All Permutations From Shuffle Stages

Consider network composed of s shuffle stages.

Problem: find minimum s for which all permutations admitted.

Two Solutions:

1: Transform Beneš network into shuffles.

2: Show omega/inverse omega can admit all permutations.

Transform omega/inverse omega into shuffles.

Transformation of Beneš Network

Plan

- Start with Beneš network.
- Replace parts of Beneš network with functionally equivalent networks.
- Stop when resulting network consists only of shuffles.

Solution

1: Start with Beneš network.

Beneš

2: Replace left n stages of the Beneš with a baseline.

Note that the left half of the Beneš is equivalent to a baseline.

3: Replace right $n-1$ stages of the Beneš with an inverse baseline.

Baseline Baseline^{-1}

Note that there is now an additional stage.

The additional stage is redundant.

4: Replace the baseline with a ρ link pattern followed by an omega network.

$$\rho\Omega \text{ Baseline}^{-1}$$

It can be shown that a baseline is terminal equivalent to $\rho\Omega$.

5: Replace the inverse baseline with a ρ link pattern followed by an omega network.

$$\rho\Omega\rho\Omega$$

Recall that a baseline and inverse baseline are terminal equivalent.

6: Replace the middle ρ with $\lambda_1 \upsilon \lambda_2$

$$\rho\Omega\lambda_1\upsilon\lambda_2\Omega$$

where $\lambda_1\upsilon\lambda_2 = \rho$, $\lambda_1 \in \Lambda$, $\lambda_2 \in \Lambda$, and $\upsilon \in \Upsilon$.

Since ρ is a linear permutation it can be factored this way.

7: Replace $\lambda_2\Omega$ with Ω.

$$\rho\Omega\lambda_1\upsilon\Omega$$

Recall lower-triangular permutations, Λ, are left-invariant.

8: Replace λ_1 with Ω.

$$\rho\Omega\Omega\upsilon\Omega$$

Yes, a link pattern is replaced with an entire omega network.

9: Replace $\Omega\upsilon$ with Ω.

$$\rho\Omega\Omega\Omega$$

Recall upper-triangular permutations, Υ, are right-invariant.

10: Replace $\rho\Omega\Omega\Omega$ with $\Omega\Omega\Omega$.

$$\Omega\Omega\Omega$$

The resulting network consists of only shuffles.

Using Fewer Stages

It is possible to remove two or three stages.

It may be possible to realize all permutations in $2n-1$ stages.

Use of fewer than $3n$ stages will not be covered.

Transformation of Omega/Inverse Omega

Plan

- Prove that omega/inverse omega can admit all permutations.
- Transform omega/inverse omega to shuffles.

Omega/Inverse Omega Network

Constructed by cascading an omega and inverse omega network.

Can realize permutations in set $\Omega\Omega^{-1}$.

Theorem: $\Omega\Omega^{-1} = \Sigma$.

Proof:

By induction on n.

Let Ω_n and Ω_n^{-1} denote n-stage omega and inverse omega networks.

Basis: $\Omega_1\Omega_1^{-1} = \Sigma_{2^1}$.

Trivially true.

Inductive hypothesis:

If $\Omega_{2,n-1}\Omega_{2,n-1}^{-1} = \Sigma_{2^{n-1}}$ then $\Omega_{2,n}\Omega_{2,n}^{-1} = \Sigma_{2^n}$.

Proof of IH:

Draw omega/inverse omega using recursive forms.

This will have four stages.

The four (recursive) cells in the two inner stages form two omega/inverse omega networks.

Consider each of these omega/inverse omega networks as a unit.

Then the larger omega/inverse omega network is equivalent to:

a shuffle followed by a Clos network, followed by an inverse shuffle.

It's known that the Clos network can realize all permutations.

Therefore the network obtained by adding the shuffles can also realize all permutations.

That is, the omega/inverse omega can realize all permutations.

Next, convert $\Omega\Omega^{-1}$ into shuffles.

Note that $\Omega^{-1} = \rho\Omega\rho$.

Therefore, $\Omega\Omega^{-1} = \Omega\rho\Omega\rho$.

The remainder of the transformation is similar to the Beneš network transformation.

Generalized Connectors

There are several ways to construct these:

- Single crossbar. (When you're rich alot of problems are easy to solve.)
- Clos-type generalized connectors.
- Networks called *fanout/concentrators* which we may not have time to cover.
- *Pack/Copy/Permute* networks.

Pack/Copy/Permute Generalized Connectors.

This network described by Ofman.

Network is rearrangeable (for generalized connection assignments).

Network has three parts:

- Pack: packs those inputs having requests to consecutive links.
- Copy: fans out (copies) inputs to multiple links.
- Permute: routes to proper output.

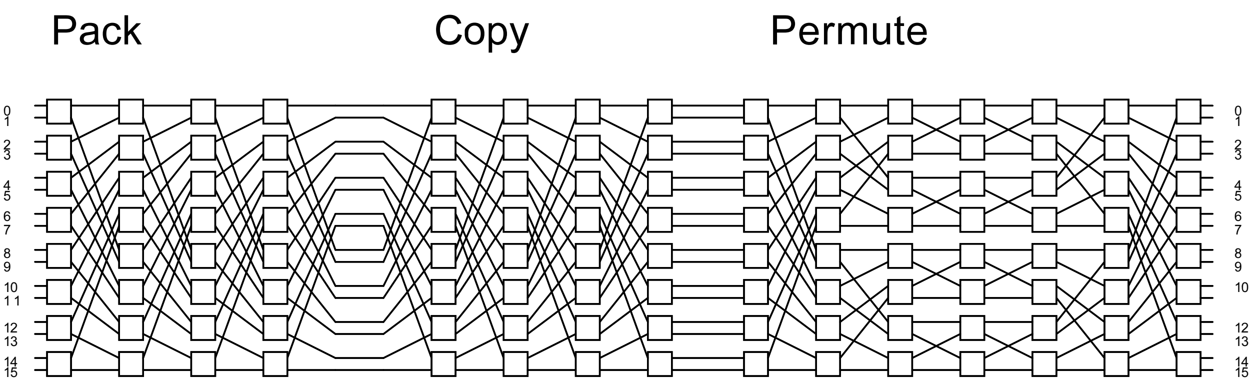
Construction

The pack part is an inverse omega network.

The copy part is an omega network.

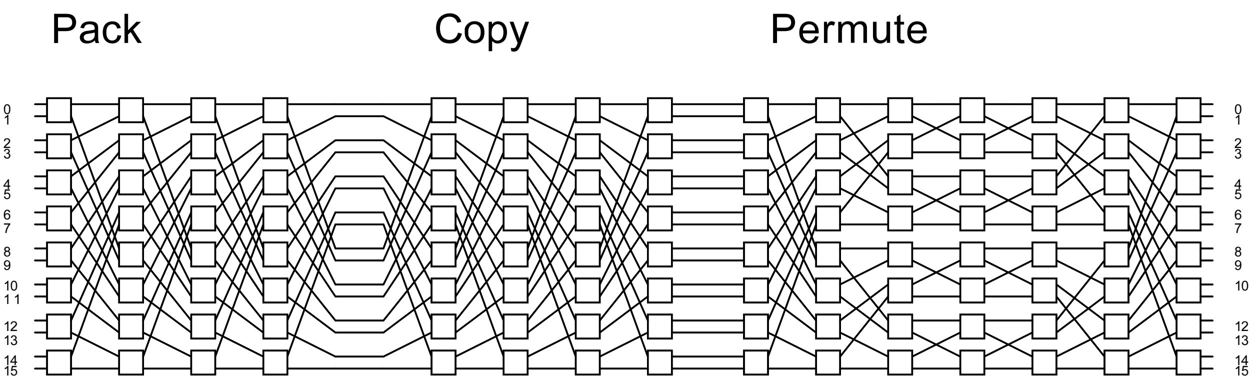
The permute part is (sometimes) a Beneš network.

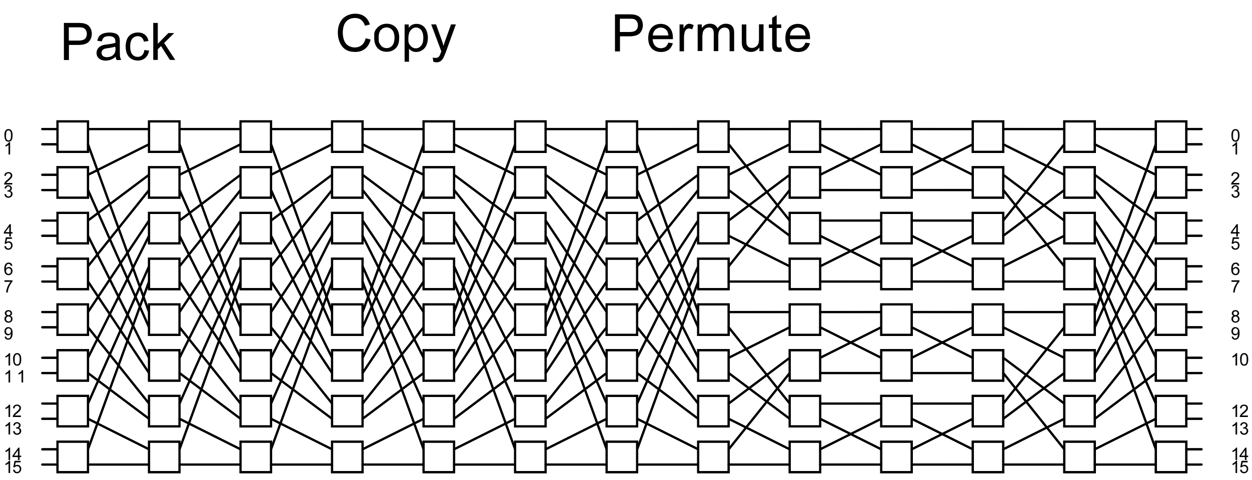
Straightforward Construction



Slightly Lower-Cost Version

Stages where parts meet can be removed, eliminating two stages.





Cost

Straightforward version: $4n2^{n-1}$ cells.

Lower-cost version: $(4n-3)2^{n-1}$ cells.

Routing CA Γ

- Route each input having a request from GC inputs (same as pack-part inputs) to consecutive copy-part inputs.

  (Using a packing CA).

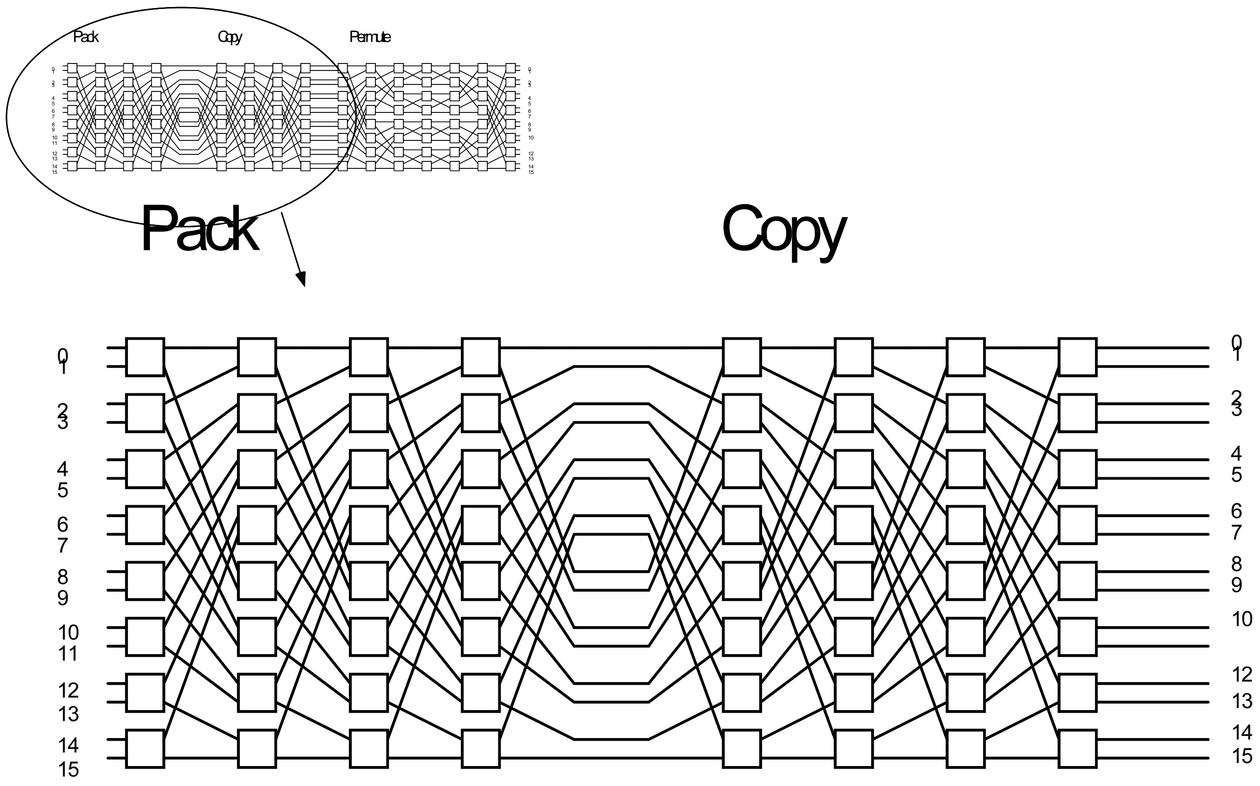
- Route $A = (a, \alpha) \in \Gamma$ at copy-part input to $|\{\,(a,\chi) \mid (a,\chi) \in \Gamma\,\}|$ consecutive copy-part outputs.

  (That is, make needed number of copies using a copy connection assignment.)

- Finally, route $A = (a, \alpha)$ to output α through permute part.

  (Use dummy requests to create a permutation if necessary, then use looping algorithm to route Beneš network.)

- Done.

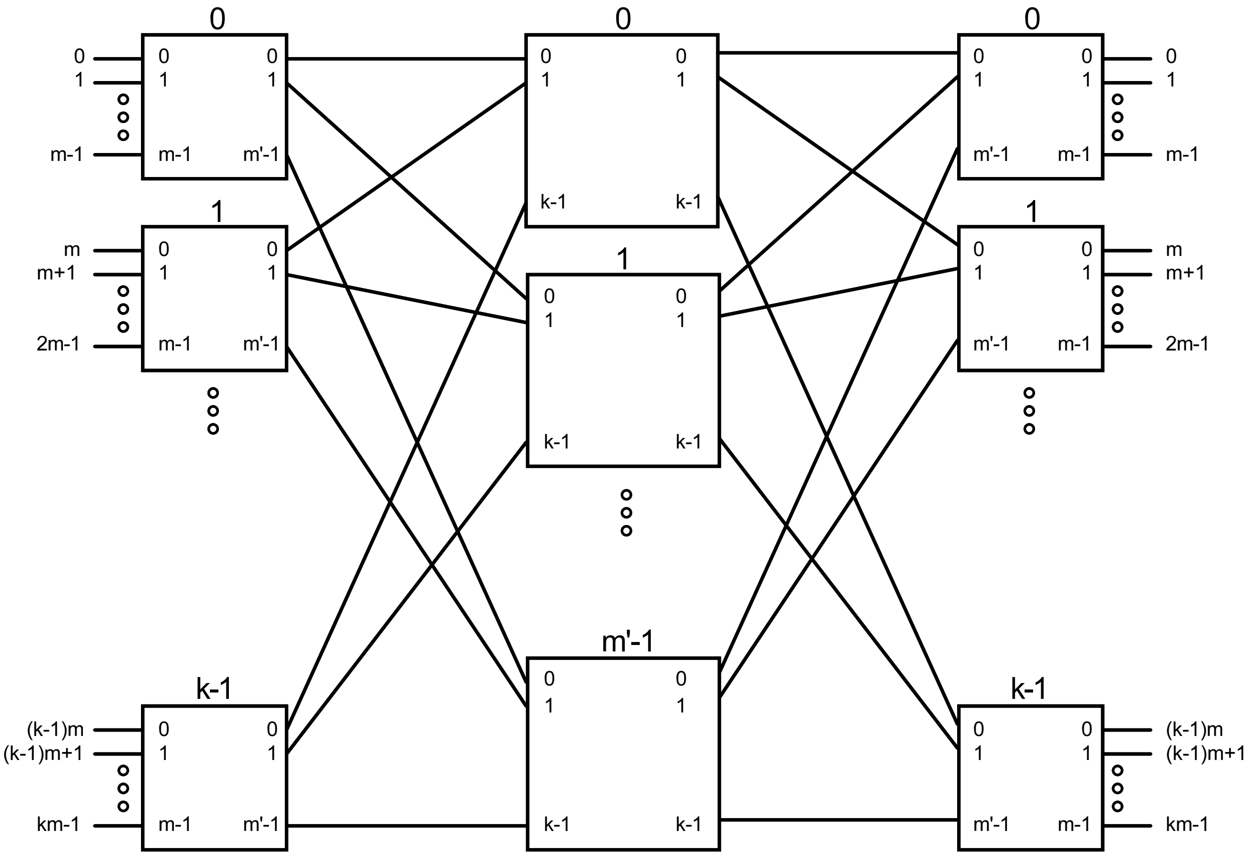


Clos-Type Generalized Connector

Described by Yang and Masson in 1991.

Is a non-blocking d-limited generalized connector.

Construction



Topology of Clos network in which $m' > (m-1)(x + d^{1/x})$,

where $0 < x \leq \min\{\, m-1, d \,\}$ and $0 < d \leq k$.

(Choice of x determines cost and performance.)

Cost

The minimum cost of a $2n+1$ stage Clos-type GC is bounded by

$$C(N, 2n+1) = O\left(N^{1+\frac{1}{n+1}}\left(\frac{\log N}{\log\log N}\right)^{\frac{n+2}{2}-\frac{1}{n+1}}\right)$$

To achieve this bound choose:

$$m' = O\left(m\frac{\log k}{\log\log k}\right) \qquad \text{or} \qquad m' > (m-1)(\log_2 k + 2)$$

and at level i choose

$$k = \frac{N^{\frac{i}{i+1}}}{(\log N / \log\log N)^{\frac{i}{2}-\frac{1}{i+1}}}.$$

Derivation of these equations are not covered.

(These equations are not easy to work with.)

Routing Performance

A new request can be routed in $O(mk)$ time.

A connection assignment can be routed in $O(m^2k)$ time.

(Routing for this network will not be covered.)

Fanout/Concentrate Generalized Connector

Described by Nassimi and Sahni in a 1982 paper.

Construction:

Consists of three stages:

- *Generalizer.* Makes 2^m copies of each input.
- *Concentrator.* Multiplexes (concentrates) inputs (to outputs).
- *Generalized Connector.* An $N2^{-m}$ generalized connector.

Operation:

Generalizer makes a copy for each of M concentrators.

Concentrator routes active inputs to recursive GC.

Recursive generalized connector completes routing.