Politecnico
di Milano

Department of
Electronics and
Information

Switching Theory – 2016-17

Rearrangeable and Strictly Non-Blocking Networks

Outline

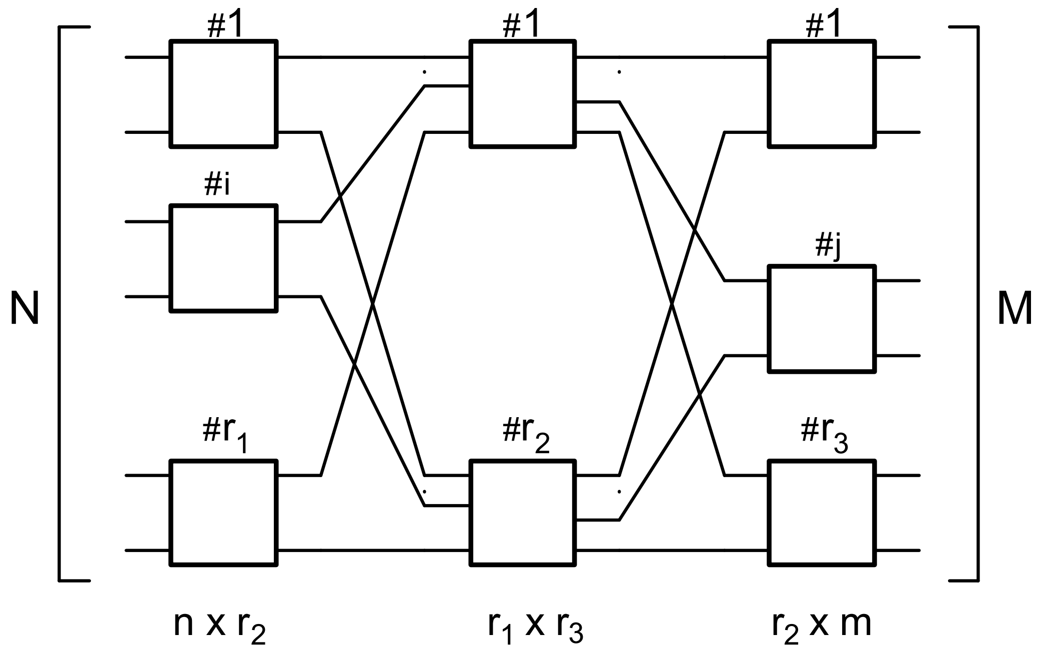
- Rearrangeable full-connection multistage network
- Strictly Non-Blocking full-connection multistage network

Outline

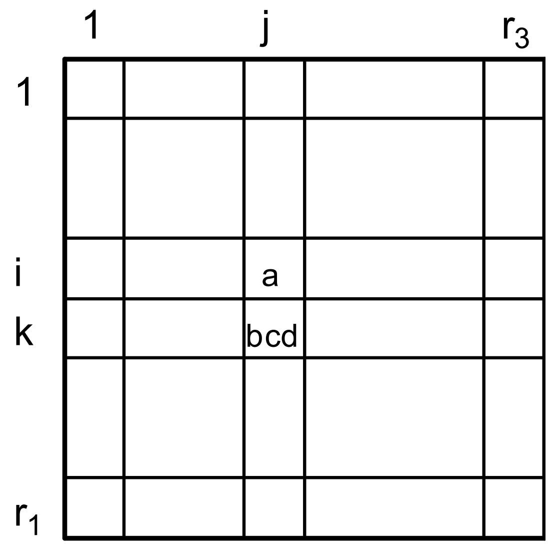
- **Rearrangeable full-connection multistage network**
  - Paull matrix
  - Rearrangeability conditions
- **Strictly Non-Blocking full-connection multistage network**

Full-connection networks

3 stages



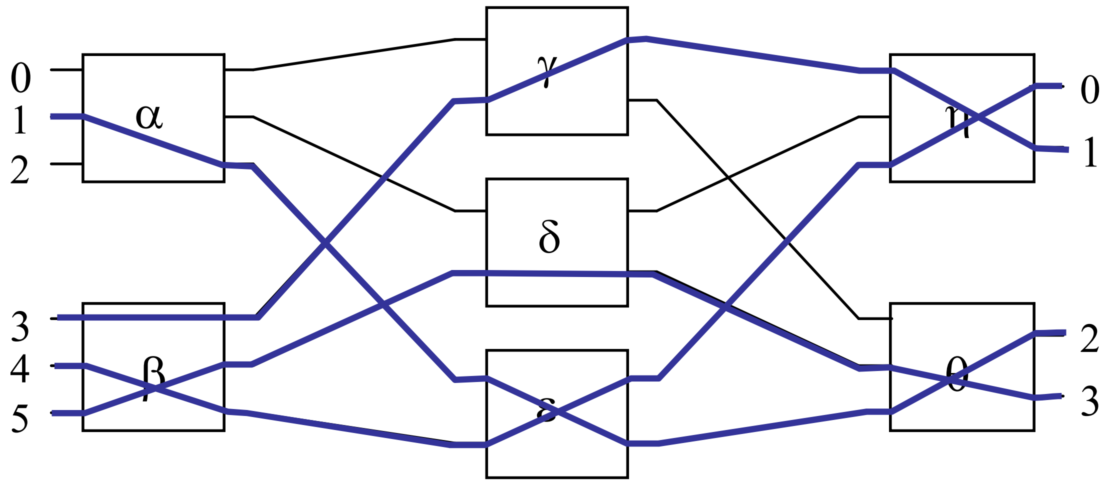
- **State of a three-stage network represented by the Paull matrix**
  - It has r_1 rows and r_3 columns
  - Each matrix entry has up to r_2 distinct symbols: 1, 2, …, r_2
    - The symbol *a* in the matrix entry means that an inlet of the first-stage matrix *i* is connected to an outlet of the last-stage matrix *j* through the middlestage matrix *a*



- **Conditions of the Paull matrix**
  - At most min(n,r_2) distinct symbols per row
  - At most min(r_2,m) distinct symbols per column

Paull matrix

Example



$n=3$
$r_1=2$

$r_2=3$

$m=2$
$r_3=2$

I	O
1	2
3	1
4	0
5	3

	η	θ
α		ε
β	γ ε	δ

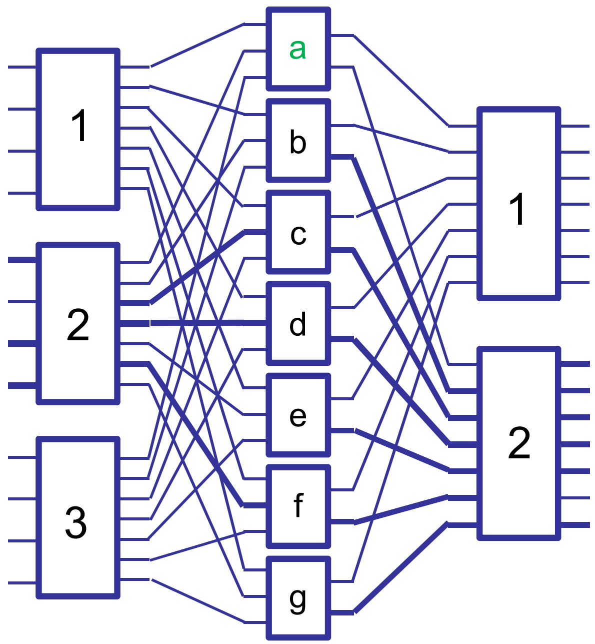
- **Slepian-Duguid theorem**
  - A 3-stage network is RNB if and only if $r_2 \geq \max(n,m)$
- **Necessity**
  - At most n-1 (m-1) connections can be supported by the first stage matrix i (third stage matrix j) with one idle inlet (outlet)
  - If $r_2 > \max(n\text{-}1,m\text{-}1)$ at least one of the r_2 symbols is missing in row i and column j

n = 4; m = 7; r_2 = 7

i\j	1	2
1		cdg
2	dc	f
3		be

abeg

a



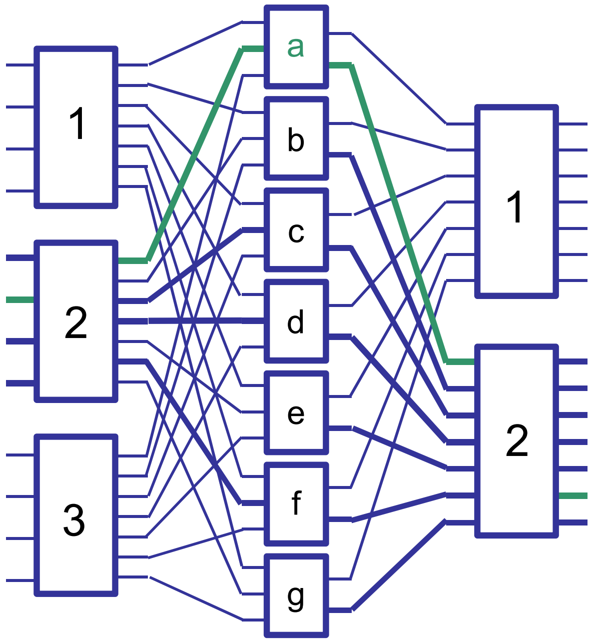
- **Slepian-Duguid theorem**
  - A 3-stage network is RNB if and only if $r_2 \geq \max(n,m)$
- **Sufficiency**
  - To set-up a new connection *i-j* either of the two conditions is true
  - 1 - There is a symbol *a* not found both in row *i* and column *j*
    - **Matrix *a* chosen for *i-j***

$n = 4;\ m = 7;\ r_2 = 7$

i\j	1	2
1		cdg
2	dc	af
3		be

$\cancel{a}$beg

$\cancel{a}$



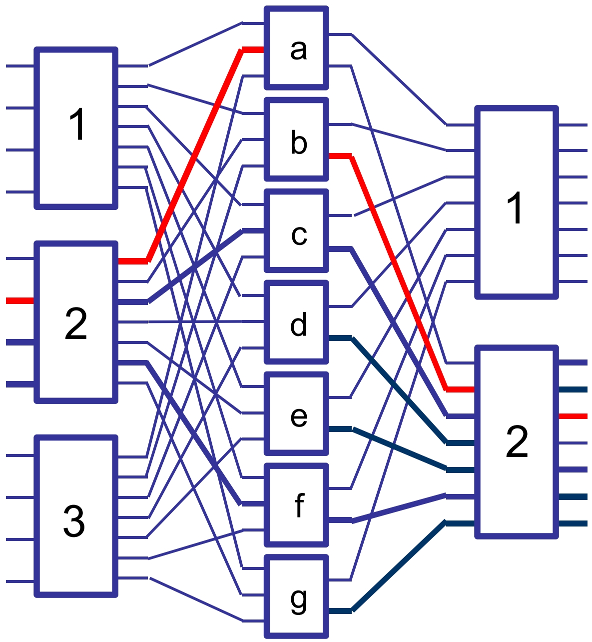
- **Slepian-Duguid theorem**
  - A 3-stage network is RNB if and only if $r_2 \geq \max(n,m)$
- **Sufficiency**
  - To set-up a new connection *i-j* either of the two conditions is true
  - 1 - There is a symbol *a* not found both in row *i* and column *j*
  - 2 - There is (at least) one symbol *a* in row *i* not found in column *j* and there is (at least) one symbol *b* in column *j* not found in row *i*
    - **Conflicting connections can be rearranged starting from *a* or *b* to set-up *i-j* through *a* or *b***

$n = 4;\ m = 7;\ r_2 = 7$

i\j	1	2
1		cdg
2	ac	f
3		be

bdeg

a



Full-connection networks

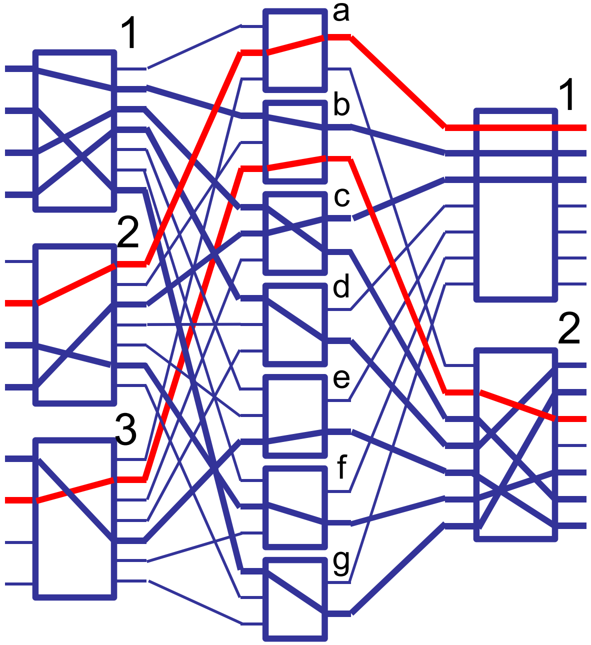
Rearrangeability conditions

$n = 4; m = 7; r_2 = 7$

i\j	1	2
1	b	cdg
2	ac	f
3		be

bdeg

a

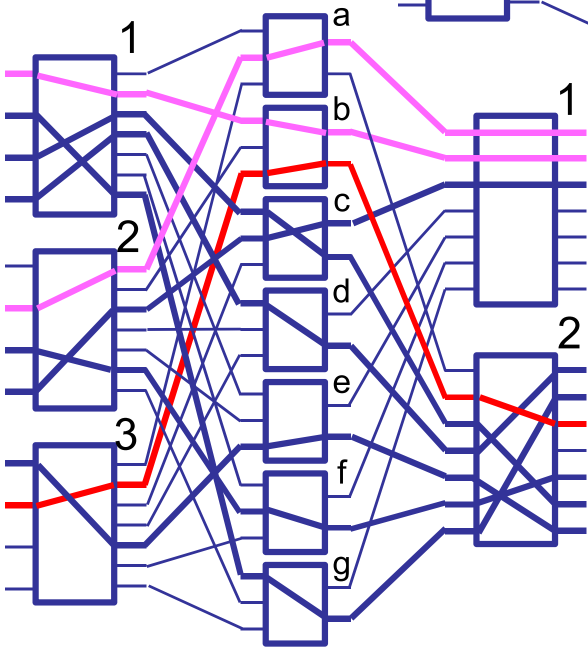


i\j	1	2
1	b	cdg
2	ac	f
3		be

aef

bdeg

a



Full-connection networks

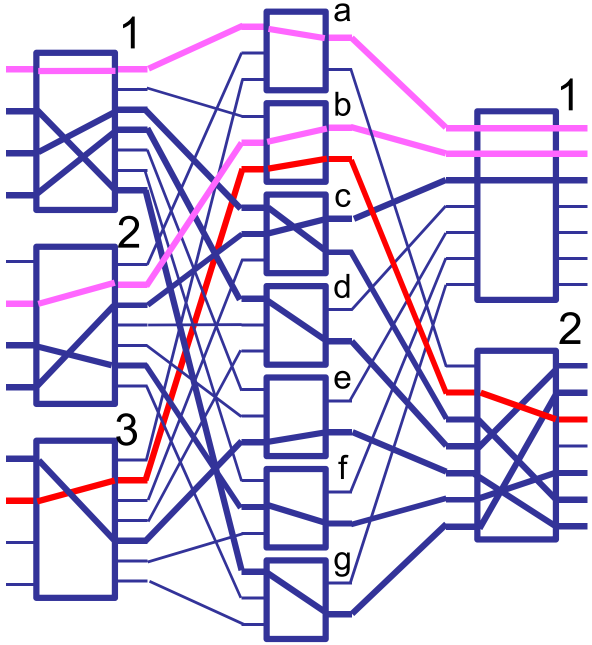
Rearrangeability conditions

$n = 4;\ m = 7;\ r_2 = 7$

i\j	1	2
1	a	cdg
2	bc	f
3		be

bef

adeg

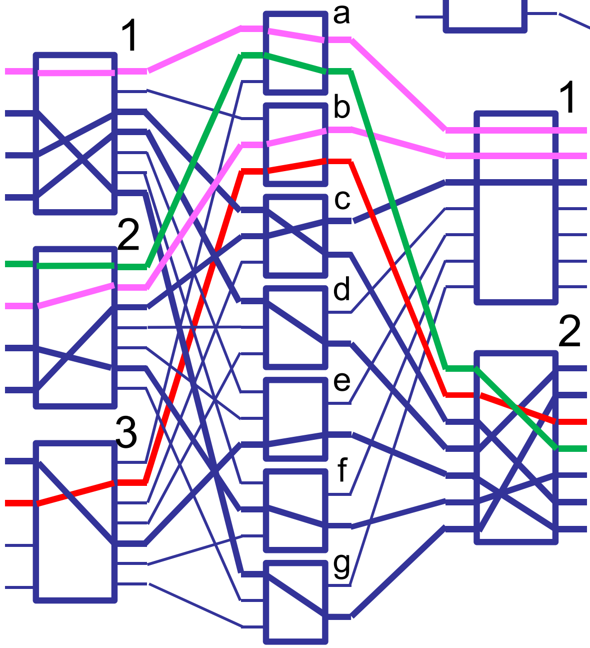


a

i\j	1	2
1	a	cdg
2	bc	af
3		be

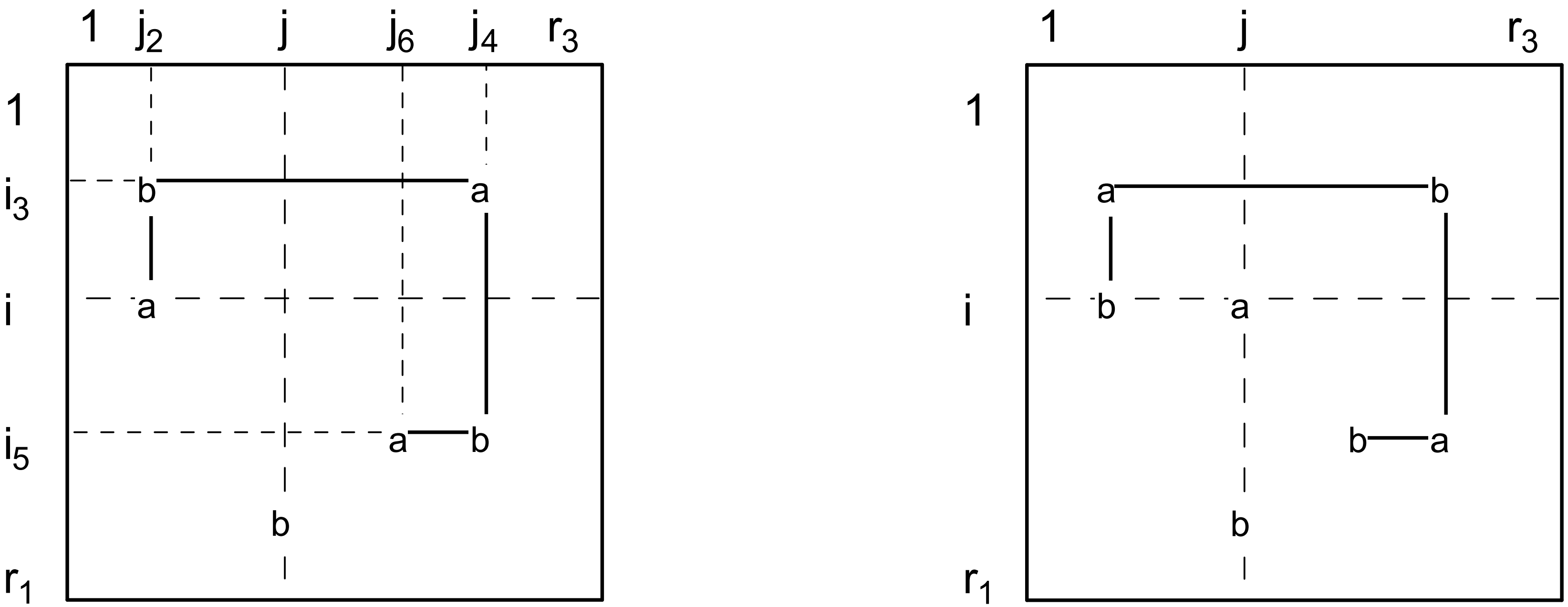
bef

deg



In our example:
We have rearranged
2 connections

- **Rearranging algorithm**

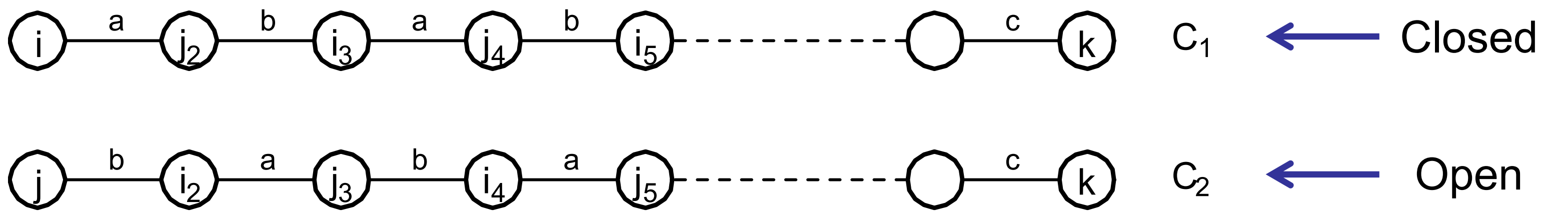


  - Assumption: start from <u>symbol *a*</u>
    - **Visit a <u>row</u> if it contains <u>symbol *b*</u>**
    - **Visit a <u>column</u> if it contains <u>symbol *a*</u>**
  - Continue as long as a symbol *a* or *b* is not found in the last column or row visited
  - Exchange *a* with *b* and vice versa throughout the set of selected symbols
  - After rearrangement starting from *a*, <u>symbol *a*</u> is written in entry (*i*,*j*)

Full-connection networks

Rearrangeability conditions

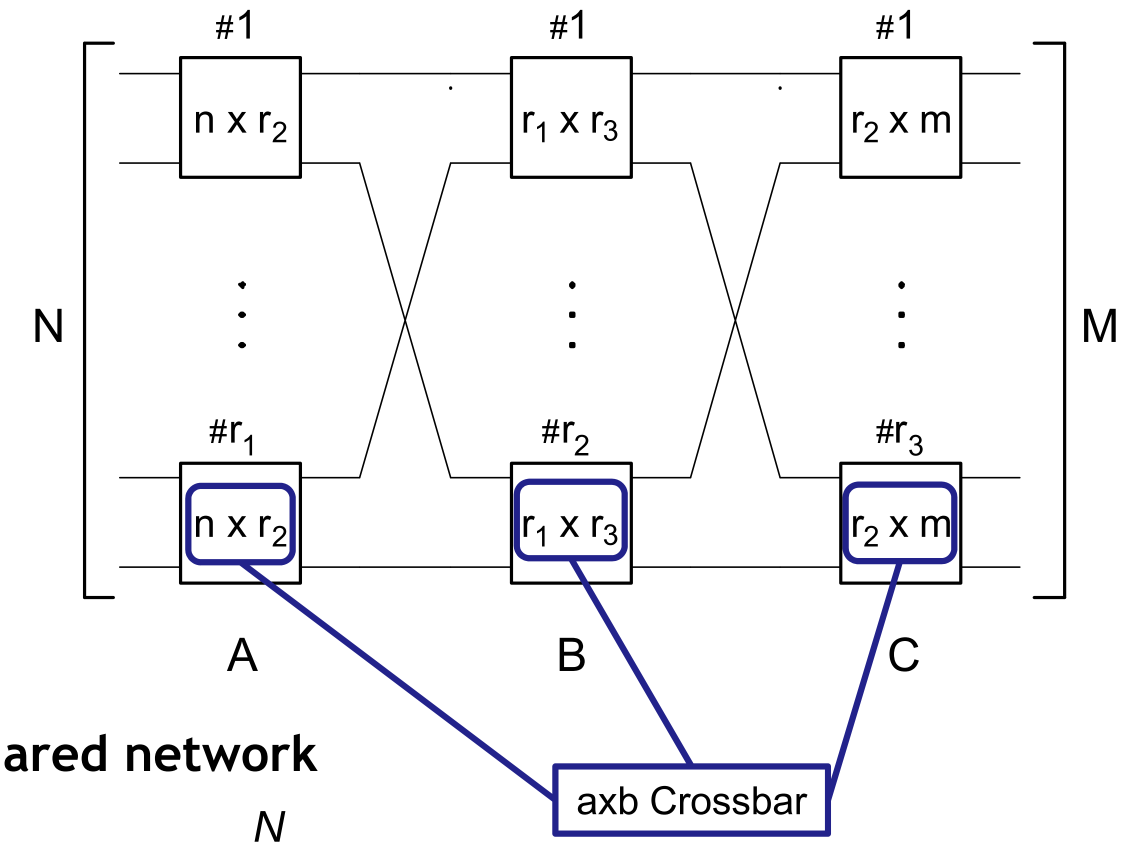
- **Sufficiency (cont.): consistency of rearrangement algorithm**
  - Chain: sequence of connections through stages 1 and 3 represented as the sequence of the crossed matrices
    - **Node: matrix at stage 1 and 3**
    - **Edge: second stage matrix**
  - Closed chain: first and last element belong to the same stage (both i-s or j-s)
    - **It always contains an even number of edges**
  - Open chain: first and last element belong to different stages (one is i, the other is j)
    - **It always contains an odd number of edges**
  - The rearrangement algorithm works if the end node (*k*) is always different from *i* and *j*
    - **Consider the last edge (c) of the chain leading to *k***



- ***c*=*a*: *k*≠*j* by hypothesis (*a* absent in column *j*), *k*≠*i* since *k*=*i* would generate a closed (open) chain with an odd (even) number of edges (see C_1 (C_2))**
- ***c*=*b*: *k*≠*i* by hypothesis (*b* absent in row *i*), *k*≠*j* since it would generate an open (closed) chain with an even (odd) number of edges (see C_1 (C_2))**

Slepian-Duguid network

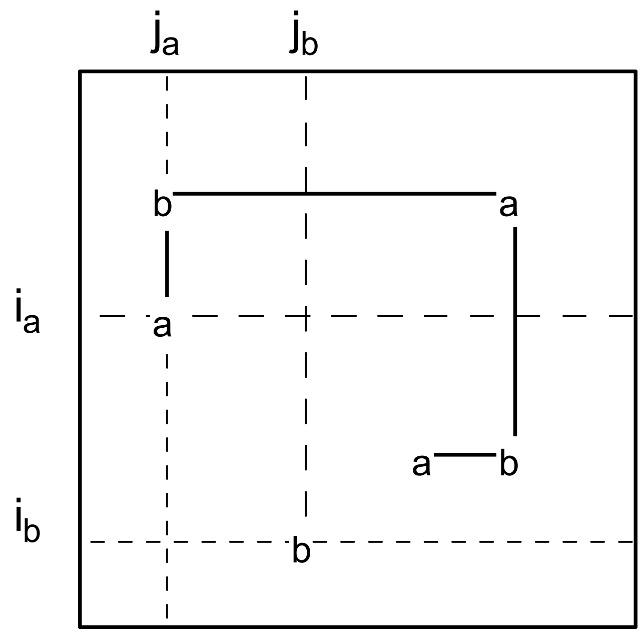
Network cost



- **Cost of a squared network**
  - $N = M,\ n = m \ \rightarrow\ r_1 = r_3 = \dfrac{N}{n}$
  - $r_2 = n$
  - $C = 2nr_2r_1 + r_1^2 r_2 = 2nN + \dfrac{N^2}{n}$
  - $\dfrac{dC}{dn} = 0 \ \rightarrow\ n = \sqrt{\dfrac{N}{2}} \quad \Rightarrow \quad C_{3,min} = 2\sqrt{2}N^{3/2}$

Slepian-Duguid network

Number of rearrangements

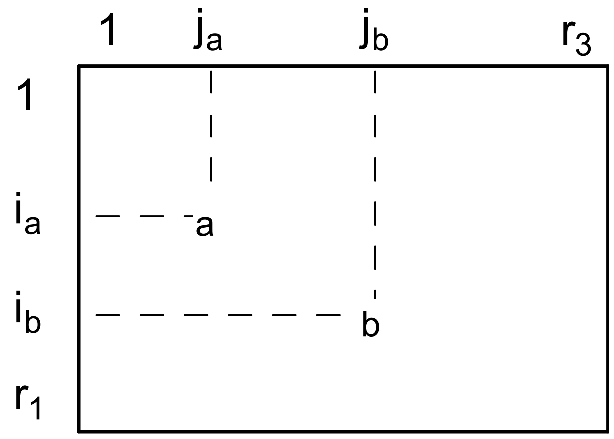


- **According to the proof of Slepian-Duguid theorem, at most r_1+r_3-2 connections to be rearranged**
  - The two symbols are located in entries (i_a, j_a), (i_b, j_b)
  - Rearrangement starts with *a* ($a \notin j_b$, $b \notin i_a$)
  - The chain does not have any symbol in column j_b, since it visits a new column only if it contains *a* (absent in j_b by definition)
  - The chain does not have any symbol in row i_b: it visits a new row only if it contains *b* (a second symbol *b* cannot appear in row i_b)
  - Hence the chain visits at most r_1-1 rows and r_3-1 columns
- **Actually the maximum # of symbols in the chain is 2min(r_1,r_3) - 2**
  - Rows and columns are visited alternatively
  - The minimum of r_1 and r_3 determines the total number of visits
- **At most 2*r*-2 if $r_1 = r_3 = r$**

In our example:
$r_1 = 3$; $r_3 = 2$
2min(r_1 r_3) – 2 = 2

Maximum number of rearrangements

- **Paull theorem**
  - At most ϕ_M = min (r_1,r_3)-1 connections need to be rearranged (upper bound)



- **Proof**
  - $r_1 \leq r_3$
    - **Two chains of symbols are built**
      - 1 - (a,b,a,b,...) starting from *a* in row i_a
      - 2 - (b,a,b,a,...) starting from *b* in column j_b
    - **The chains grow alternatively (their lengths differ by at most 1)**
    - **The chain that cannot grow further is selected for rearrangement**
    - **Max # growth steps : r_1-2**
      - rows i_a (*b* absent by definition) and i_b (*b* cannot appear twice) are not visited $\Rightarrow \phi_M = r_1$-1 (first symbol is changed too)
  - $r_1 \geq r_3$
    - **Analogously $\phi_M = r_3$-1 → ϕ_M = min (r_1,r_3) -1**
- **If $r_1 = r_3 = r$, then $\phi_M = r$-1**

In our example:
r_1 = 3; r_3 = 2
ϕ_M = min(r_1, r_3) – 1 = 1
→ There is a better solution

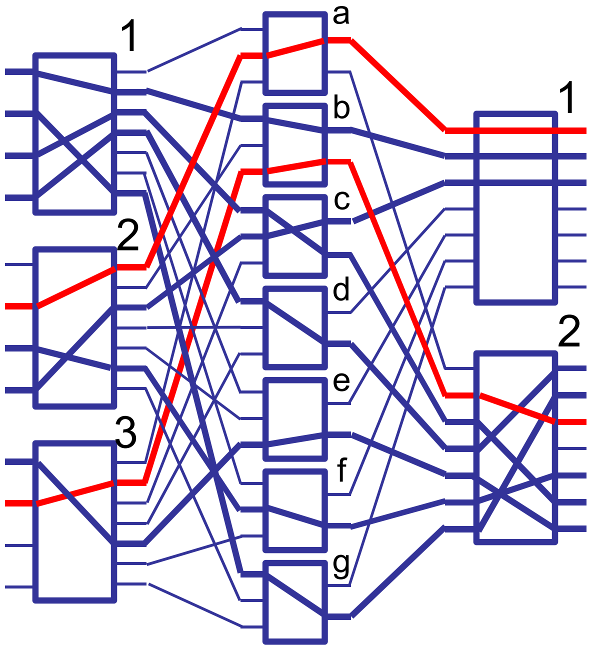
Number of rearrangements

$n = 4; m = 7; r_2 = 7$

i\j	1	2
1	b	cdg
2	ac	f
3		be

bdeg

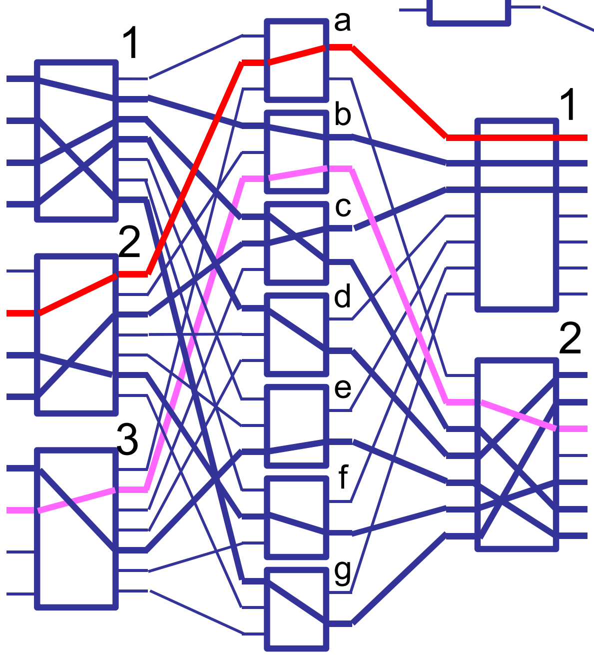
a



i\j	1	2
1	b	cdg
2	ac	f
3		be

bdeg

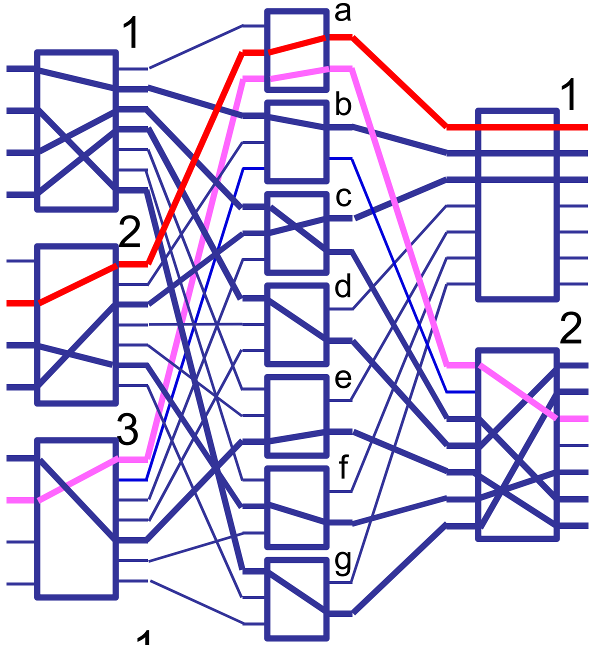
a



$n = 4;\ m = 7;\ r_2 = 7$

i\j	1	2
1	b	cdg
2	ac	f
3		ae

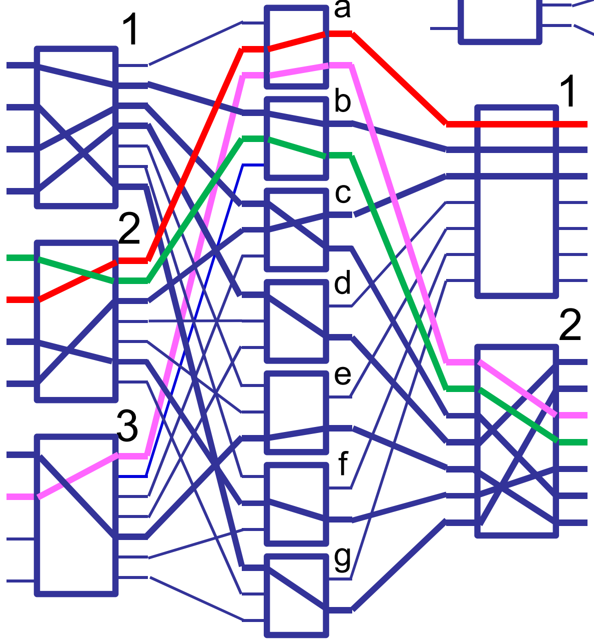
bdeg



b

i\j	1	2
1	b	cdg
2	ac	bf
3		ae

deg



Slepian-Duguid network

Number of rearrangements

- **Example:** N **= 24,** M **= 25,** r_1 **= 4,** r_3 **= 5,** n **= 6,** m **= 5**

Requested a new connection between matrices 1 and 1 of first and last stage

- Rearrangeable if $r_2 = 6$

	1	2	3	4	5
1	f		a	b e	c
2	a b	d		c	
3		c	e f		d
4	d		c	a	b f

Either **d** or **c** can be used for the new connection

- Starting from c, $\phi = 5$

	1	2	3	4	5
1	**c** f		a	b e	***d***
2	a b	***c***		***d***	
3		***d***	e f		***c***
4	d		c	a	b f

$\phi > \phi_M = \min(r_1, r_3) - 1 = 3$
→ There is a better solution

- Starting from d, $\phi = 2$

	1	2	3	4	5
1	**d** f		a	b e	c
2	a b	d		c	
3		c	e f		d
4	***c***		***d***	a	b f

$\phi < \phi_M = \min(r_1, r_3) - 1 = 3$

Slepian-Duguid network

Number of rearrangements

- **Example:** N **= 24,** M **= 25,** r_1 **= 4,** r_3 **= 5,** n **= 6,** m **= 5**

  - Rearrangeable if $r_2 = 6$

	1	2	3	4	5
1	f		a	b e	c
2	a b	d		c	
3		c	e f		d
4	d		c	a	b f

  - Starting from e, $\phi = 1$

	1	2	3	4	5
1	**e** f		a	b ***d***	c
2	a b	d		c	
3		c	e f		d
4	d		c	a	b f

  - Starting from d, $\phi = 1$

	1	2	3	4	5
1	**d** f		a	b e	c
2	a b	d		c	
3		c	e f		d
4	***e***		c	a	b f

Requested a new connection between matrices 1 and 1 of first and last stage

We can also chose the pair **d** or **e** for the new onnection

By selecting the proper connection pair, we can achieve better results (less rearrangements)

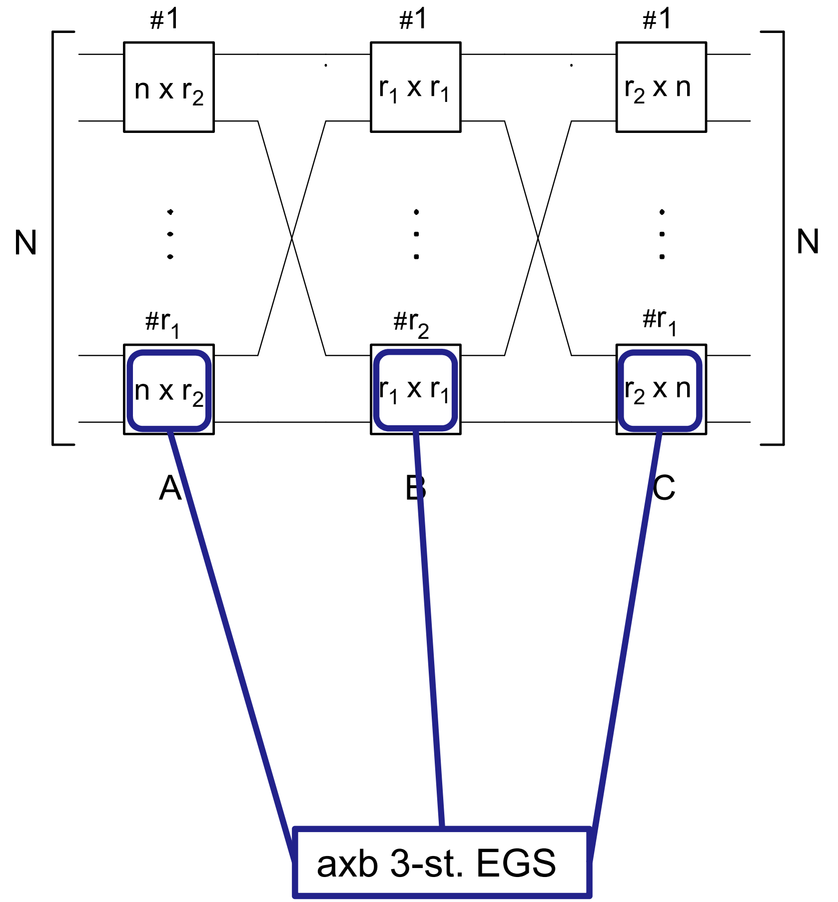
Slepian-Duguid squared network

Cost index with EGS networks

- **Cost index with EGS construction of all matrices**

$$\#SE_{M\times KM} = \frac{5}{4}kM^2 - (K+1)M$$

- $\log_2 N$ even
  - $\log_2 N\, even \rightarrow n = \sqrt{N}$
  - $r_1 = \frac{N}{\sqrt{N}} = \sqrt{N};\;\; r_2 = n = \sqrt{N}$
  - $\#SE_{tot} = \frac{15}{4}N\sqrt{N} - 6N$
  - $\#XP_{tot} = 3N\sqrt{N}$
- $\log_2 N$ odd
  - $\log_2 N\, odd \rightarrow n = \sqrt{\frac{N}{2}}$
  - $r_1 = \frac{N}{n} = \sqrt{2N};\;\; r_2 = n = \sqrt{\frac{N}{2}}$
  - $\#SE_{tot} = \frac{10\sqrt{2}}{4}N\sqrt{N} - 6N$
  - $\#XP_{tot} = 2\sqrt{2}N\sqrt{N}$



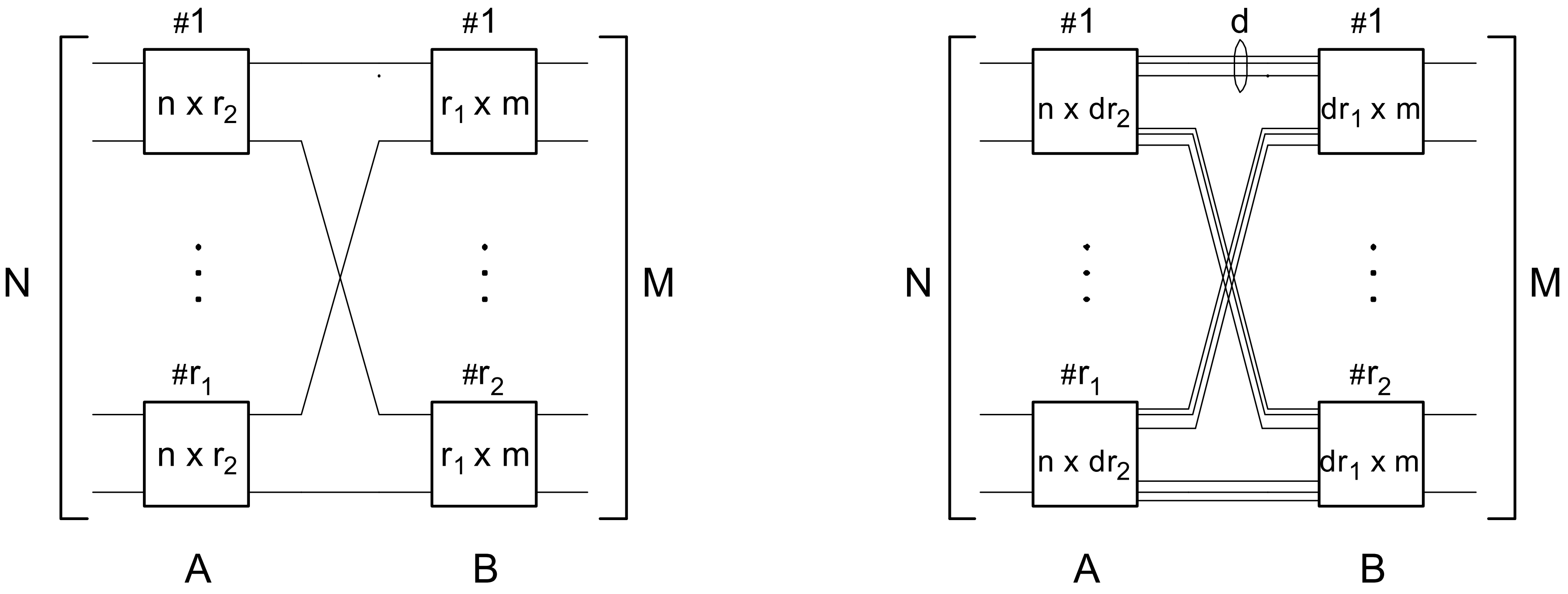
Outline

- **Rearrangeable full-connection multistage network**
- **Strictly Non-Blocking full-connection multistage network**
  - Two stages
  - Three stages
  - Clos networks

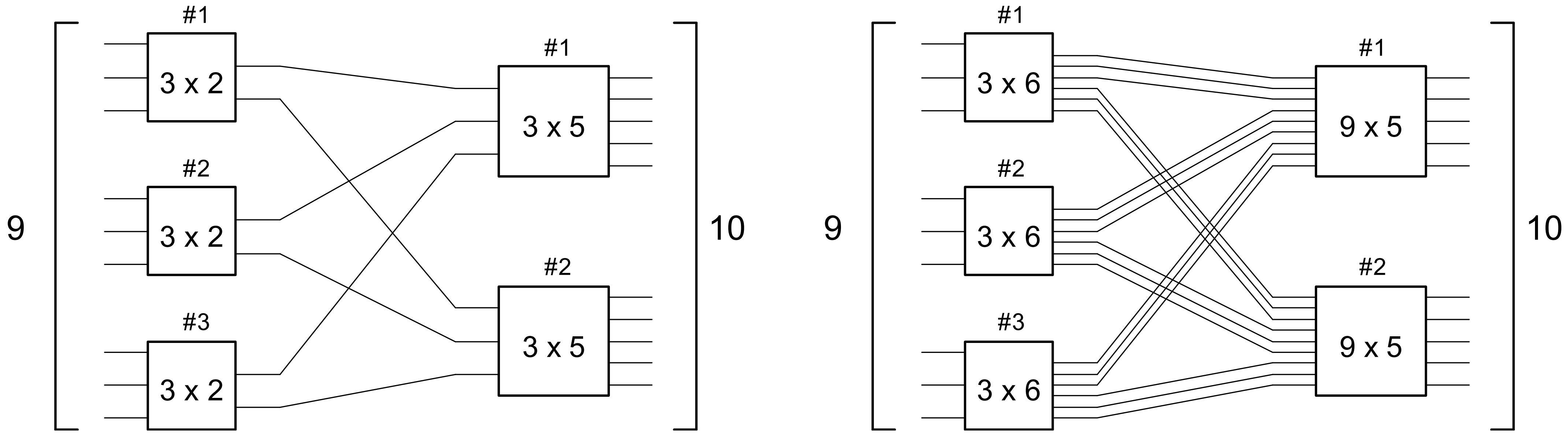
FC networks

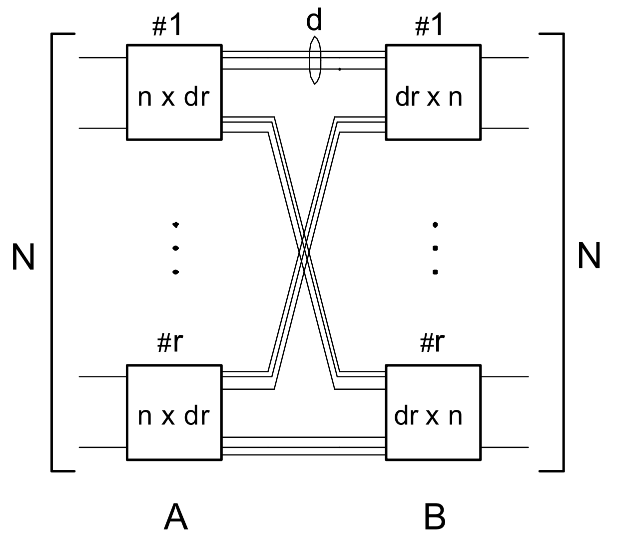
Two stages (link dilation)

- **Fully accessible, but blocking**
- **SNB if $d \geq$ min [n,m]**
- **Cost of dilated network = d times cost of non dilated**
- **$C = ndr_2r_1 + dr_1mr_2 = 2n^2r^2 = 2N^2$**
  - $N = M$, $n = m$, $r_1 = r_2 = r$, $N = r \cdot n$
  - The two-stage non-blocking network doubles the crossbar network cost



Two stages – Example



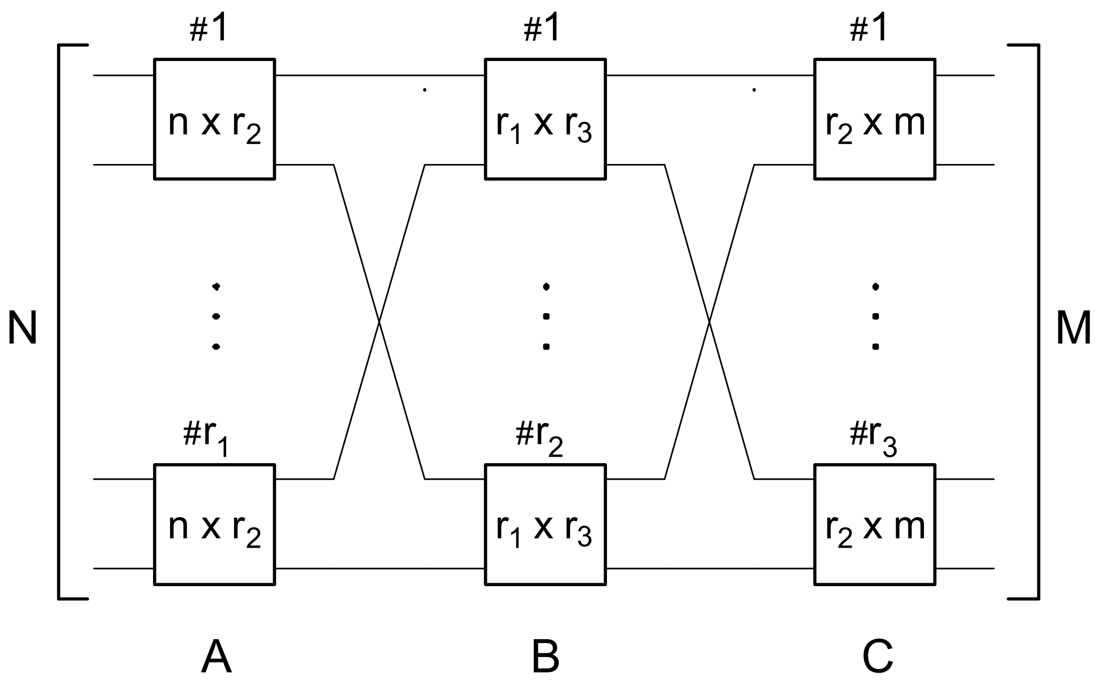


- **log$_2$N <u>even</u>**
  - $n = \sqrt{N}$
  - $SE_1 : \sqrt{N} \times N; \;\; SE_2 : N \times \sqrt{N}$
  - $\# SE_{tot} = \frac{5}{2} N^2 - 2N\sqrt{N} - 2N$
  - $\# XP_{tot} = 2N^2$
- **log$_2$N <u>odd</u>**
  - $n = \sqrt{\frac{N}{2}}$
  - $SE_1 : \sqrt{\frac{N}{2}} \times N; \;\; SE_2 : N \times \sqrt{\frac{N}{2}}$
  - $\# SE_{tot} = \frac{5}{2} N^2 - 2\sqrt{2} N\sqrt{N} - 2N$
  - $\# XP_{tot} = 2N^2$

FC networks

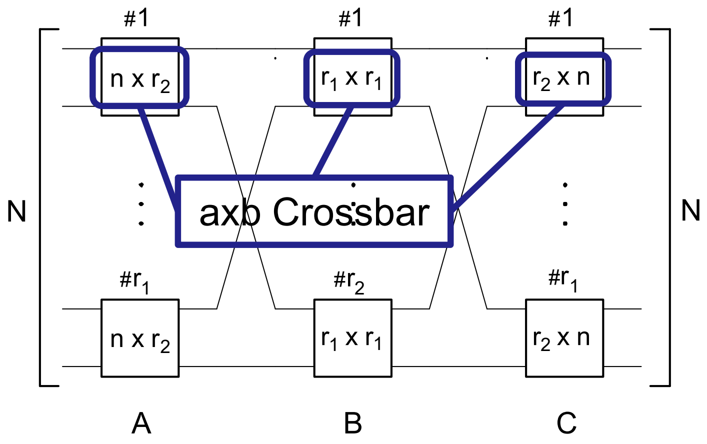
Three stages

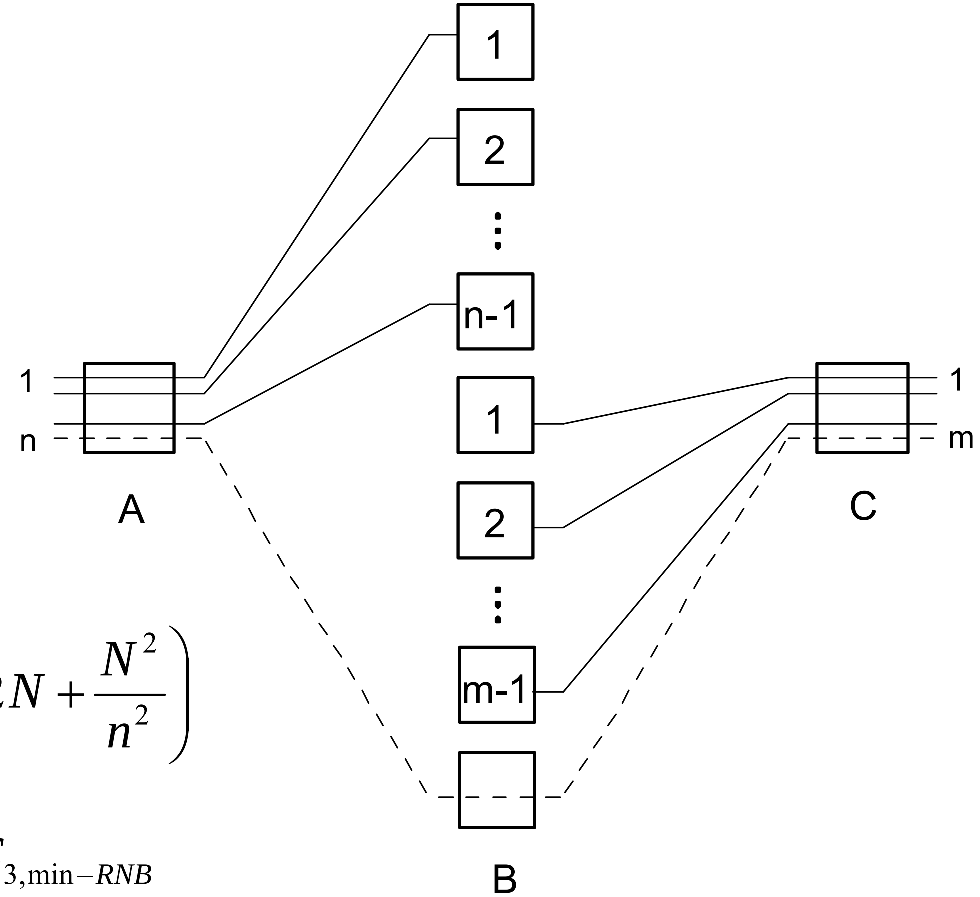
- **Fully accessible**
- **SNB?**



Optimum Clos network

- **3-stage network N x M with matrices** $n \times r_2$, $r_1 \times r_3$, $r_2 \times m$ ($N = nr_1$, $M = mr_3$)
- **Clos theorem. A 3-stage network is strict-sense non-blocking (SNB) if and only if** $r_2 \geq n+m-1$ ($r_2 = 2n-1$ **if** $n = m$)
- **Network cost** ($M = N$)

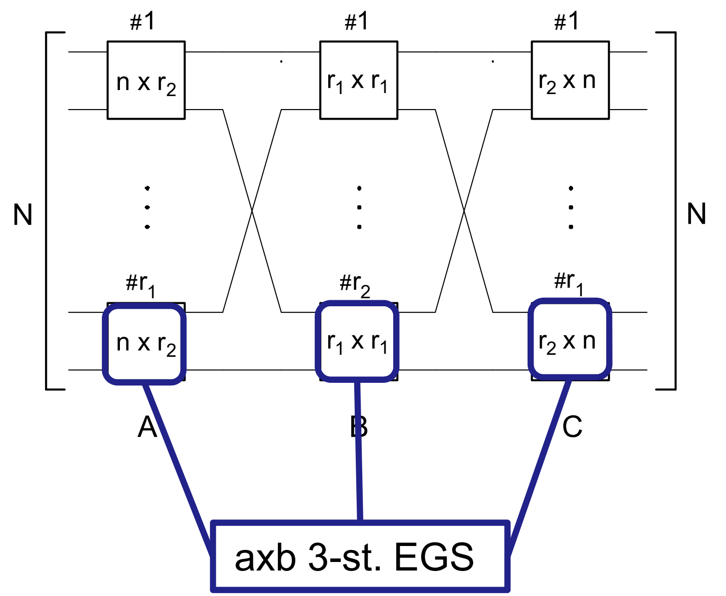




- $C_3 = 2nr_1r_2 + r_1^2 r_2 = 2nr_1(2n-1) + r_1^2(2n-1) = (2n-1)\left(2N + \frac{N^2}{n^2}\right)$
- $\frac{dC_3}{dn} = 0 \;\rightarrow\; n \cong \sqrt{\frac{N}{2}} \;\Rightarrow\; C_{3,\min} = 4\sqrt{2}N^{3/2} - 4N \cong 2C_{3,\min-RNB}$
- $C_3 < C_1 \;\rightarrow\; 4\sqrt{2}N^{3/2} - 4N < N^2 \;\Rightarrow\; \sqrt{N} > 2 + 2\sqrt{2} \;\Rightarrow\; N > 23.314 \;\Rightarrow\; N > 24$

Clos squared network

Cost index with EGS networks

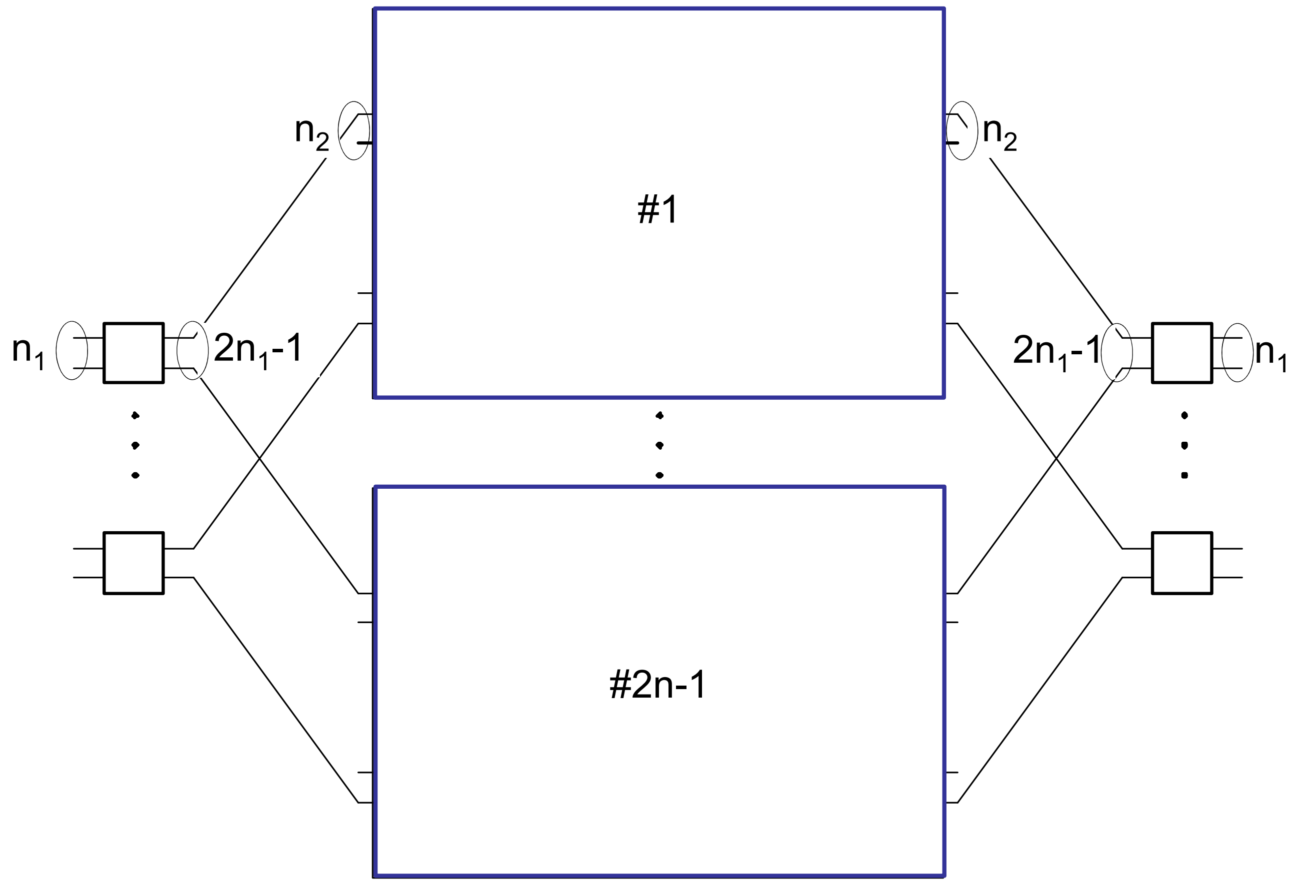


- **$\log_2 N$ even**
  - $n = \sqrt{N}$
  - $r_1 = \dfrac{N}{\sqrt{N}} = \sqrt{N};\quad r_2 = 2n - 1 = 2\sqrt{N} - 1 \cong 2\sqrt{N}$
  - $SE_1 : \sqrt{N} \times 2\sqrt{N};\quad SE_2 : \sqrt{N} \times \sqrt{N};\quad SE_3 : 2\sqrt{N} \times \sqrt{N}$
  - $\#SE_{tot} = \dfrac{15}{2} N\sqrt{N} - 10N$
  - $\#XP_{tot} = 6N\sqrt{N}$
- **$\log_2 N$ odd**
  - $n = \sqrt{\dfrac{N}{2}}$
  - $r_1 = \dfrac{N}{n} = \sqrt{2N};\quad r_2 = 2n - 1 = 2\sqrt{\dfrac{N}{2}} - 1 \cong \sqrt{2N}$
  - $SE_1 : \sqrt{\dfrac{N}{2}} \times \sqrt{2N};\quad SE_2 : \sqrt{2N} \times \sqrt{2N};\quad SE_3 : \sqrt{2N} \times \sqrt{\dfrac{N}{2}}$
  - $\#SE_{tot} = 5\sqrt{2} N\sqrt{N} - 10N$
  - $\#XP_{tot} = 4\sqrt{2} N\sqrt{N}$

Non-blocking FC networks

Extension of the Clos network – 5 stages

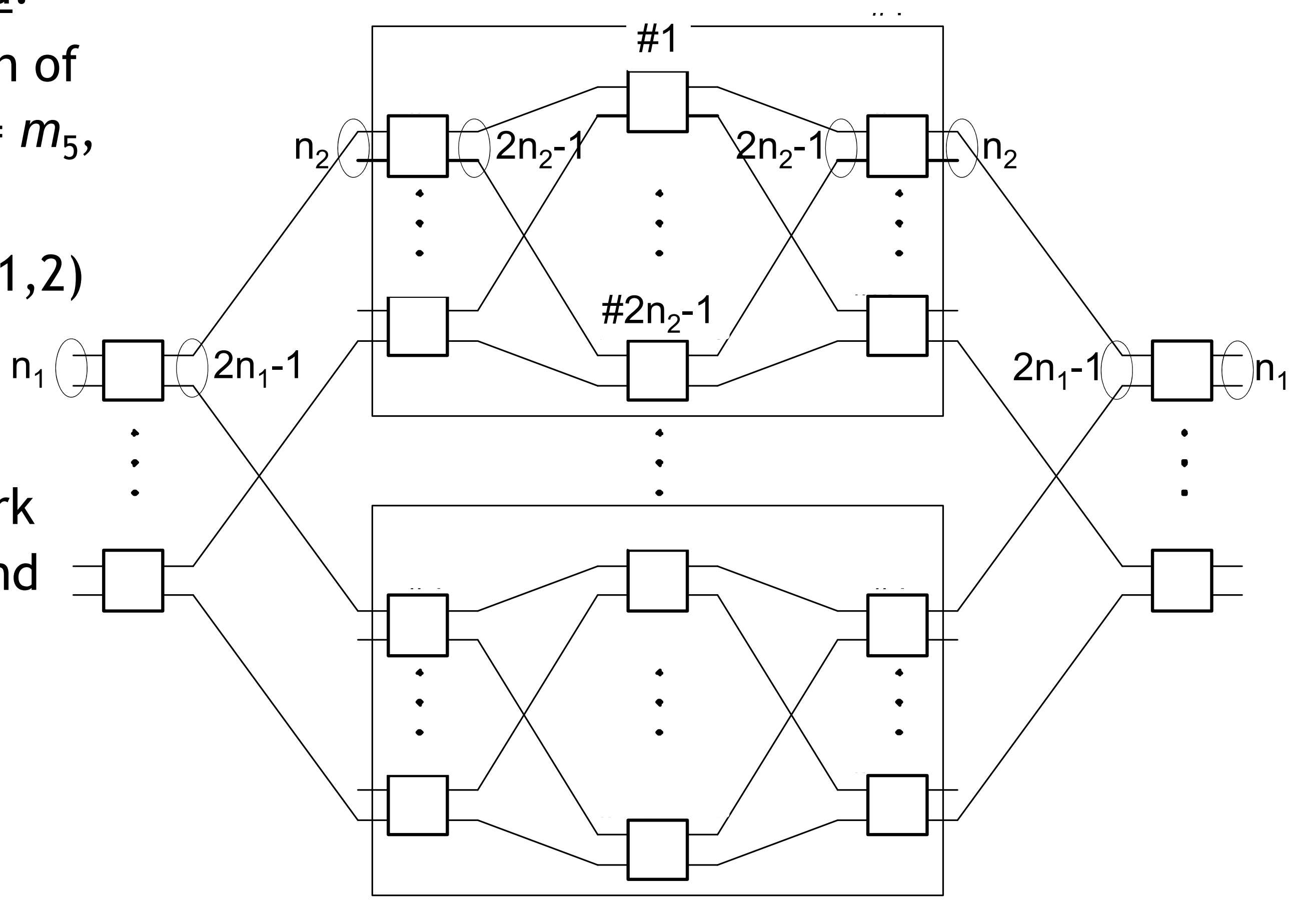
- **Five-stage SNB minimum-cost Clos network can be built**
  - Starting from a three-stage optimum Clos network
  - Expanding the central stage according to the 3-stage Clos rule
  - No longer fully connected!



Non-blocking FC networks

Extension of the Clos network – 5 stages

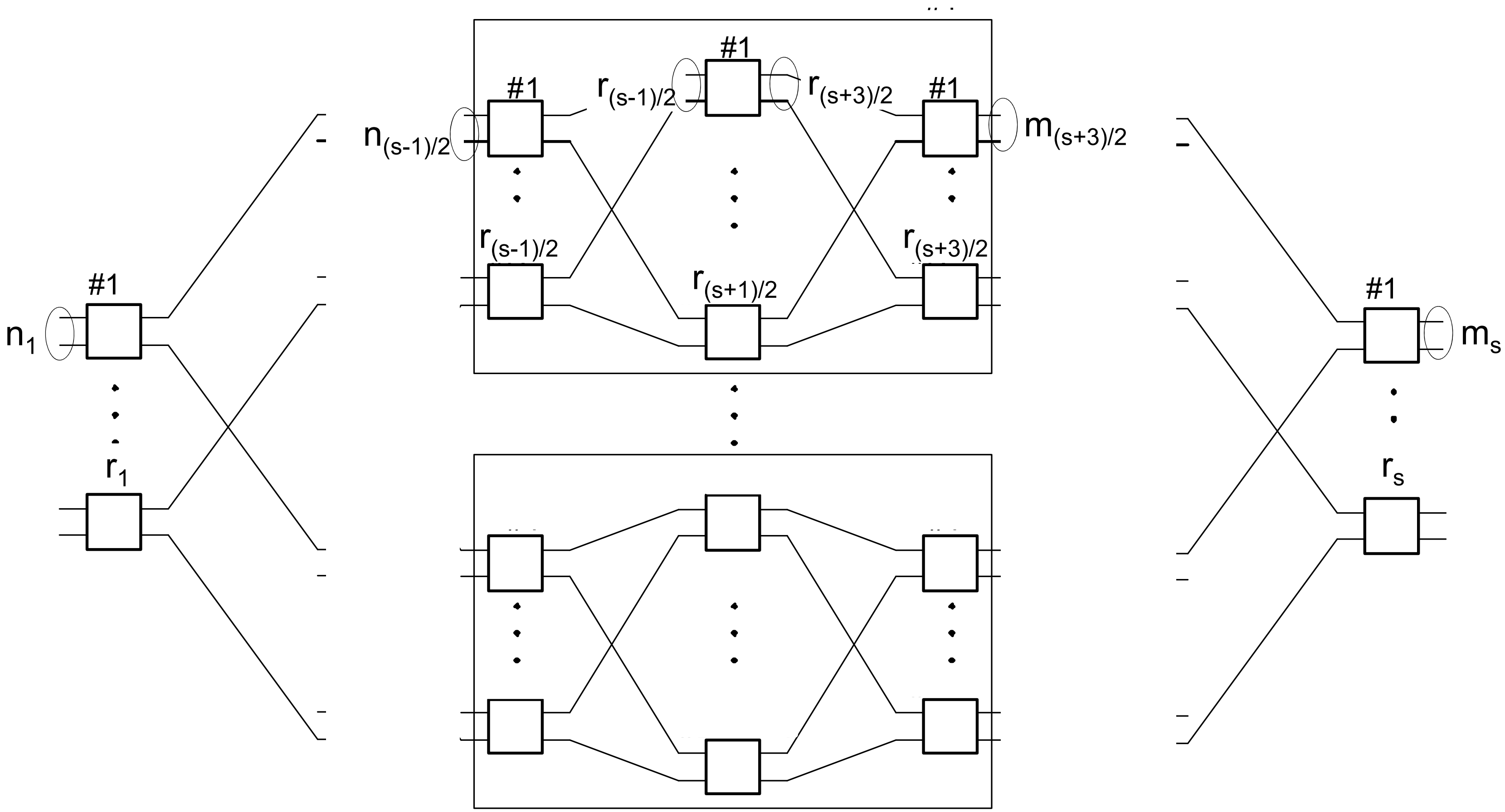
- **Five-stage SNB minimum-cost Clos network can be built**
  - Starting from a three-stage optimum Clos network
  - Expanding the central stage according to the 3-stage Clos rule
  - No longer fully connected!
  - Requires the specification of 2 parameters, that is $n_1 = m_5$, $n_2 = m_4$
  - Recall that m_i = **$2n_i$-1** (i=1,2)
- **Five-stage squared optimum Clos network**
  - First derivative of network cost with respect to n_1 and n_2 set to 0 gives
    - $N = \dfrac{2n_1 n_2^3}{n_2 - 1}$
    - $N = \dfrac{n_1 n_2^2 \left(2n_1^2 + 2n_2 - 1\right)}{\left(2n_2 - 1\right)\left(n_1 - 1\right)}$



Non-blocking FC networks

Extension of the Clos network – s stages

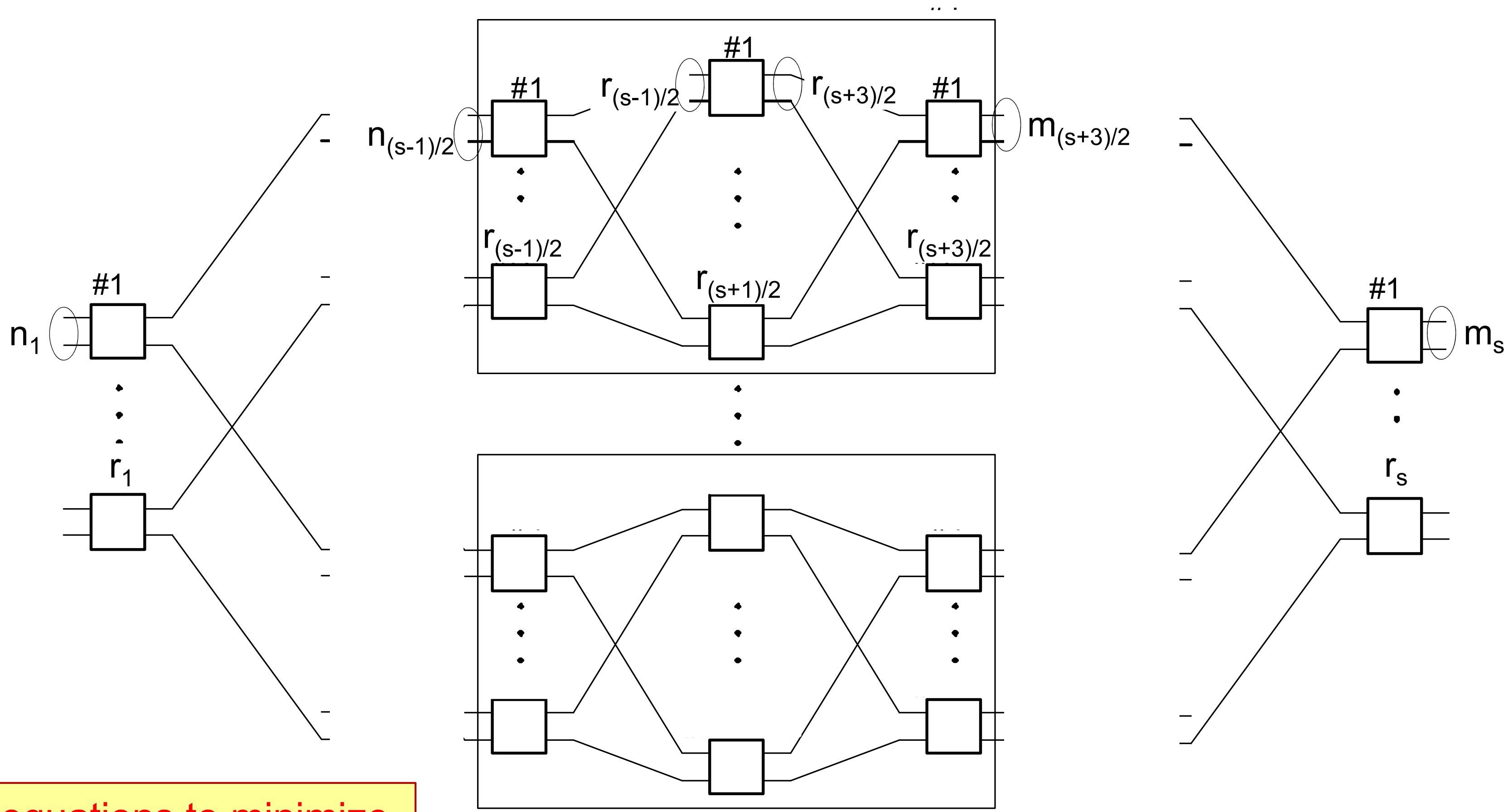
- **s-stage SNB Clos network (s odd) can be built recursively**
  - Starting from a s-2 stage Clos network
  - Expanding the central stage according to the 3-stage Clos rule (→ adding 2 stages)



Non-blocking FC networks

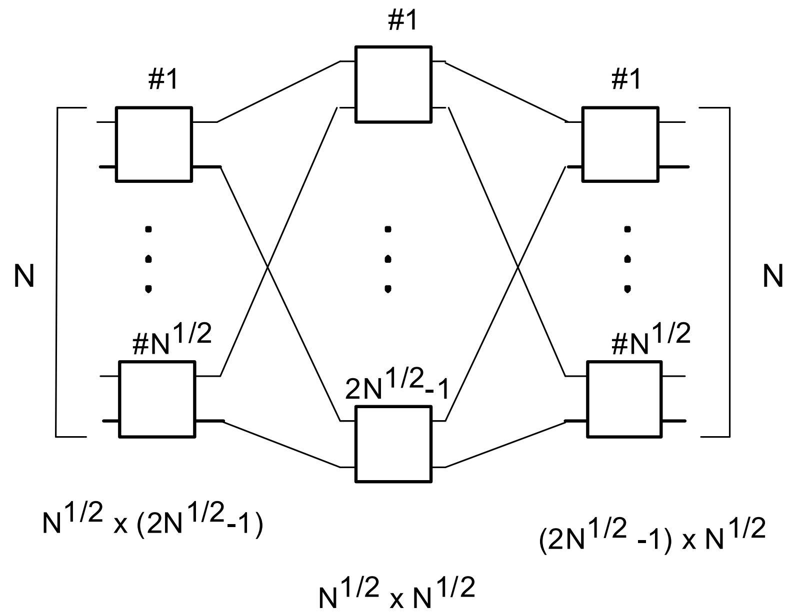
Extension of the Clos network – s stages

- **s-stage squared optimum Clos network (s odd)**
  - Requires the specification of $(s\text{-}1)/2$ parameters, that is $n_1 = m_s$, $n_2 = m_{s-1}$, ..., $n_{(s-1)/2} = m_{(s+3)/2}$
  - Recall that m_i **=** $2n_i$**-1** $(i=1,...,(s\text{-}1)/2)$



Solving all equations to minimize the cost is very complex

- **Non-blocking *s*-stage network (*s* odd)**
  - Recursive construction of the three-stage network
  - It does not minimize the network cost
- **Condition to split the N_{in} inlets**
  - $n_1 = (N_{in})^{2/(s+1)}$
  - $r_1 = \dfrac{N_{in}}{n_1}$

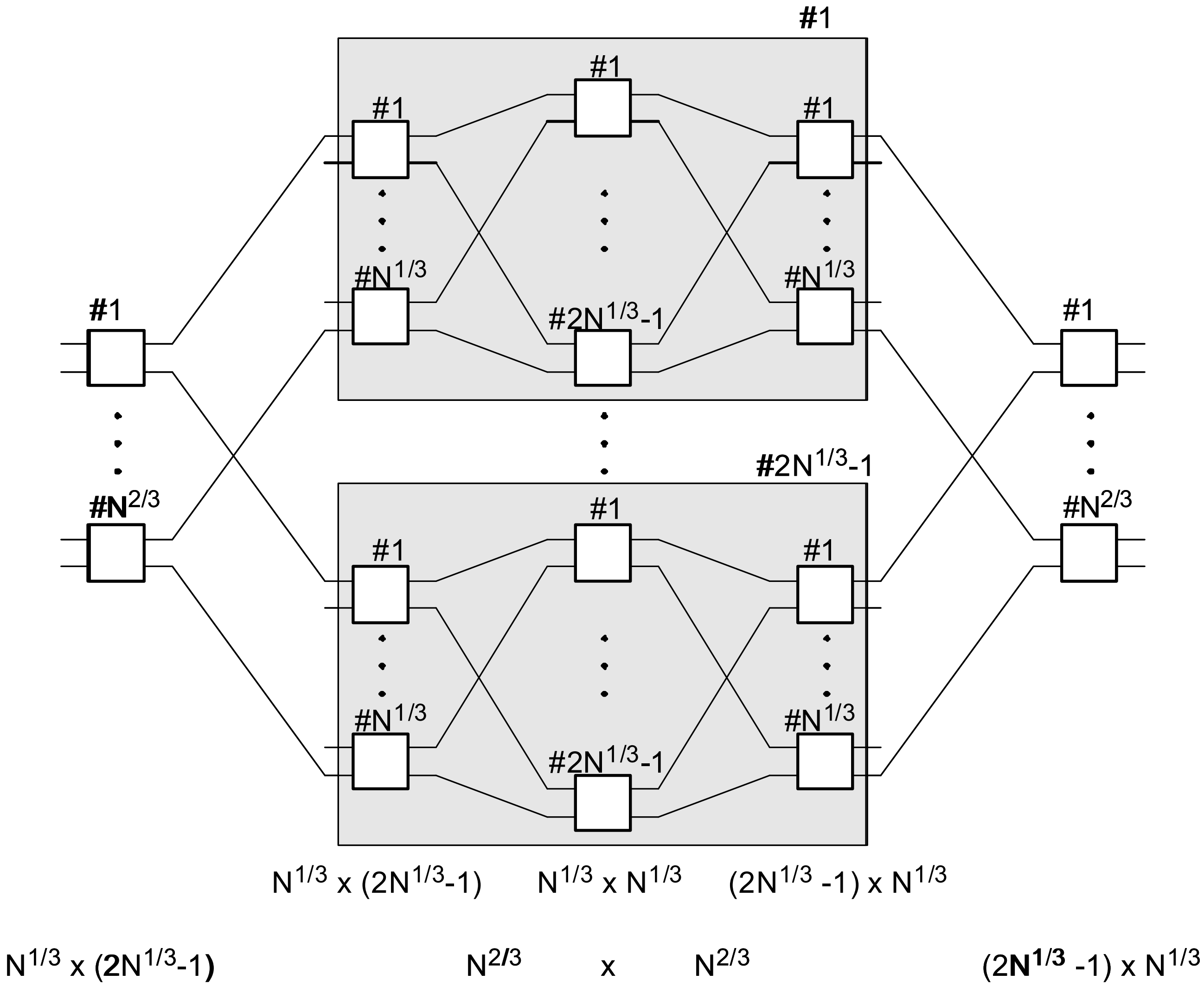


- **s = 3 stages**
  - $C_3 = (2\sqrt{N} - 1)3N = 6N^{3/2} - 3N$

Clos network

Recursive network construction

- **s = 5 stages**
  - $n_1 = (N_{in})^{2/(s+1)}$
  - $r_1 = \dfrac{N_{in}}{n_1}$



- $C_5 = \left(2N^{1/3} - 1\right)^2 3N^{2/3} + \left(2N^{1/3} - 1\right)2N = 16N^{4/3} - 14N + 3N^{2/3}$

- $s \geq 7$ **stages**

  - $C_7 = \left(2N^{1/4}-1\right)^3 3N^{1/2} + \left(2N^{1/4}-1\right)^2 2N^{3/4} + \left(2N^{1/4}-1\right)2N$
    $= 36N^{5/4} - 46N + 20N^{3/4} - 3N^{1/2}$
  - $C_s = 2\sum_{k=2}^{\frac{s+1}{2}} \left(2N^{\frac{2}{s+1}}-1\right)^{\frac{s+3}{2}-k} N^{\frac{2k}{s+1}} + \left(2N^{\frac{2}{s+1}}-1\right)^{\frac{s-1}{2}} N^{\frac{4}{s+1}}$
  - $C_s = \frac{n^2(2n-1)}{n-1}\left[(5n-3)(2n-1)^{t-1} - 2n^t\right] \quad s = 2t+1,\ N = n^{t+1}$

- **Numerical data of Clos recursive construction**

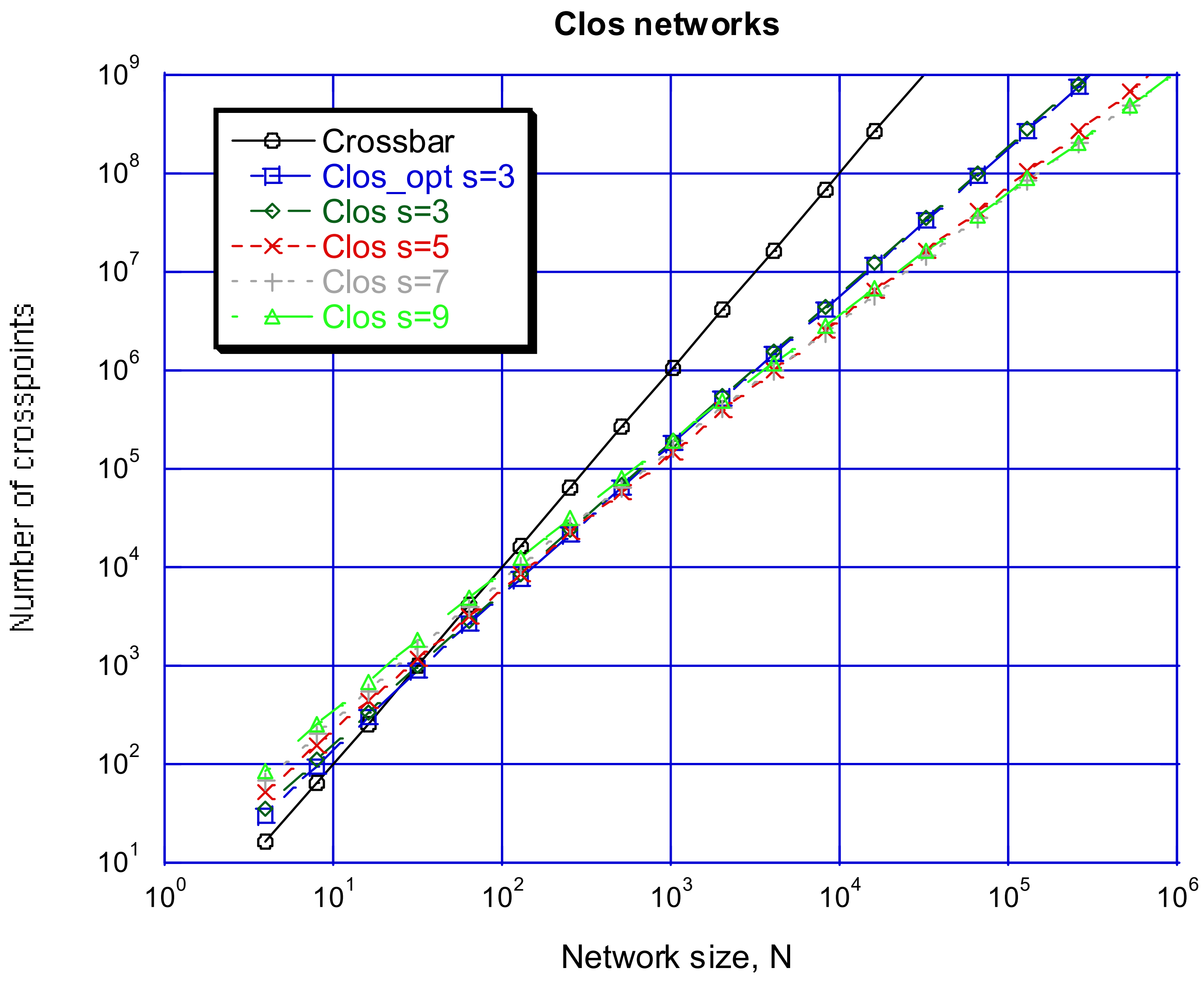
N	$s=1$	$s=3$	$s=5$	$s=7$	$s=9$
100	10,000	5,700	6,092	7,386	9,121
200	40,000	16,370	16,017	18,898	23,219
500	250,000	65,582	56,685	64,165	78,058
1000	1,000,000	186,737	146,300	159,904	192,571
2,000	4,000,000	530,656	375,651	395,340	470,292
5,000	25,000,000	2,106,320	1,298,858	1,295,294	1,511,331
10,000	100,000,000	5,970,000	3,308,487	3,159,700	3,625,165

- **Exhaustive search of minimum cost network**

N	s	n_1	n_2	n_3	C_s
100	3	5			5,400
500	5	10	5		53,200
1001	5	11	7		137,865
5,005	7	13	7	5	1,176,175
10,000	7	20	10	5	2,854,800

Clos network

Network cost



Outline

- Full-connection multistage network
- Partial-connection multistage network
- Bounds on network cost

Partial-connection networks

Classification

- **Banyan networks are self-routing but also blocking**
- **Rearrangeable networks can be built using banyan networks as the basic building block**
  - Rearrangeability conditions proven ONLY for connections set-up all together
    - → **In dynamic conditions ALL existing connections are disconnected and set-up again together with the new connection**
- **Network classes**
  - Partially self-routing
    - **Self-routing applied in some stages**
    - **Both distributed and centralized control**
    - **Techniques**
      - Horizontal extension (HE): *m* additional stages
      - Vertical replication (VR): *K* replicated planes
      - Combined vertical replication-horizontal extension (VR/HE)
  - Fully self-routing
    - **Self-routing applied in all the stages**
    - **Only distributed control**
    - **Network based on both sorting and banyan networks**

Outline

- **Full-connection multistage network**
- **Partial-connection multistage network**
  - Horizontal extension
  - Vertical replication
  - Bounds
- **Bounds on network cost**

Partially self-routing networks

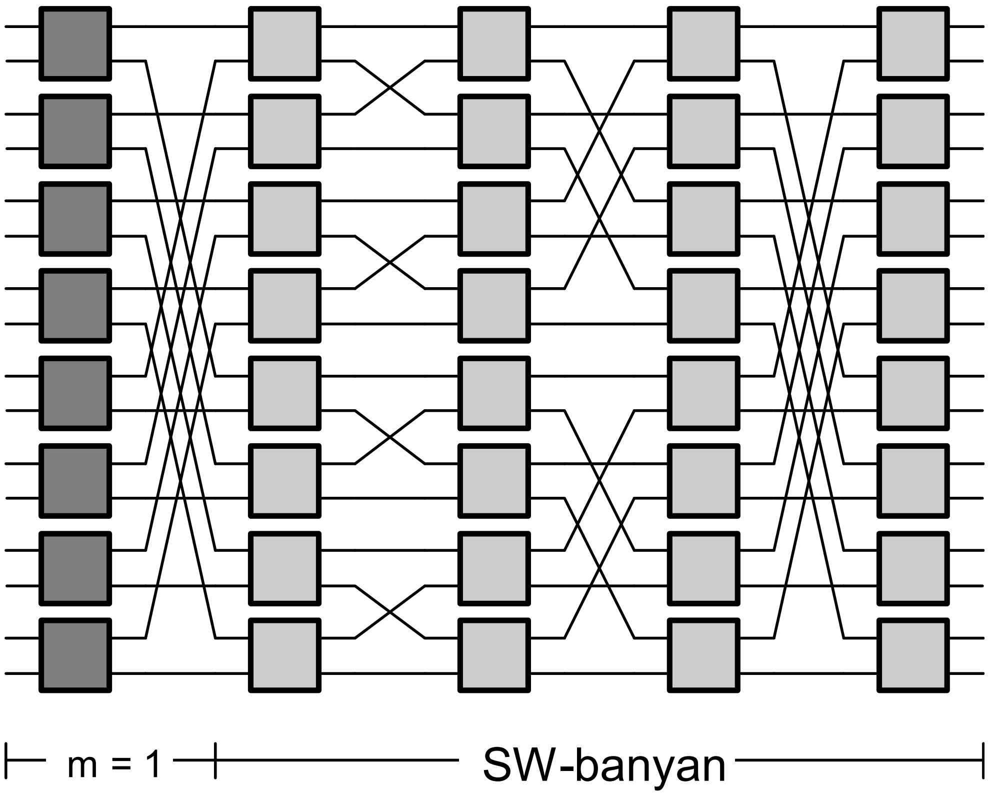
Horizontal extension

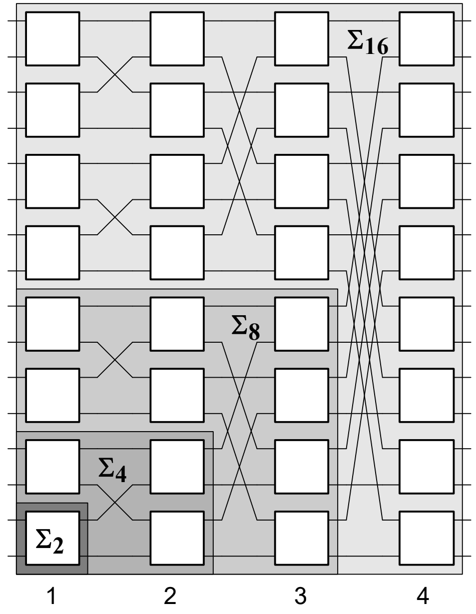
- **An extended banyan network (EBN) N x N includes**
  - The original banyan network N x N
  - m initial stages ($m \leq n$) obtained as the mirror-image of the last m stages including the preceding interstage pattern
- **2^m paths per I/O pair**
- **Distributed self-routing in the <u>n banyan stages</u>**
- **Centralized routing in the <u>additional m stages</u>**

Partially self-routing networks

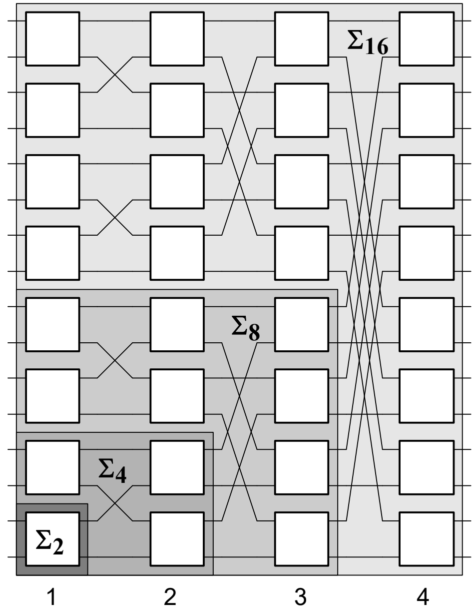
Horizontal extension

- **Starting topology: SW-banyan**
  - One additional stage



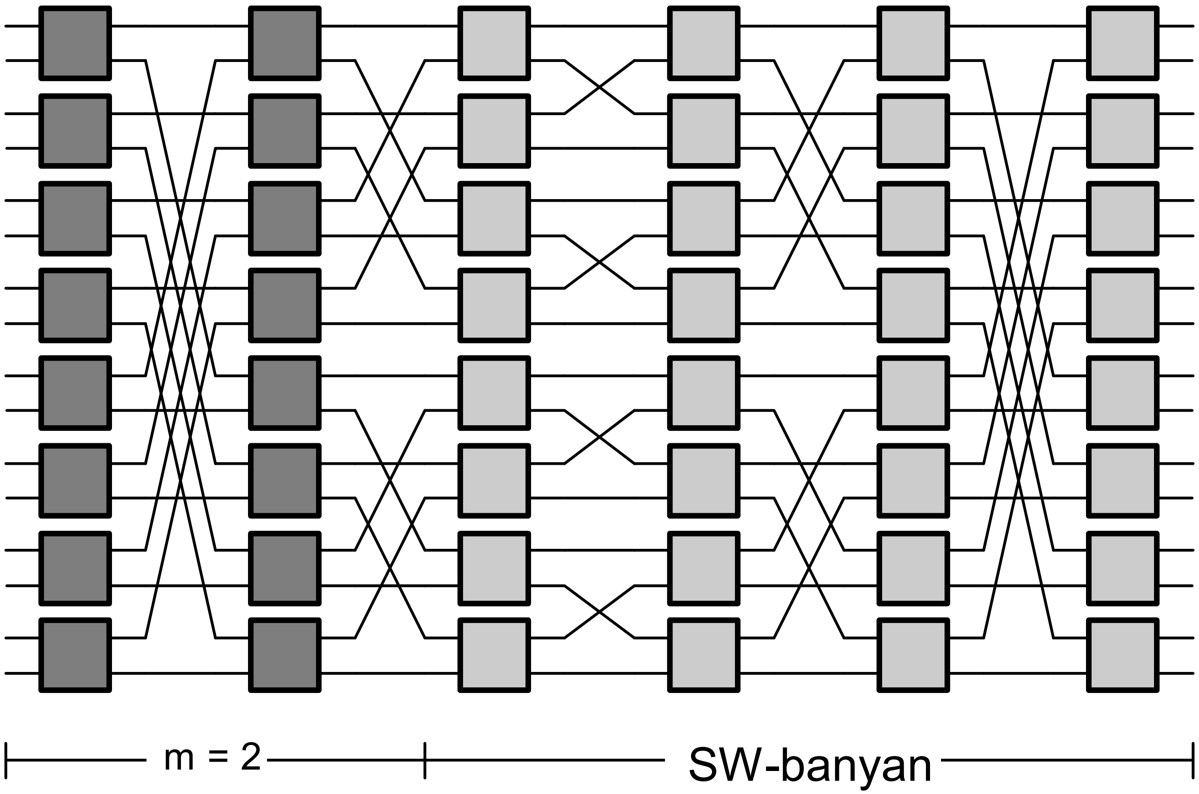


(b) - SW-banyan



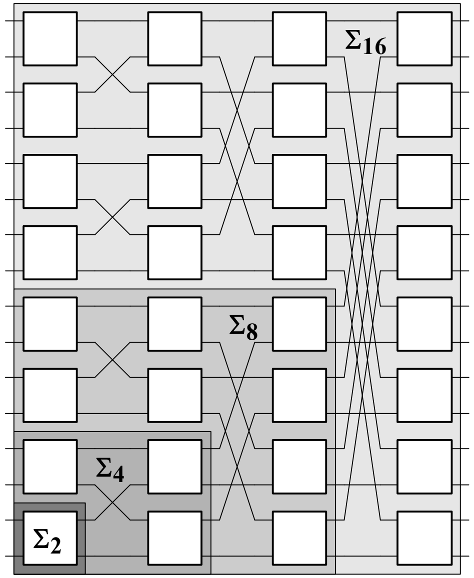
(b) - SW-banyan

- **Starting topology: SW-banyan**
  - Two additional stages



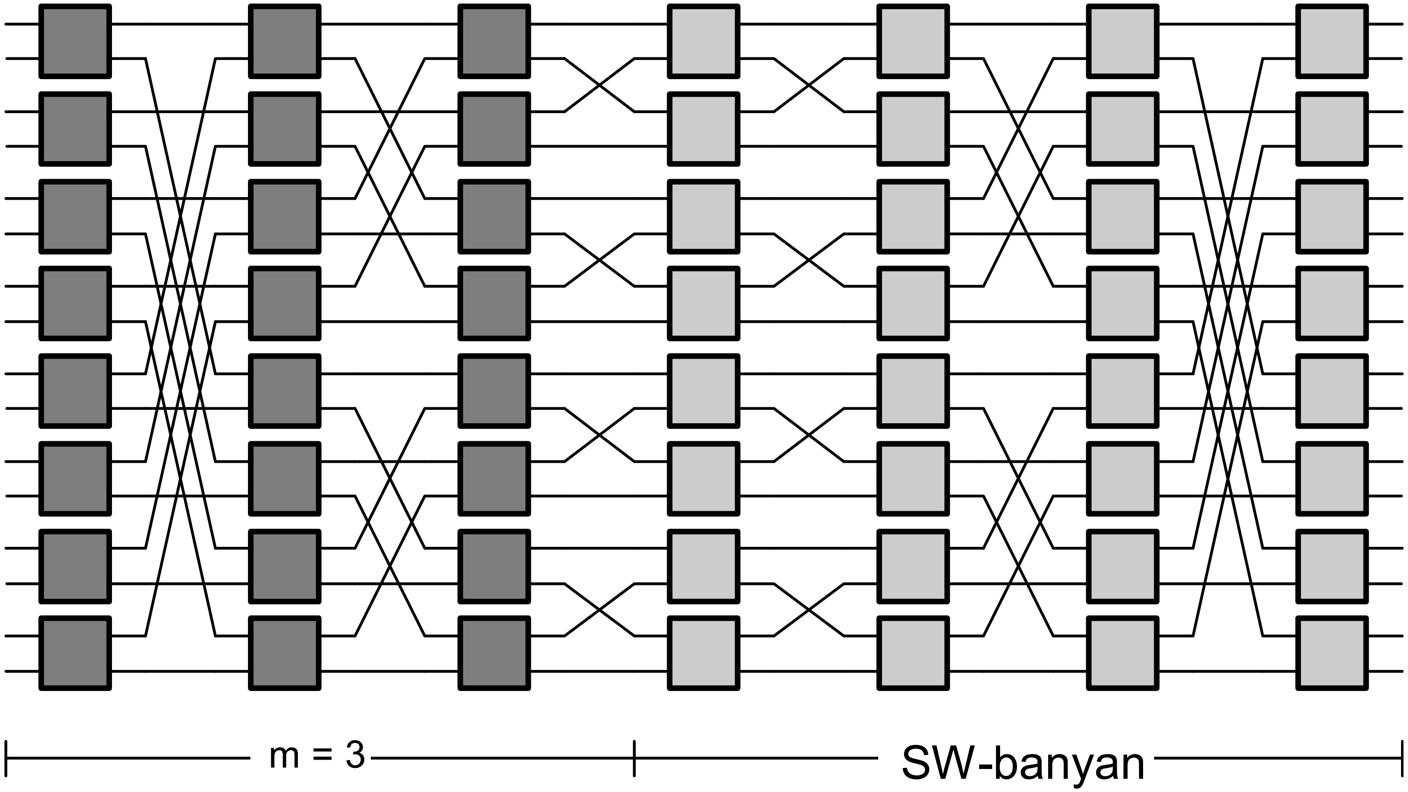
Partially self-routing networks

Horizontal extension



(b) - SW-banyan

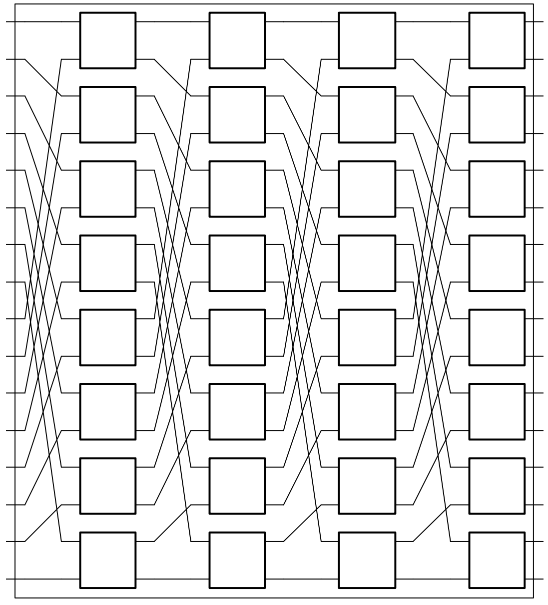
- **Starting topology: SW-banyan**
  - Three additional stages



Partially self-routing networks

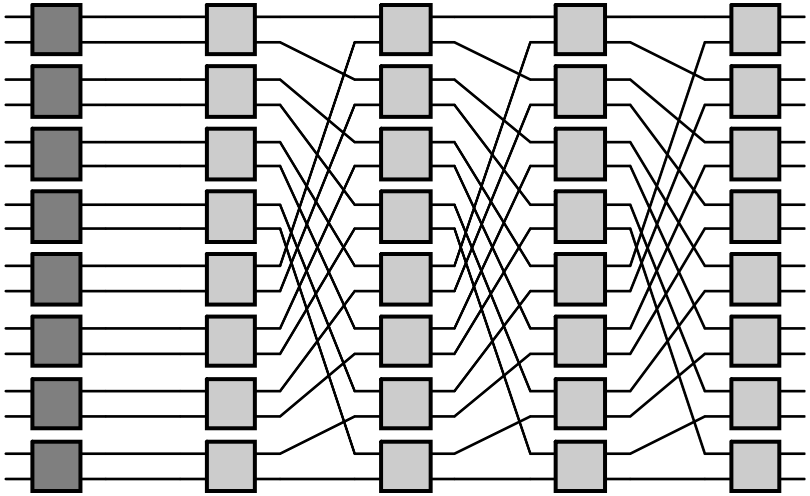
Horizontal extension

- **Any banyan network can be selected as starting topology**

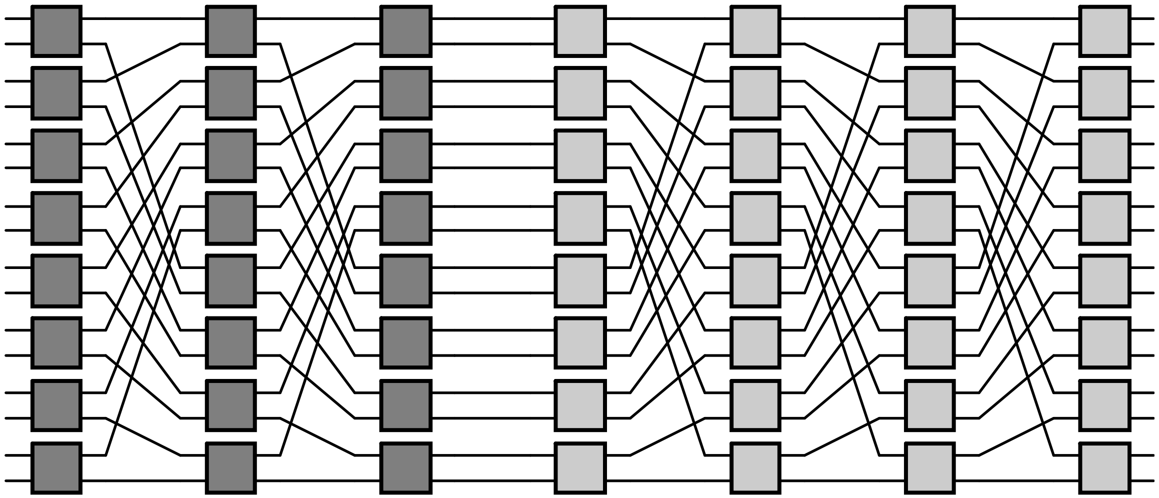


(a) - Omega

  - *Omega network*
  - *m* = 1



  - *Omega network*
  - *m* = 3

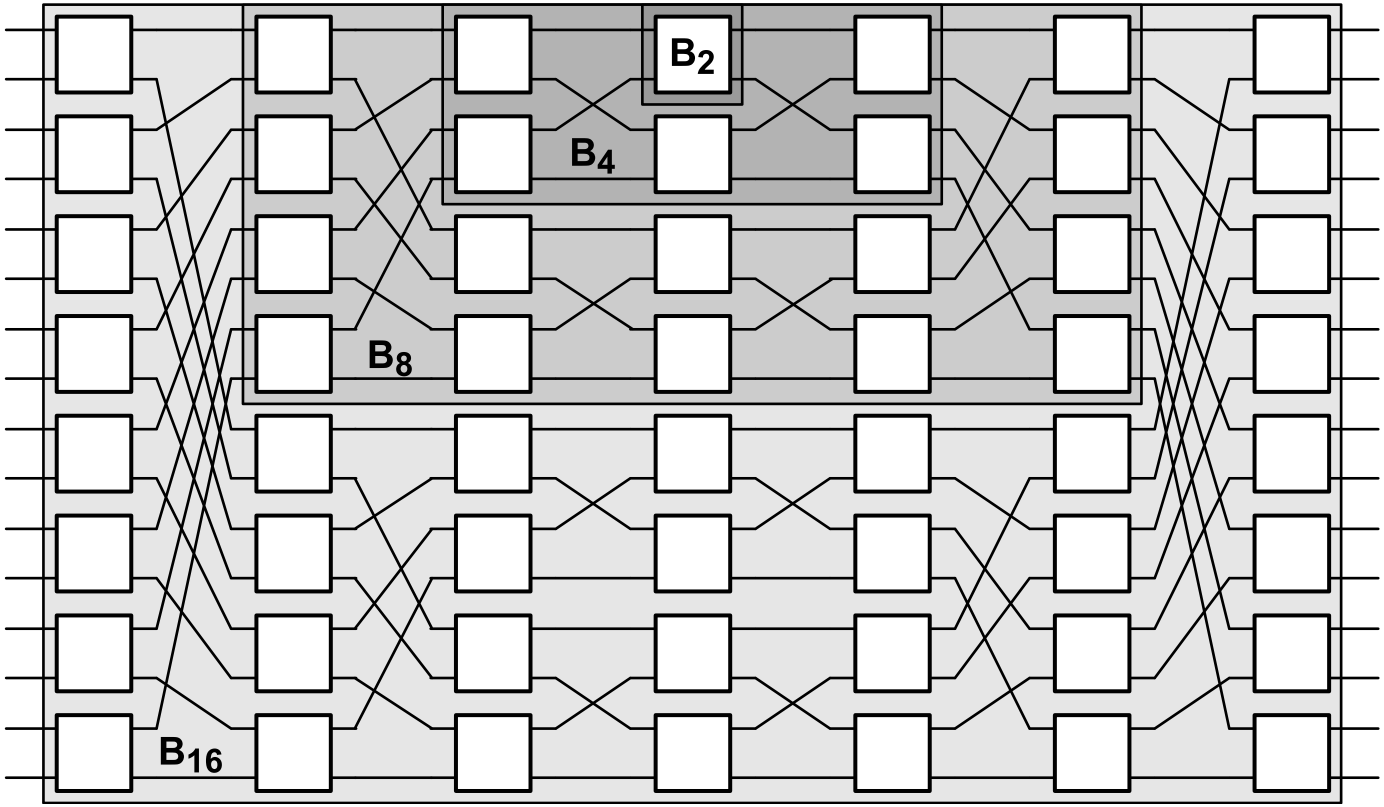


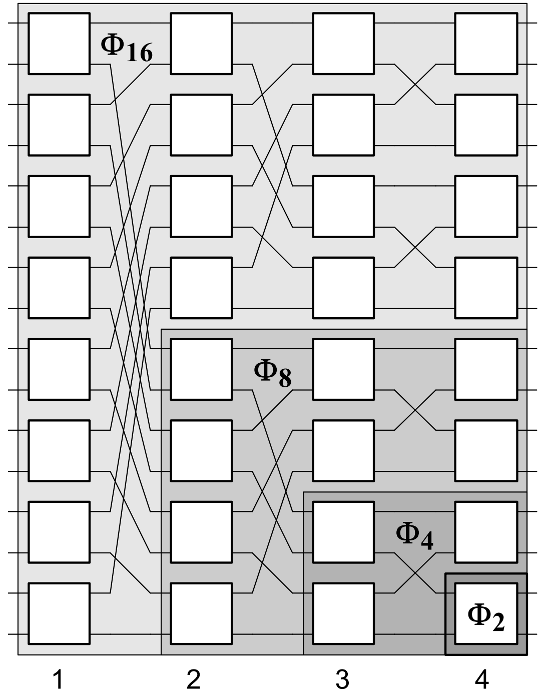
- **However the following conditions apply only to recursive EBN, that is starting from SW-banyan or reverse baseline**

Horizontal extension

Benes network – Mirror imaging

- **Benes network**
  - EBN with reverse baseline as starting topology
  - $m = log_2N-1$





(d) - Baseline

- **Recursive construction of the Benes network**
  - Three stage structure with perfect unshuffle and perfect shuffle patterns
  - Iteration for n-1 steps until the central subnetworks have size 2x2 ($N/2^{n-1} = 2$)

Horizontal extension

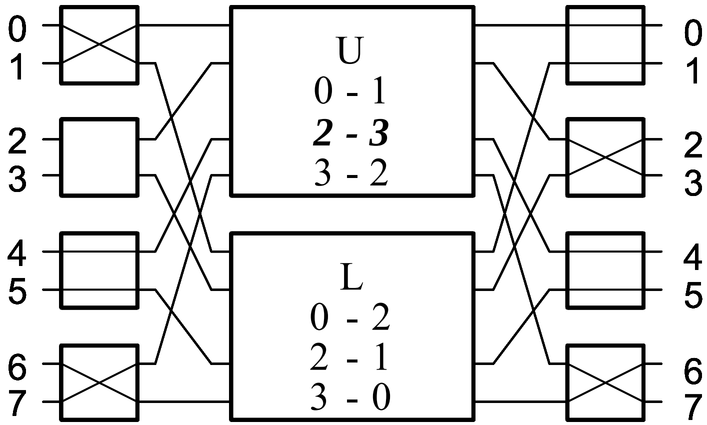
Network control of a RNB

- **Inlets/outlets**
  - Busy: a connection has been requested
  - Idle: a connection has not been requested
- **Connection set**
  - Complete: N connections requested (all busy inlets/outlets)
  - Incomplete: $k < N$ connections requested (at least 1 idle inlet/outlet)
- **Connection set-up by <u>looping algorithm</u>**

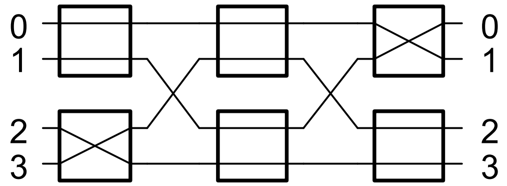
Looping algorithm

Benes network - Example

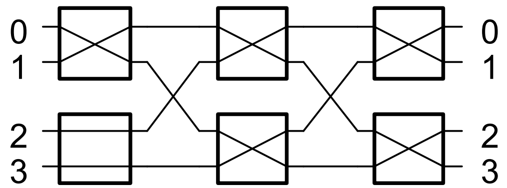
- **Connection set-up**
  - 0-5
  - 7-4
  - 6-1
  - 1-3
  - 5-2
  - 4-7



Subnetwork U

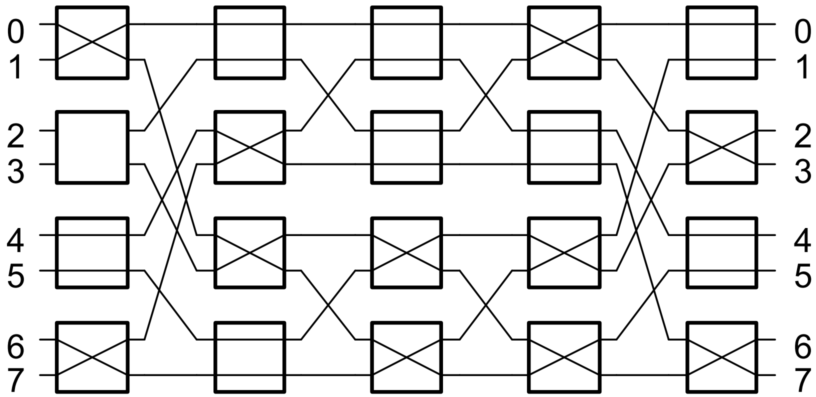


Subnetwork L



Connection set

0 - 5
1 - 3
4 - 7
5 - 2
6 - 1
7 - 4



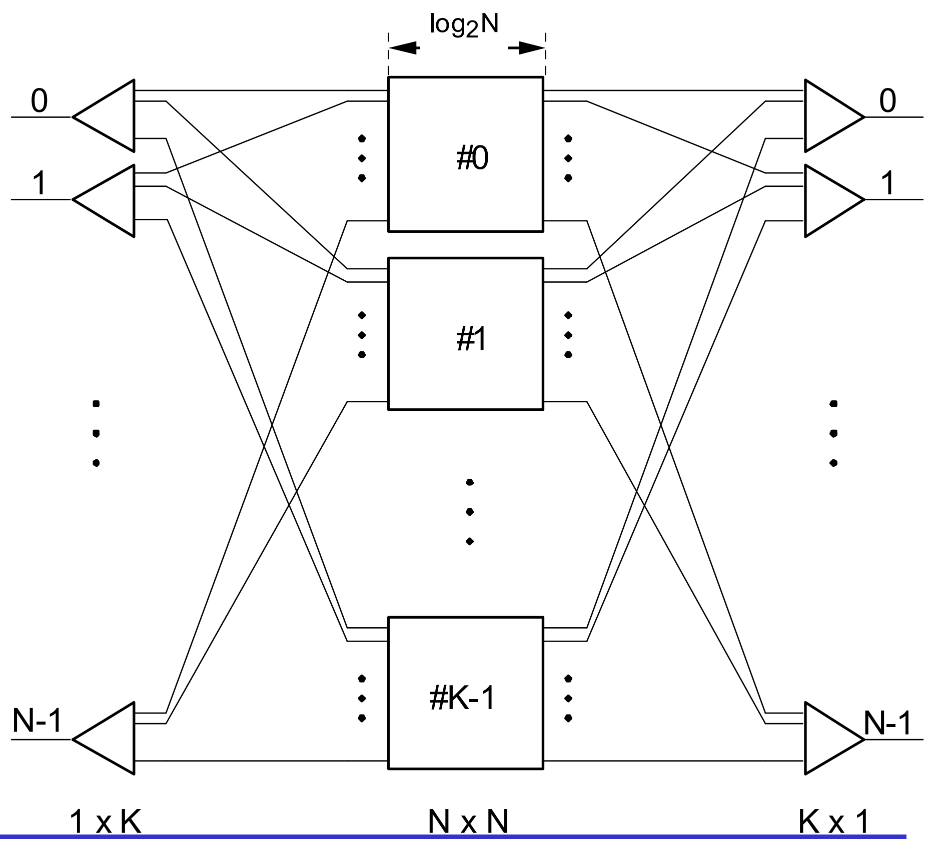
Outline

- **Full-connection multistage network**
- **Partial-connection multistage network**
  - Horizontal extension
  - Vertical replication
  - Bounds
- **Bounds on network cost**

Partially self-routing networks

Vertical replication

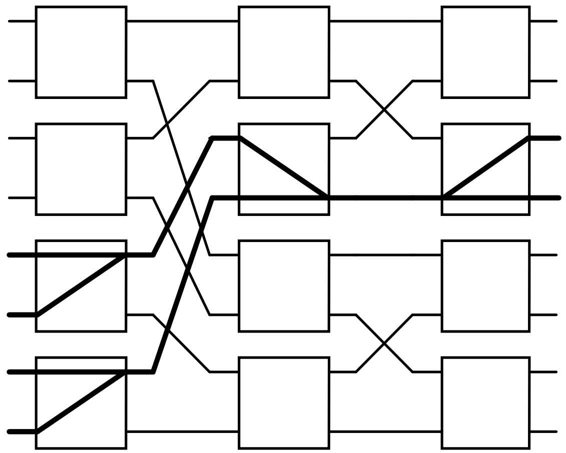
- **A replicated banyan network (RBN) *N* x *N* includes**
  - *K* banyan networks *N* x *N*
  - *N* splitters 1 x *K*
  - *N* combiners *K* x 1
- **EGS pattern in splitters-banyans and banyans-combiners connection**
- **Network control**
  - *Distributed* self-routing in the banyan planes
  - *Centralized* routing in the splitters



Vertical replication

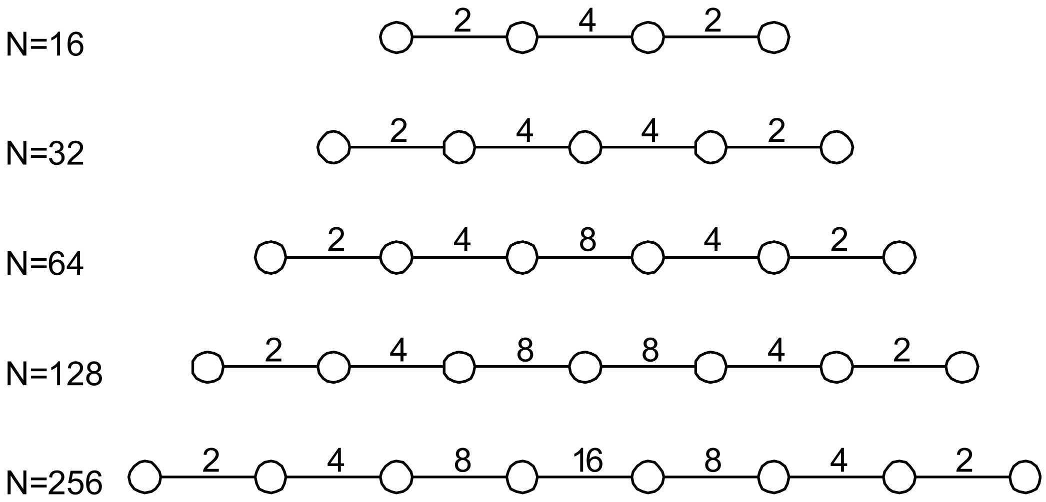
Utilization factor

- **Utilization factor u_k of a link in stage k: $u_k = \min(I_k, O_k)$**
  - I_k : number of inlets reachable from a link in stage k
  - O_k : number of outlets reachable from a link in stage k
- **All the links in an interstage pattern have the same u_k**



$u_2 = \min(I_2, O_2) = \min(4,2) = 2$

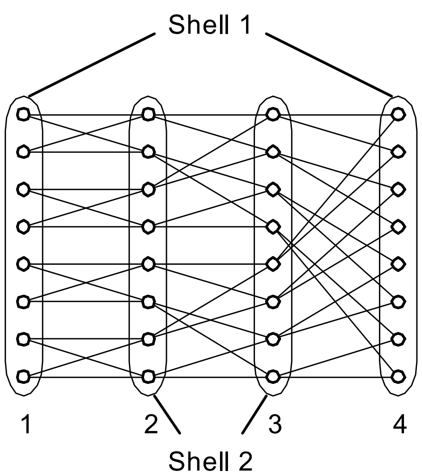
- **Utilization factors**

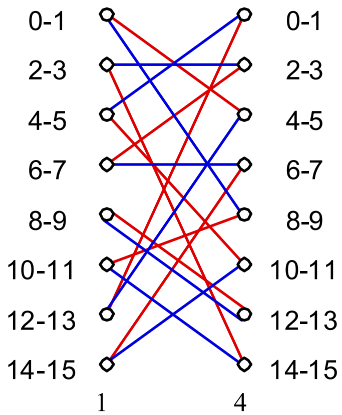


- **Maximum utilization factor**
  - $u_{max} = 2^{\left\lfloor \frac{n}{2} \right\rfloor} = 2^{\left\lfloor \frac{\log_2 N}{2} \right\rfloor}$

- **Example with *N* = 16**

0-4
1-9
2-2
3-14
4-11
5-0
6-7
7-3
8-13
9-12
10-8
11-15
12-1
13-5
14-10
15-6

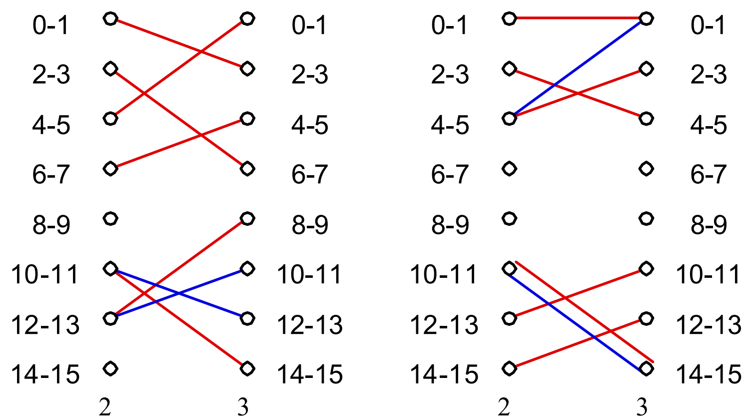




0-4	I	8-13	I
1-9	II	9-12	II
2-2	II	10-8	I
3-14	I	11-15	II
4-11	I	12-1	I
5-0	II	13-5	II
6-7	II	14-10	II
7-3	I	15-6	I

Shell 1

0-4	I	0-2
1-9	II	2-4
2-2	II	1-1
3-14	I	3-7
4-11	I	6-5
5-0	II	4-0
6-7	II	5-3
7-3	I	5-1
8-13	I	10-14
9-12	II	10-14
10-8	I	11-12
11-15	II	11-15
12-1	I	12-8
13-5	II	12-10
14-10	II	15-13
15-6	I	13-11



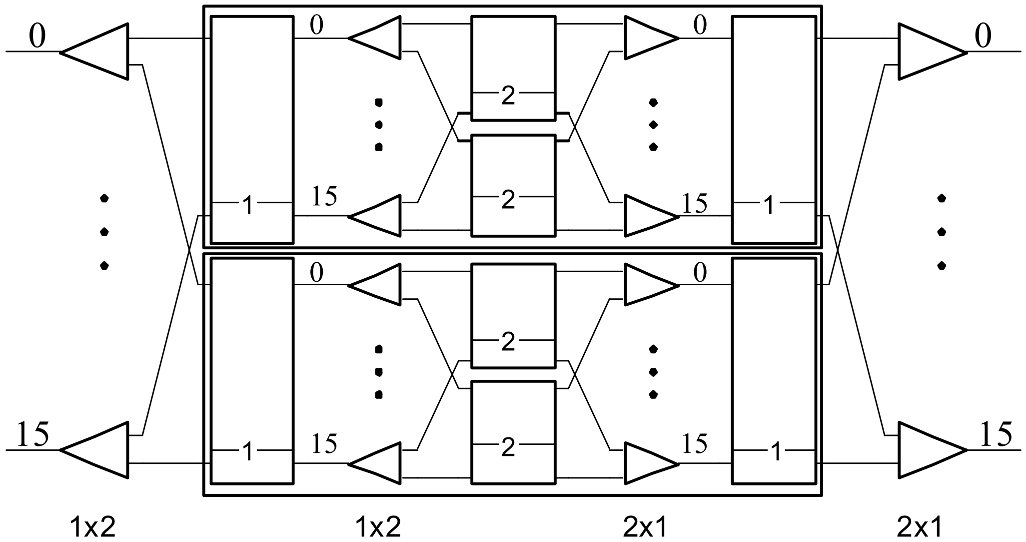
I	II
0-2	1-1
3-7	2-4
5-1	4-0
6-5	5-3
10-14	10-14
11-12	11-15
12-8	12-10
13-11	15-13

Shell 2

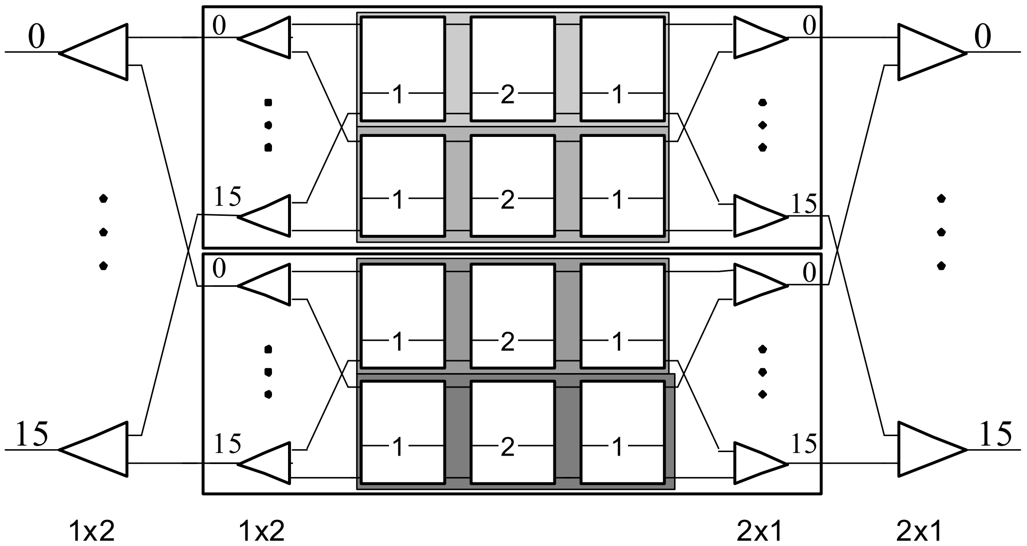
Vertical replication

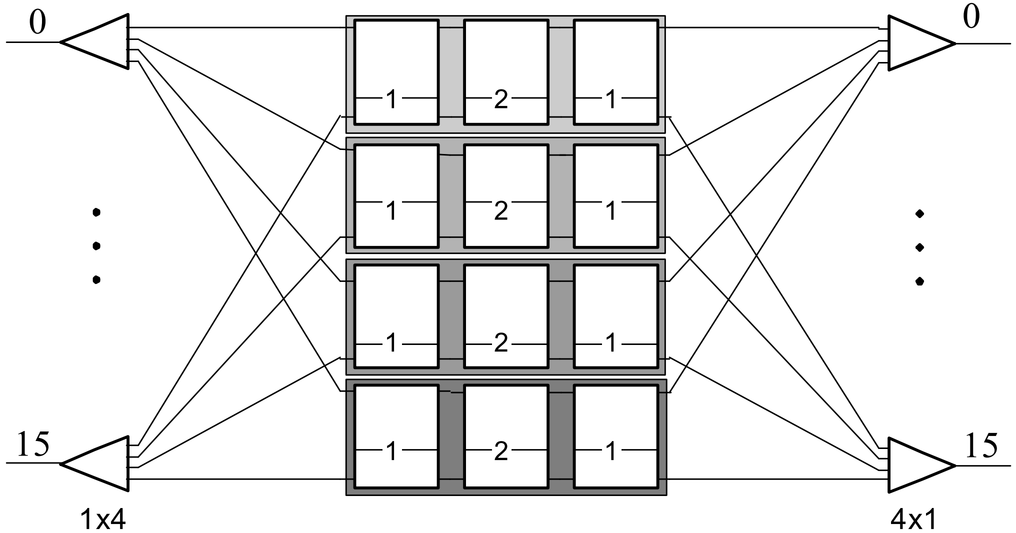
Rearrangeability conditions

- Overall network resulting from the looping algorithm



- RBN is obtained by moving and merging splitters and combiners





- **Replication factor of an *N* x *N* rearrangeable RBN**
  - Obtained from $\lfloor n/2 \rfloor$ steps of routing through shells
    - $K = 2^{\left\lfloor \frac{\log_2 N}{2} \right\rfloor}$

N	*K*
8	2
16	4
32	4
64	8
128	8
256	16
512	16
1024	32

Outline

- **Full-connection multistage network**
- **Partial-connection multistage network**
  - Horizontal extension
  - Vertical replication
  - Bounds
- **Bounds on network cost**

Rearrangeable networks

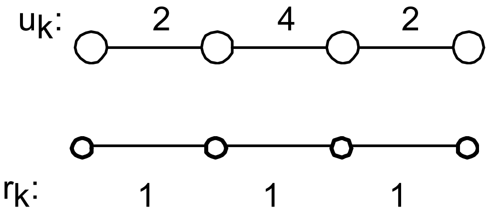
Bounds

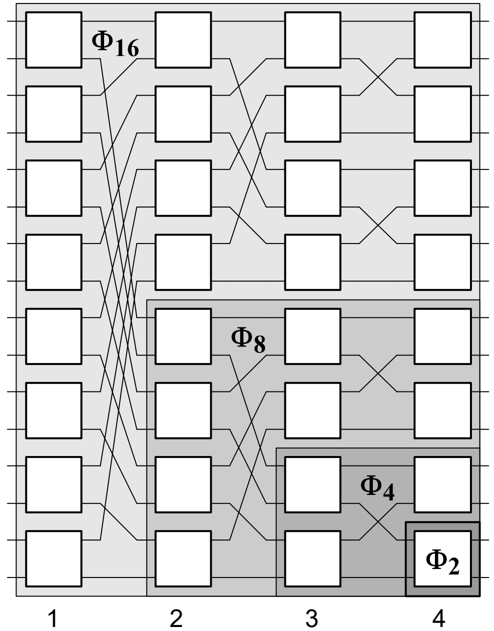
- **Channel graph of a regular network**
  - r_k = number of branches of stage k
  - u_k = maximum number of I/O paths that can share the generic link of stage k (utilization factor)
  - All the I/O paths include s+c nodes (s+c-1 branches): s switching stages + c splitter/combiner stages
- **Normalized utilization factor $U_k = u_k/r_k$**
- **Theorem. A multistage network with regular topology is rearrangeable only if $U_k \leq 1$ $(1 \leq k \leq s+c-1)$ (necessity)**
  - Proof. The number r_k of different paths where to route a connection at stage k must be at least equal to u_k so that no one link must support more than one connection at stage k

Rearrangeable networks

Examples

- $N = 16$, $s = n = 4$

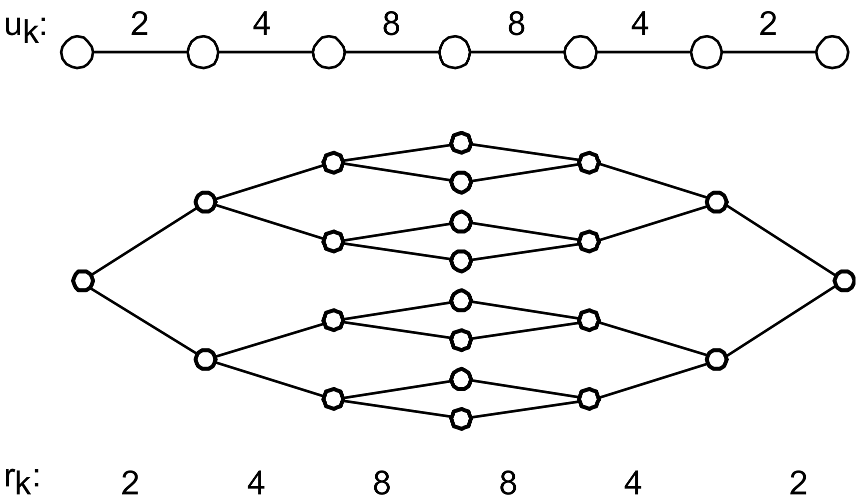


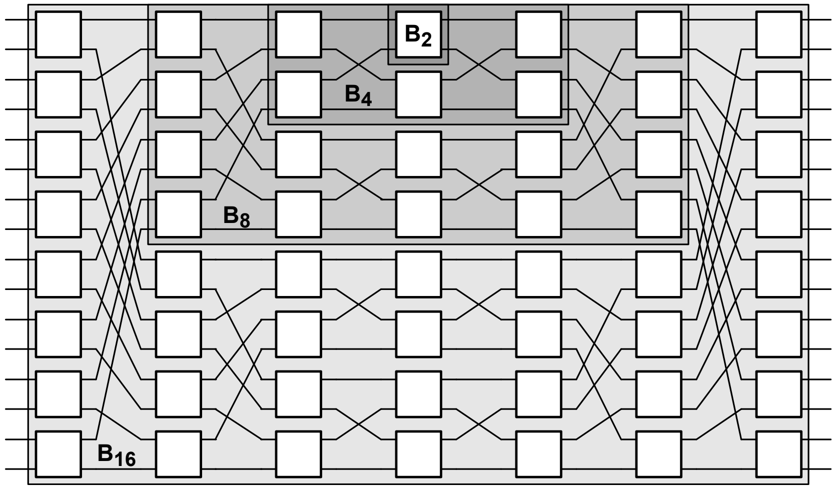


(d) - Baseline

→ $U_k > 1$

- $N = 16$, $s = 2n-1 = 7$



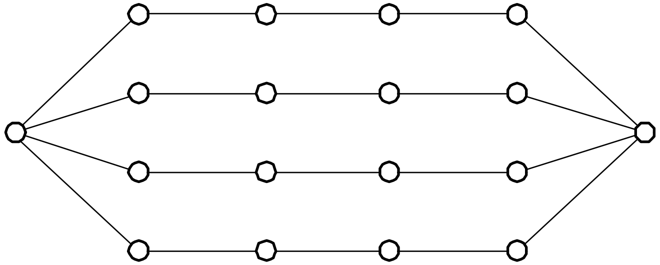


→ $U_k = 1$

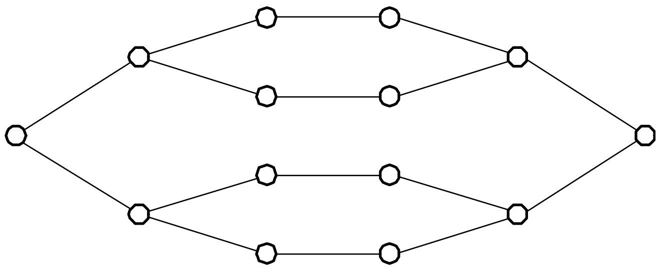
Rearrangeable networks

Examples

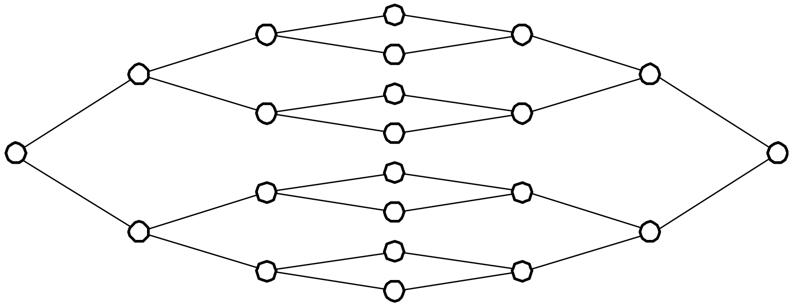
- **Rearrangeable/non rearrangeable**



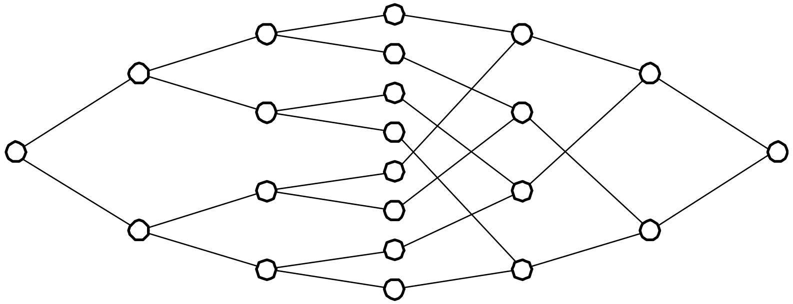
RBN N=16, K=4



EBN N=16, m=2



EBN N=16, m=3



Shuffle-exchange N=16, s=7

Number of SEs

- $2^{S_{\min}} = N!$

  $S_{\min} = \log_2 N! \cong N\log_2 N - 1.443N + 0.5\log_2 N$
- $S_{Walksman} = N\log_2 N - N + 1 \rightarrow$ very close to $S_{\min}$

Number of stages

- Network with s stages of $N/2$ SEs : $S = s\frac{N}{2}$

  $\Rightarrow s_{\min} = \left\lceil \frac{S_{\min}}{N/2} \right\rceil = 2\log_2 N - 2$
- Rearrangeable network : $u_{k\max} = 2^{\lfloor (n+m)/2 \rfloor}$; $r_{k\max} = 2^m \;\; (m \le n)$

  $\frac{2^{\lfloor (n+m)/2 \rfloor}}{2^m} \le 1 \rightarrow \;\; m \ge n-1$

  $\Rightarrow s_{\min-rearr} = 2\log_2 N - 1$

  $s_{Benes} = s_{\min-rearr}$

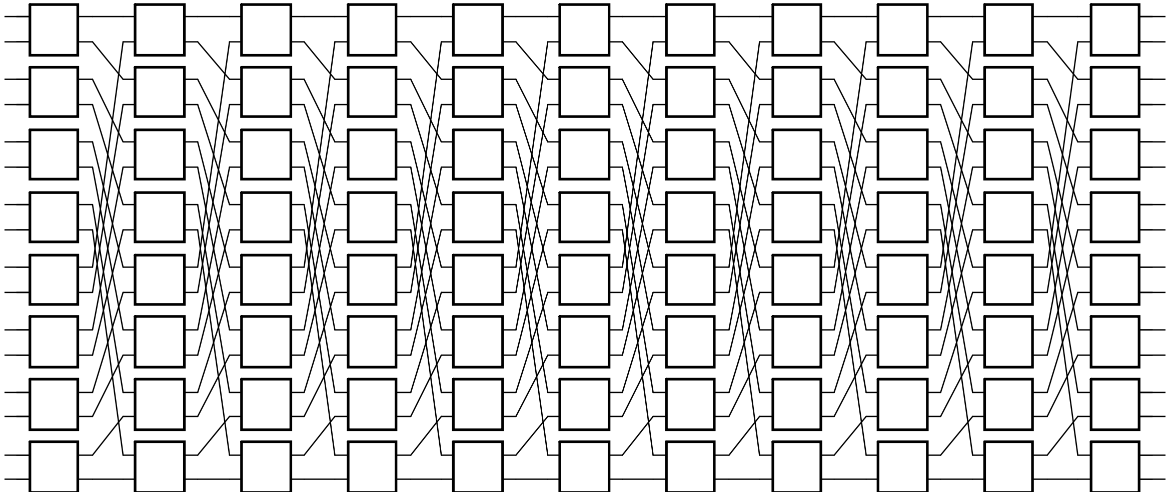
Crosspoint growth order

- $VR : \# XP = O\left(N^{\frac{3}{2}}\log_2 N\right)$
- $HE : \# XP = O(N\log_2 N)$

Rearrangeable networks

Bounds

- **The permutations realized by an Ω network followed by a δ permutation can be realized by the cascade of two Ω networks → total number of stages $n+n+(n-1) = 3n-1$**



- **Other limits on the number of stages of Shuffle-exchange networks**
  - Huang, Tripathi: $3\log_2 N-3$
  - Varma, Raghavendra: $3\log_2 N-4$
  - Stone: $(\log_2 N)^2$
    - **An arbitrary permutation is set-up directly by the sorting operation (any input can set-up a connection to any output)**

Outline

- **Full-connection multistage network**
- **Partial-connection multistage network**
- **Bounds on network cost**

- **Network cost C**
  - Using PC networks: $C(N,N) = 4N\log_2 N + O(N)$
    (Benes/Waksman)
  - Using 3×3 SEs:
    $C(N,N) \leq 6N\log_3 N + O\left(N(\log_2 N)^{1/2}\right) = 3.79N\log_2 N + O(\ldots)$
    (Pippenger [Pip78])

Rearrangeable networks

Network cost