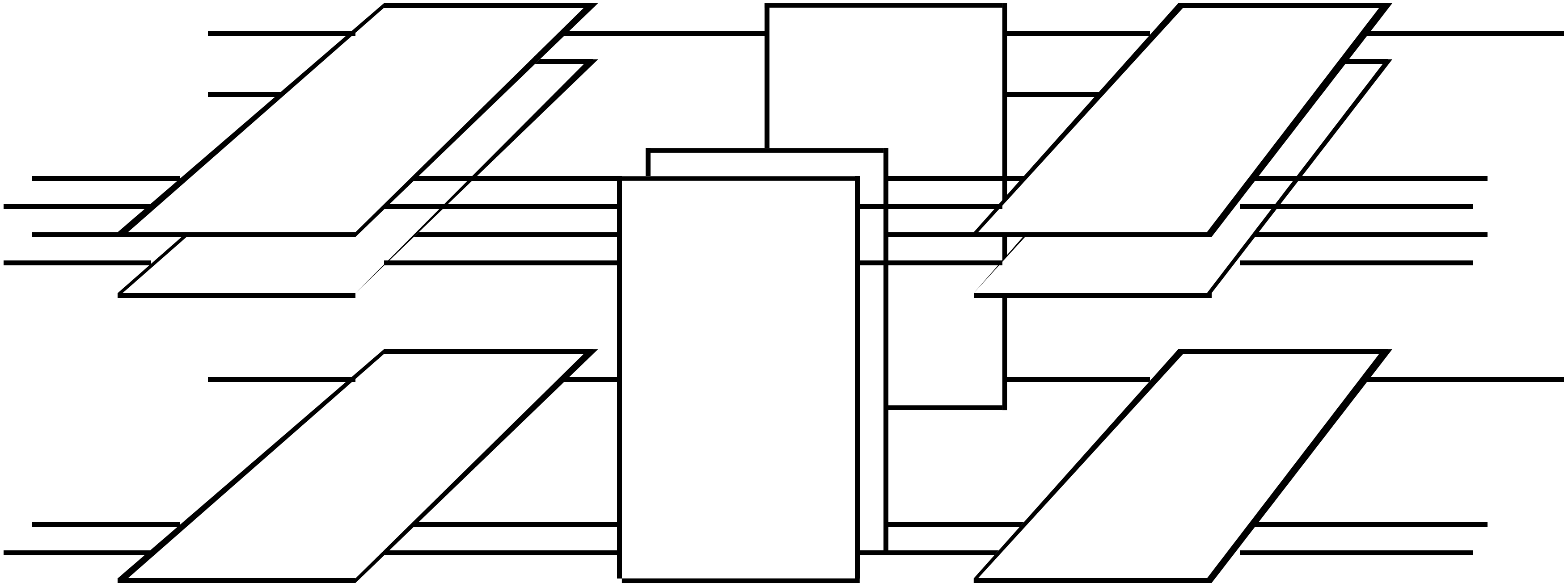
Strictly non-blocking Routing

- Number of stages

- Vector of free middle stages at every input and output switch
- E.g. with 15 middle stages:
  - Input i : 0 0 0 1 0 0 0 1 0 1 1 1 1 1 1

  - Output j : 0 1 1 0 1 1 1 0 1 0 0 0 0 1 1

3D View of a Clos Network



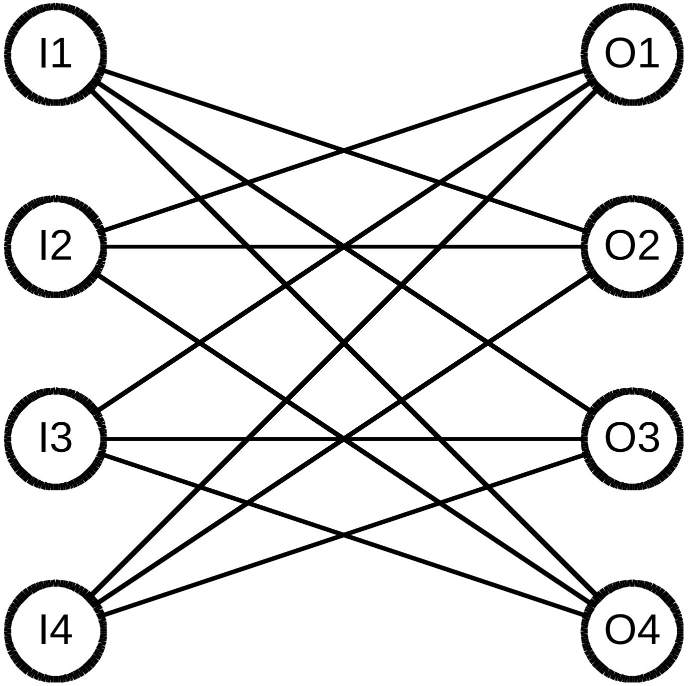
r - n x m
input switches

m - r x r
middle switches

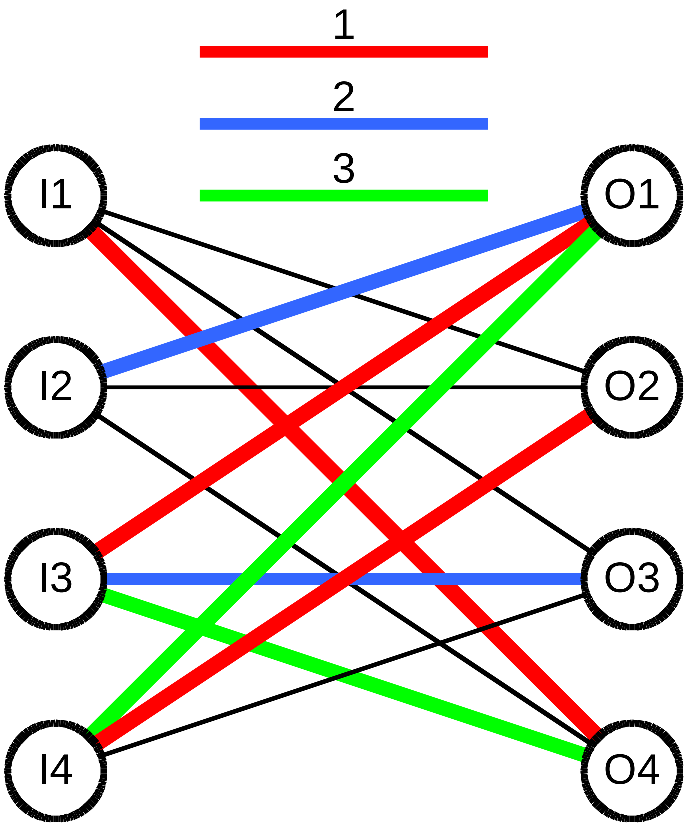
r - m x n
output switches

Rearrangement Example

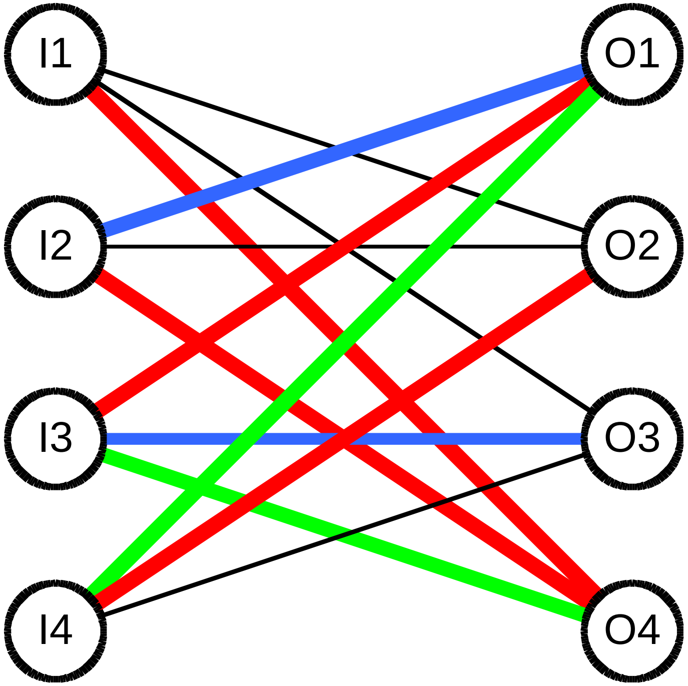
(3,3,4) Routing Problem



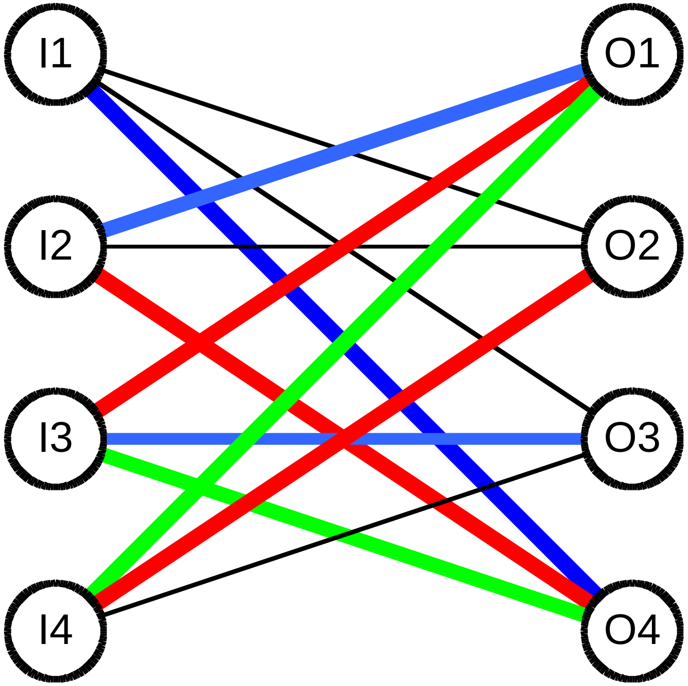
Before Setting Up (2,4)



Connect (2,4) using RED

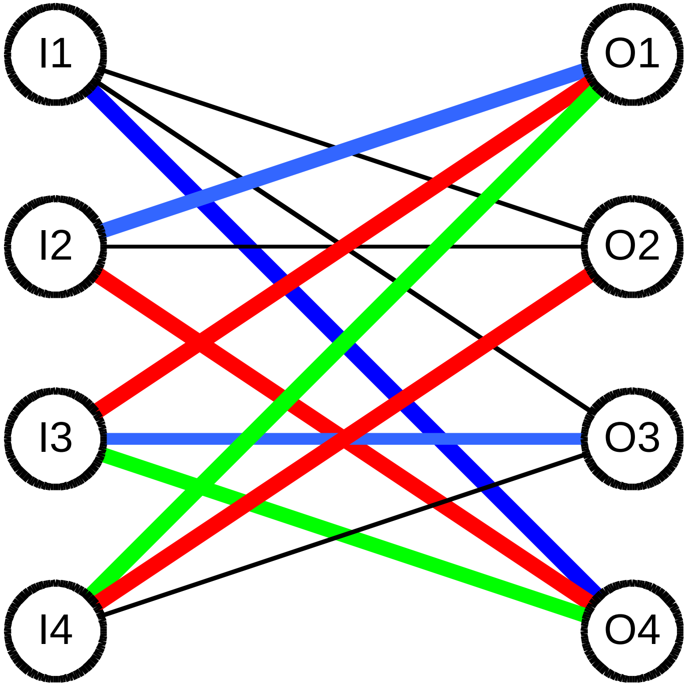


Switch (1,4) to BLUE

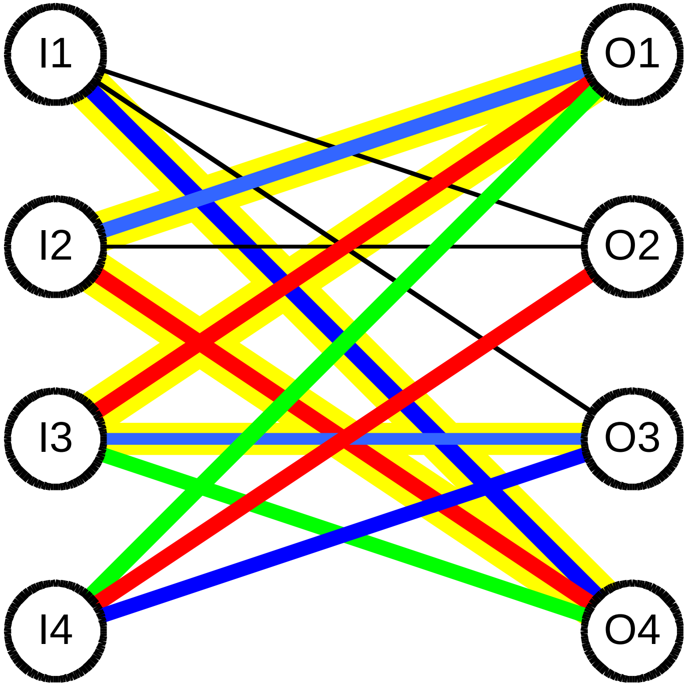


The red link being moved cannot lead back to i2 since i2's red link is already accounted for.

Connect (4,3) using BLUE

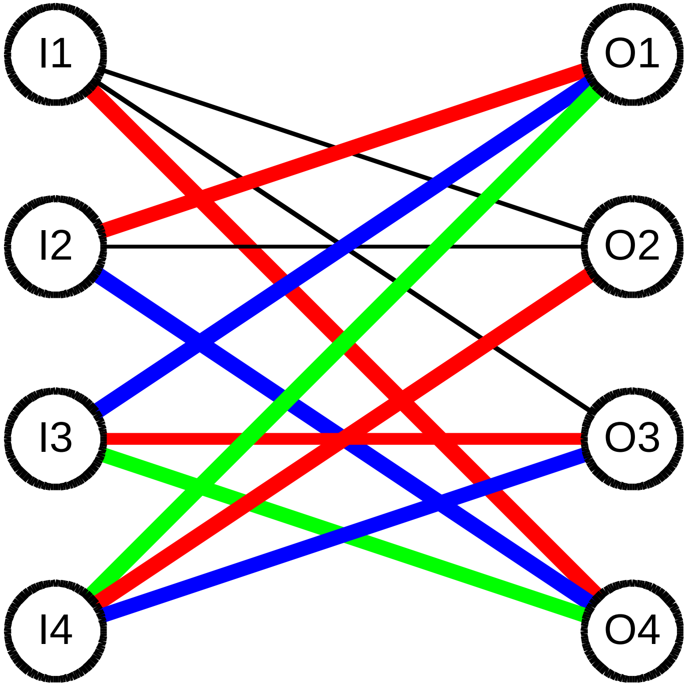


Chain of calls to be switched BLUE/RED

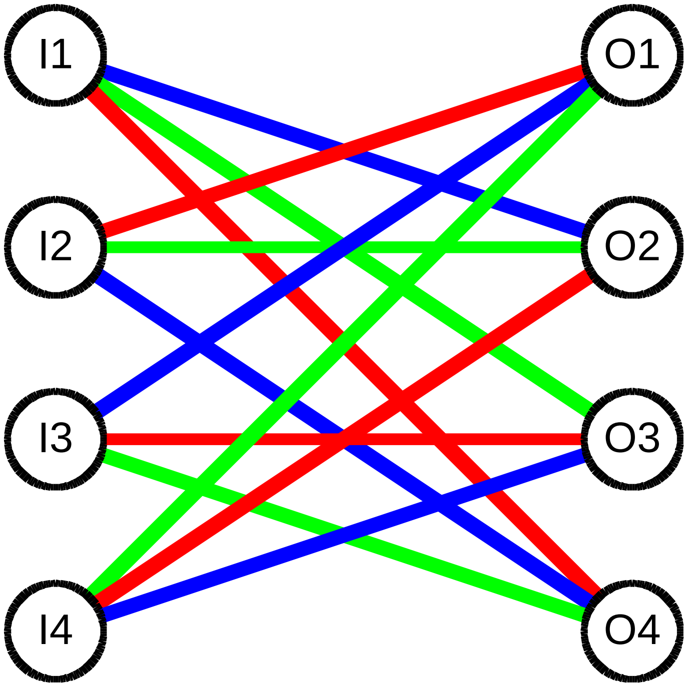


Chain is guaranteed to be acyclic since input blue and output red links of previous nodes are already accounted for.

After Switching 5 calls



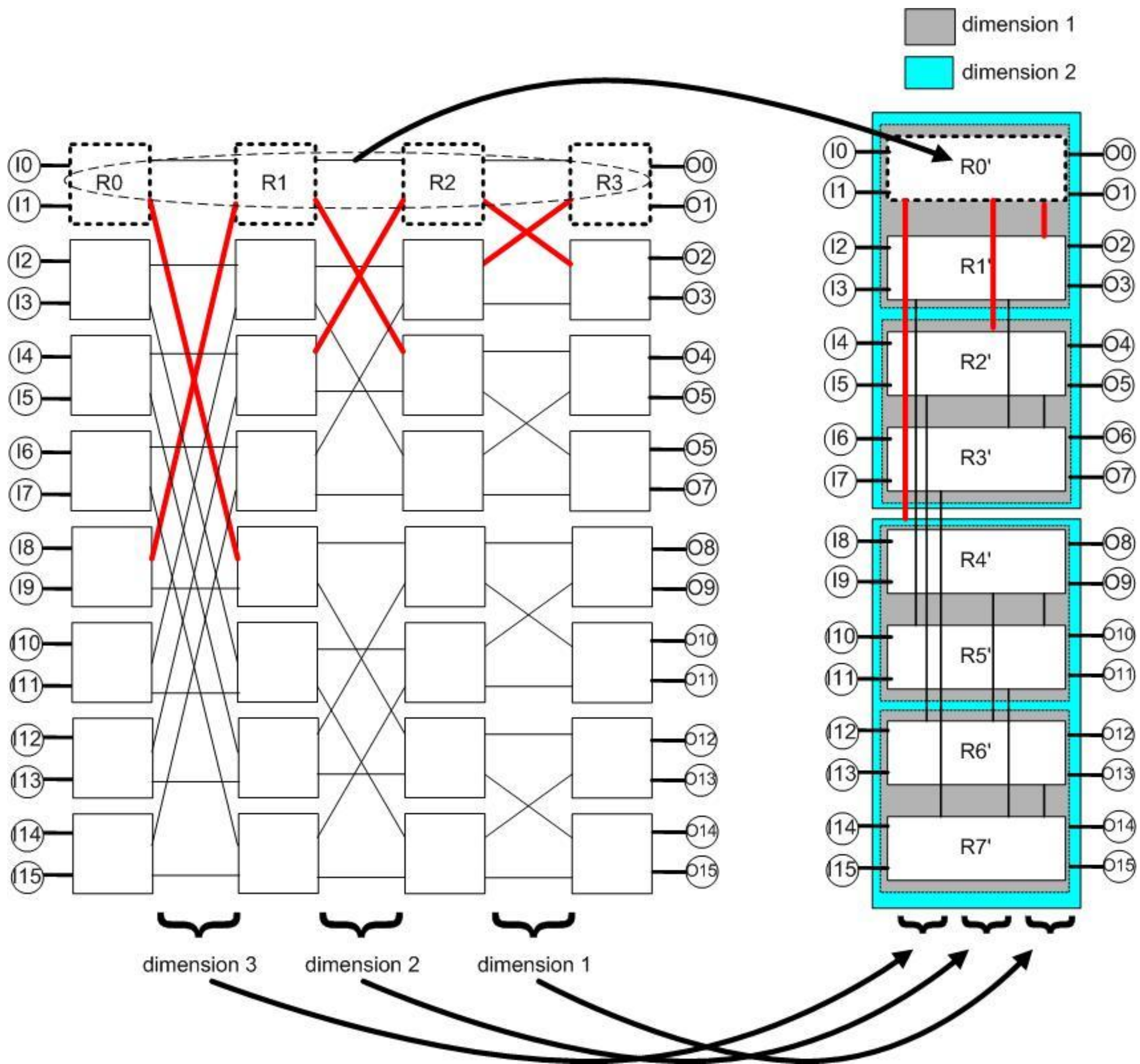
Complete Colored Graph



Question of the day

- What topology has an average hop count, H_{avg} for *load-balanced traffic* that is *close* to the $\log_{d/2}N$ bound and is able to route *arbitrary traffic* with an H_{max} of twice this amount?

Flattened Butterfly Topology



Non-Blocking Summary

- Non blocking
  - Can connect any unconnected input to any unconnected output
    - Strictly non-blocking: without moving any other connections
    - Rearrangeably non-blocking: may require moving other connections
  - Applies to *circuit switching*
    - For packet switching you usually want a *non-interfering* network
- Crossbar
  - Trivially non-blocking
  - Also has stiff backpressure
- Clos network
  - 3 stages of crossbars – each switch of stage i connects to all switches of stage i+1
  - (m,n,r)
  - Strictly non blocking if $m \geq 2n-1$, rearrangeable if $m \geq n$
  - Schedule with the *looping* algorithm – augmented paths
  - Multicast
    - Conflict vector representation
    - Can fanout in input or middle stage (split calls)