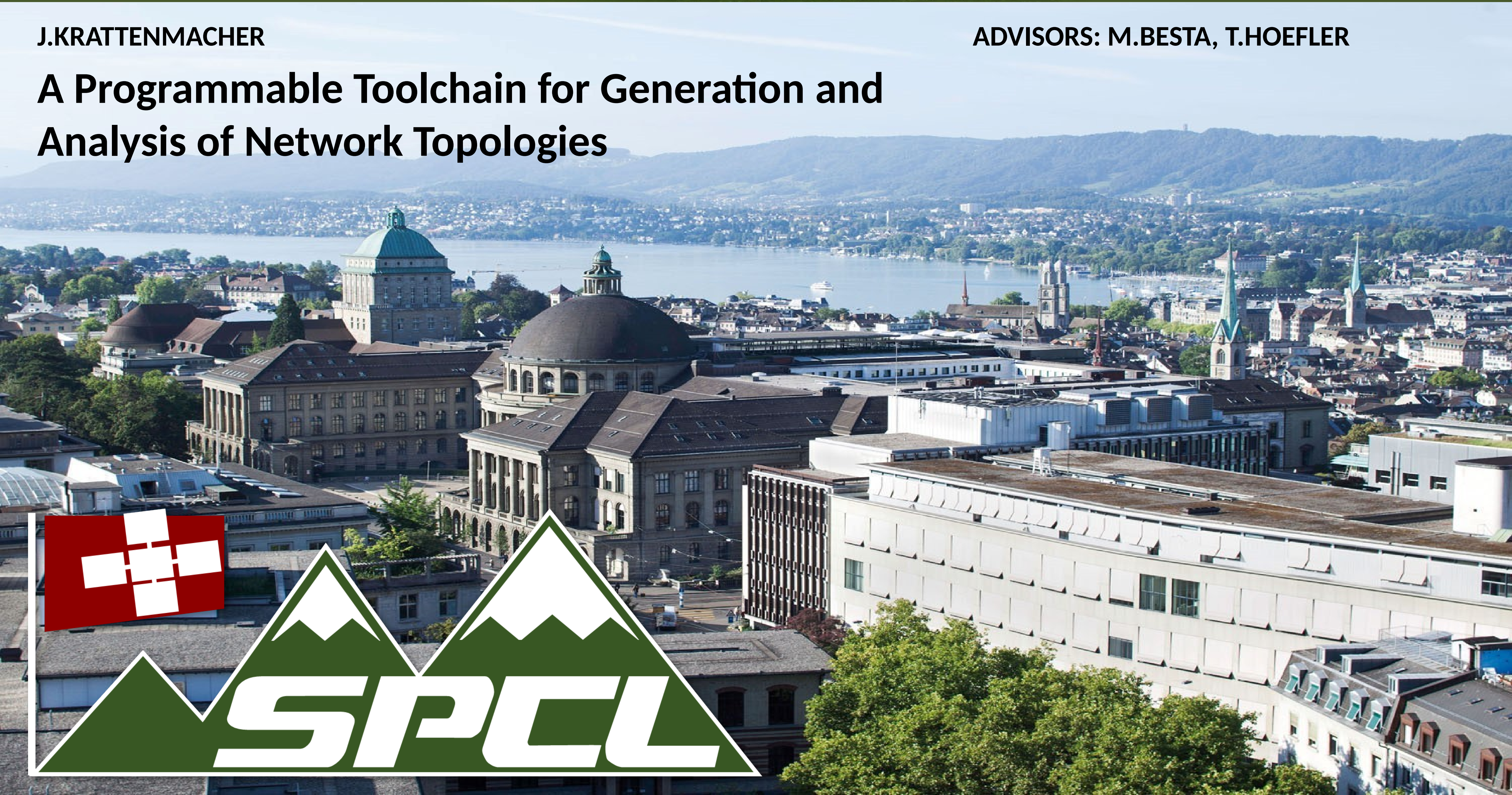
J.KRATTENMACHER

ADVISORS: M.BESTA, T.HOEFLER

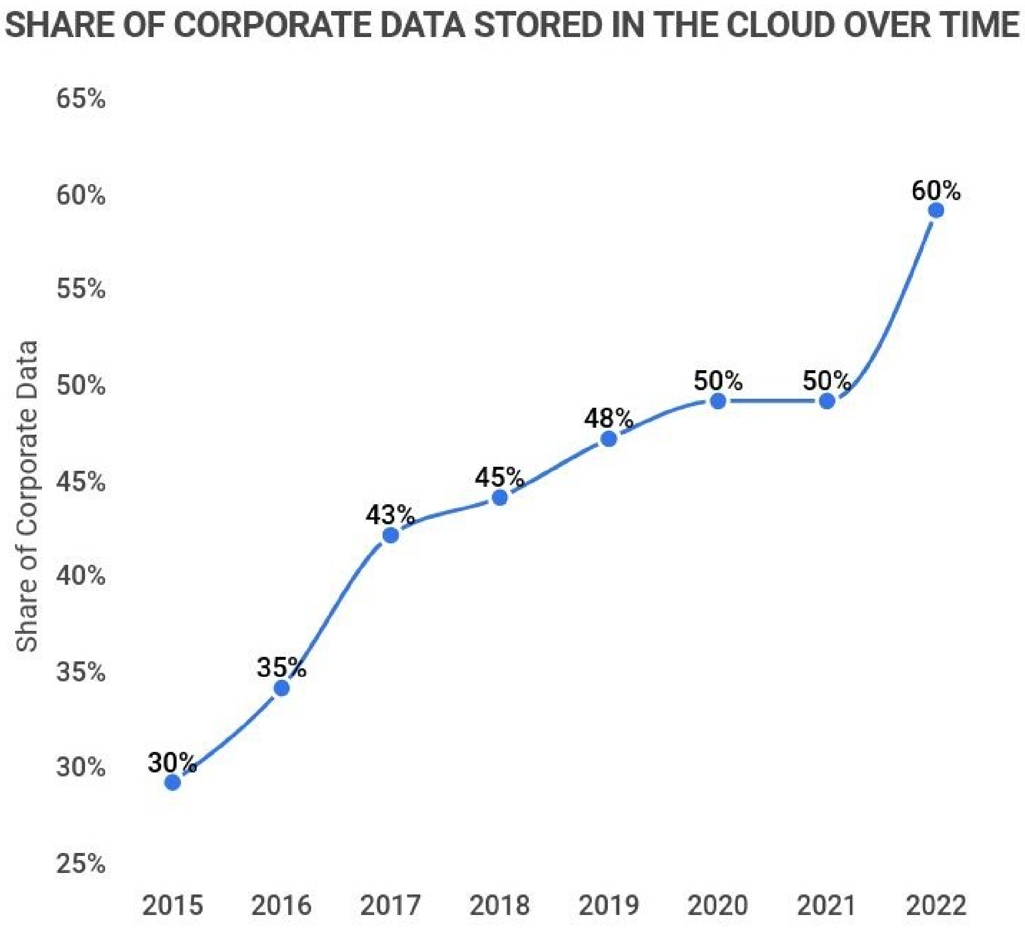
A Programmable Toolchain for Generation and Analysis of Network Topologies



Motivation

With the growth of cloud computing and large-scale computing, having fast and reliable data centers are more important than ever.

A high-performant network topology is key for high performance.



Source: https://www.zippia.com/advice/cloud-adoption-statistics/

Motivation

Traditionally people were using fat trees to reach high performance.
Fat trees have multiple shortest paths between any nodes.

Newer low diameter networks like Slimfly and Dragonfly have been shown to be more efficient and cost effective. As an example, Slimfly has ~2x lower latency and ~15% higher throughput compared to similar cost fat trees [1].

For these networks to be performant, we need multipathing (especially over non minimal paths) . Therefore, it makes it harder to generate routing strategies.

What does it mean to have multiple (non minimal) paths?

[1] Fatpaths: Routing in supercomputers and data centers when shortest paths fall short. M. Besta, M. Schneider, K. Cynk, M. Konieczny, E. Henriksson, S. Di Girolamo, A. Singla, and T. Hoefler

Motivation

Traditionally people were using fat trees to reach high performance.
Fat trees have

Newer low di wn to be
more efficient latency and
~15% higher

For these net over non
minimal paths gies.

What does it mean to have multiple (non minimal) paths?

We need to understand the path diversity of a network before we start developing routing protocols and before we start doing simulations

[1] Fatpaths: Routing in supercomputers and data centers when shortest paths fall short. M. Besta, M. Schneider, K. Cynk, M. Konieczny, E. Henriksson, S. Di Girolamo, A. Singla, and T. Hoefler

Goal

Create a Toolchain that is:

Goal

Create a Toolchain that is:

Variety of Networks

Goal

Create a Toolchain that is:

Variety of Networks

Analyze different path Diversity Properties

Goal

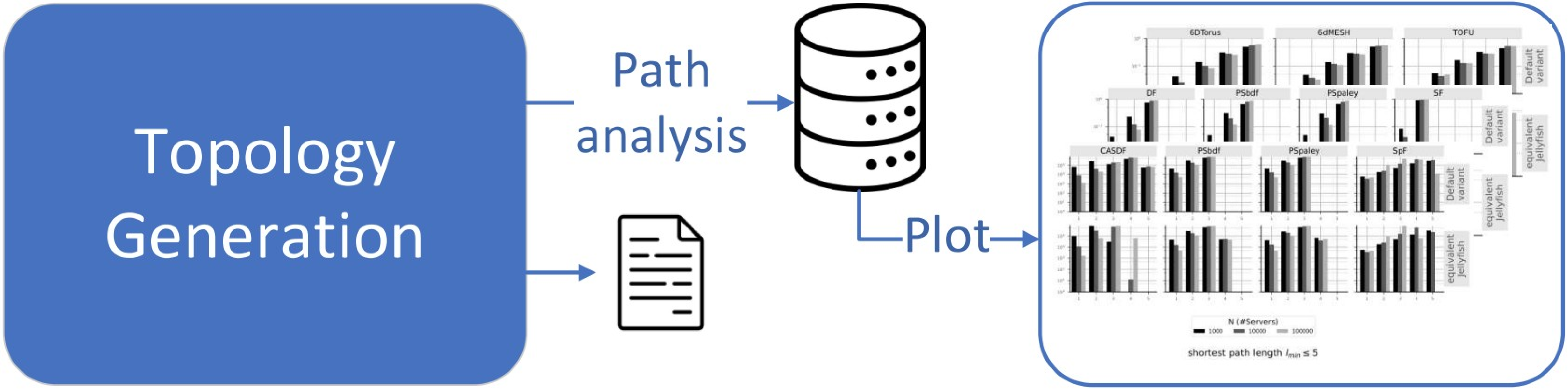
Create a Toolchain that is:

Variety of Networks

Analyze different path Diversity Properties

User-friendly, performant & ensure extensibility

Toolchain Overview

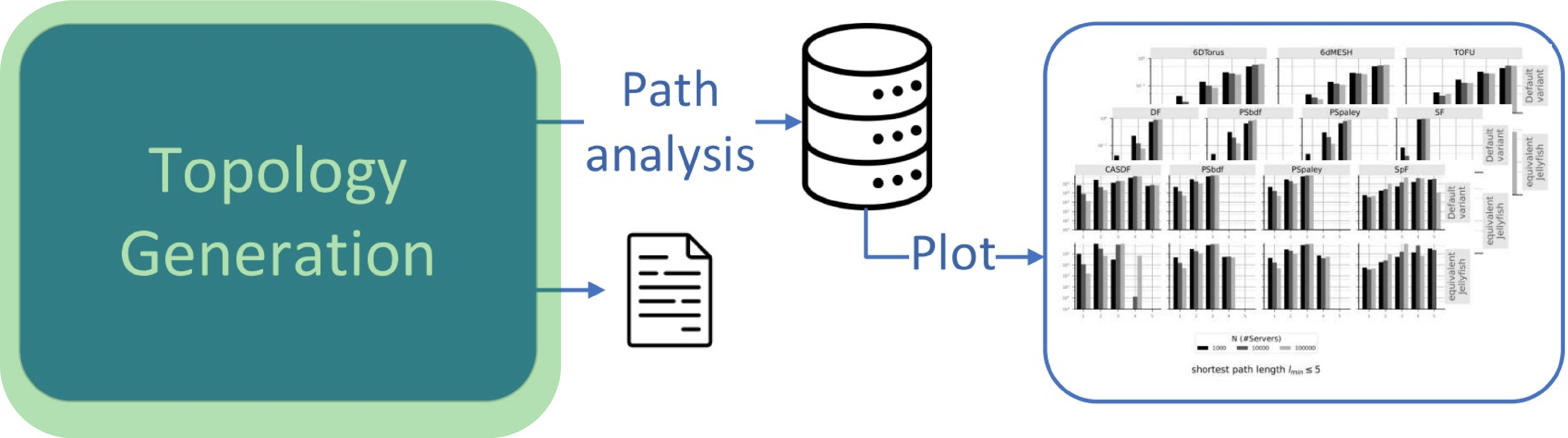


Topology Generation

Path analysis

Plot

Toolchain Overview



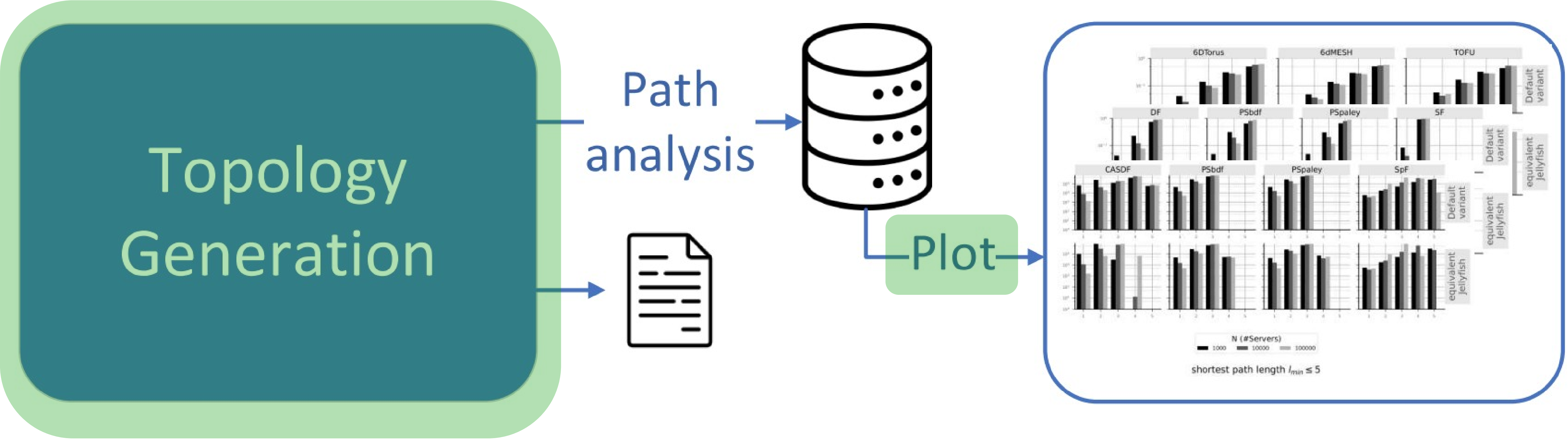
Topology Generation

Path analysis

Plot

- Adding multiple new topologies
- Make it expandable

Toolchain Overview



- Adding multiple new topologies
- Make it expandable

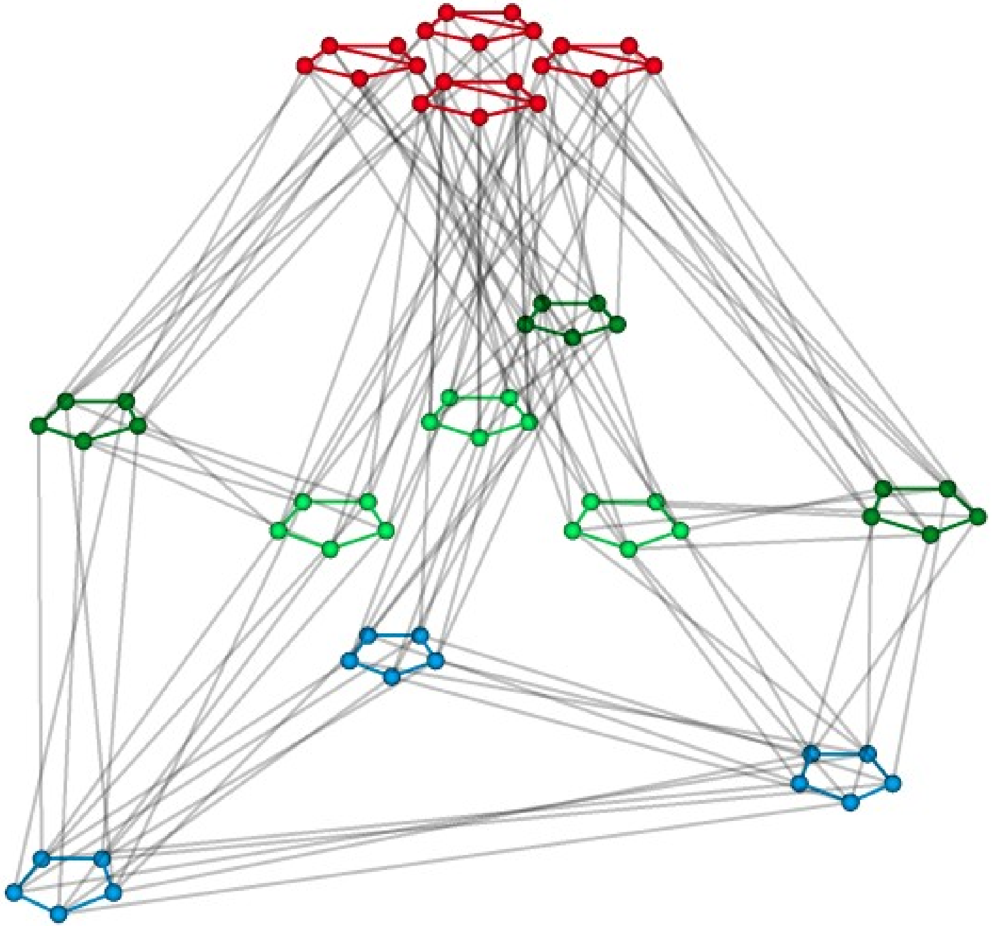
Visualization Module

- Increased productivity/performance
- Better visualization

Topologies

Modern low diameter Networks

- Slimfly
- Polarfly
- Expander
- Polarstar
- Megafly
- Spectralfly
- Dragonfly
- Cascade Dragonfly
- Random (Jellyfish)



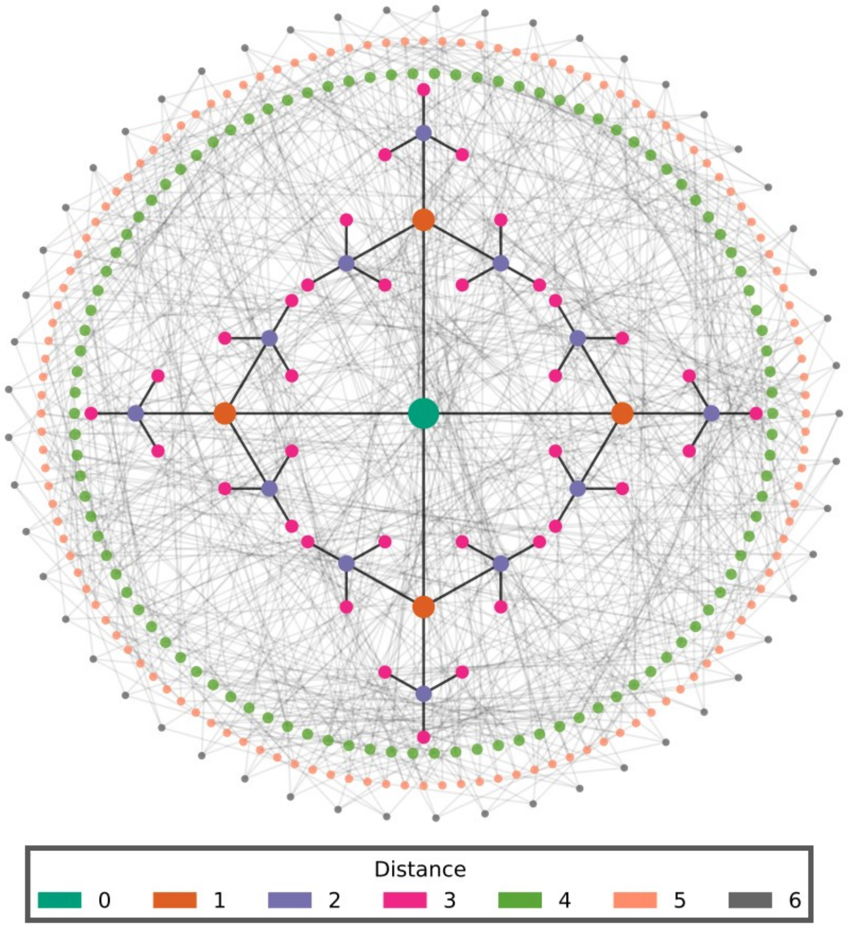
$G{*}G' : ER_3 * Paley(5)$

Source: PolarStar: Expanding the Scalability Horizon of Diameter-3 Networks. K. Lakhotia, L. Monroe, K. Isham, M. Besta, N. Blach, T. Hoefler, F. Petrini

Topologies

Modern low diameter Networks

- Slimfly
- Polarfly
- Expander
- Polarstar
- Megafly
- Spectralfly
- Dragonfly
- Cascade Dragonfly
- Random (Jellyfish)



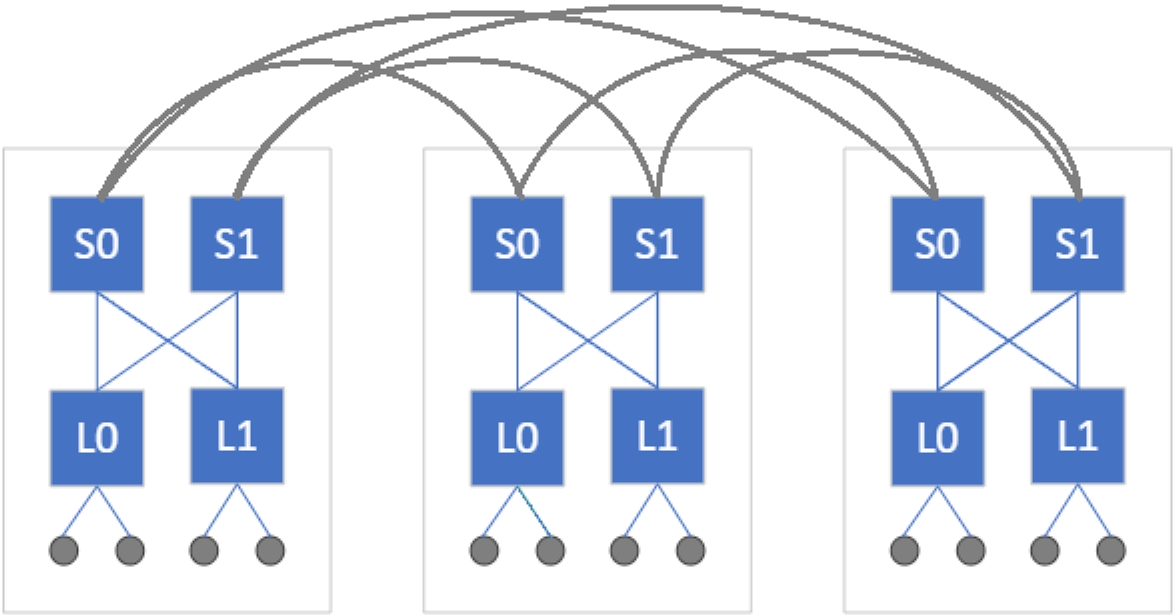
$\mathrm{SpF}_{3,7}$

Source: SpectralFly: Ramanujan Graphs as Flexible and Efficient Interconnection Networks. S. Young, S. Aksoy, J. Firoz, R Gioiosa, T. Hagge, M. Kempton, J. Escobedo, M. Raugas

Topologies

Modern low diameter Networks

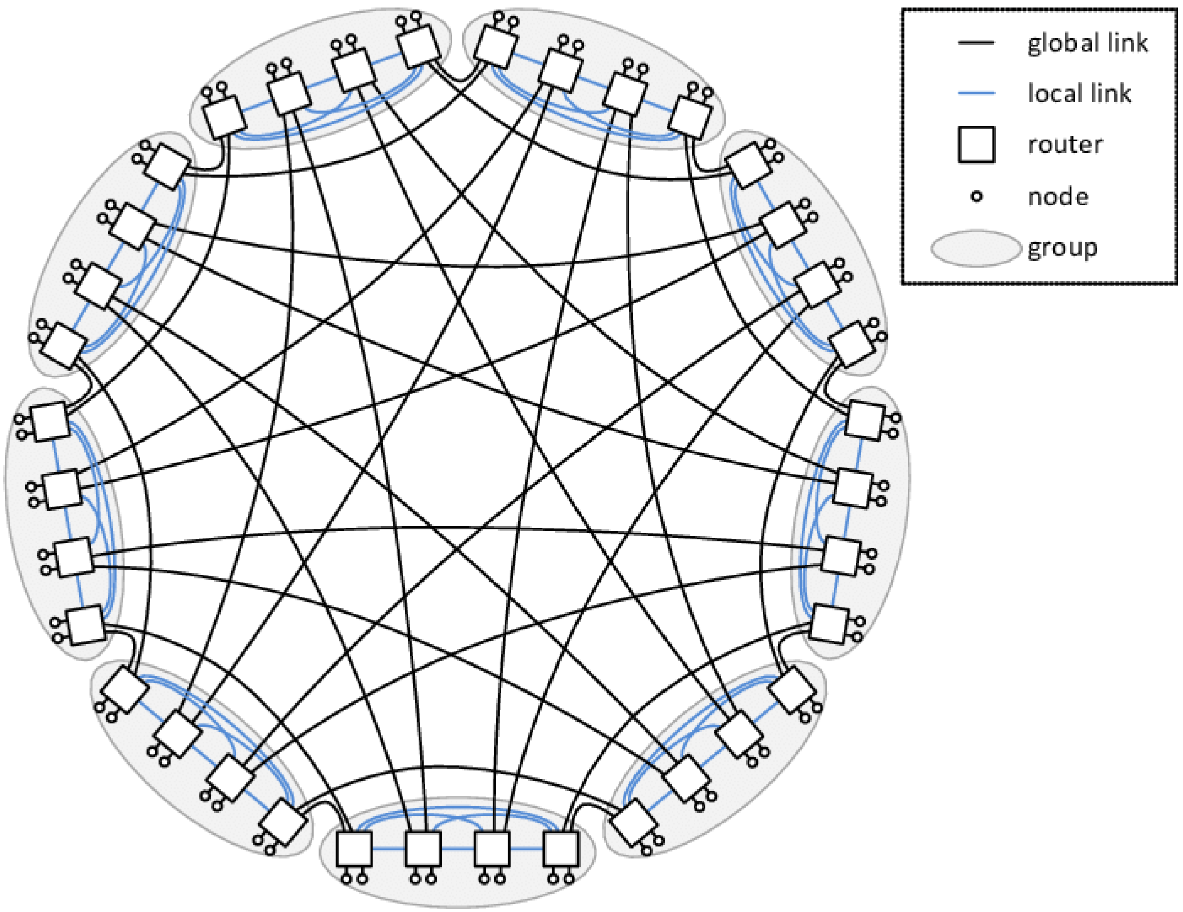
- Slimfly
- Polarfly
- Expander
- Polarstar
- Megafly
- Spectralfly
- Dragonfly
- Cascade Dragonfly
- Random (Jellyfish)



Topologies

Modern low diameter Networks

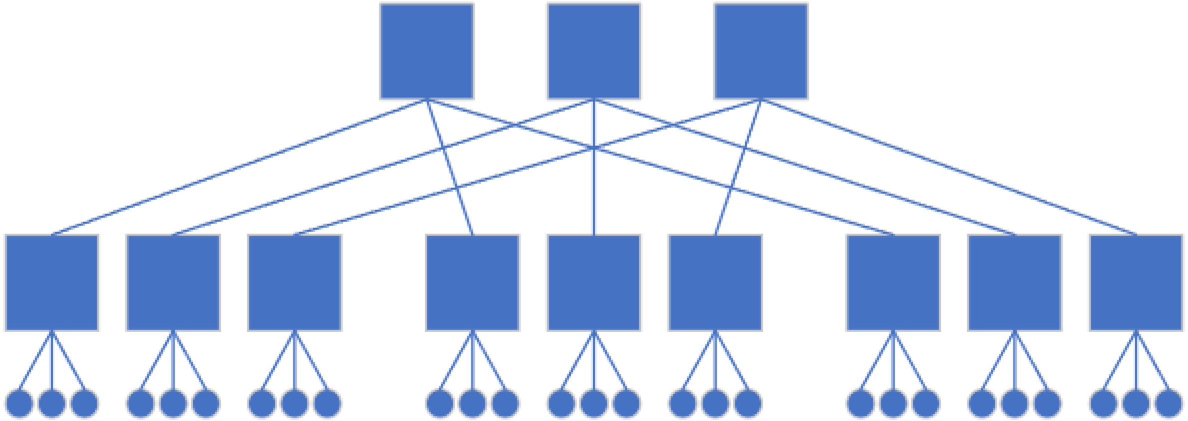
- Slimfly
- Polarfly
- Expander
- Polarstar
- Megafly
- Spectralfly
- Dragonfly
- Cascade Dragonfly
- Random (Jellyfish)



Dragonfly

Source: On-the-Fly Adaptive Routing in High-Radix Hierarchical Networks. M. Garcia, E. Vallejo, R. Beivide, M. Odriozola, C. Camarero, M. Valero, G. Rodriguez, J. Labarta, C. Minkenberg

Topologies



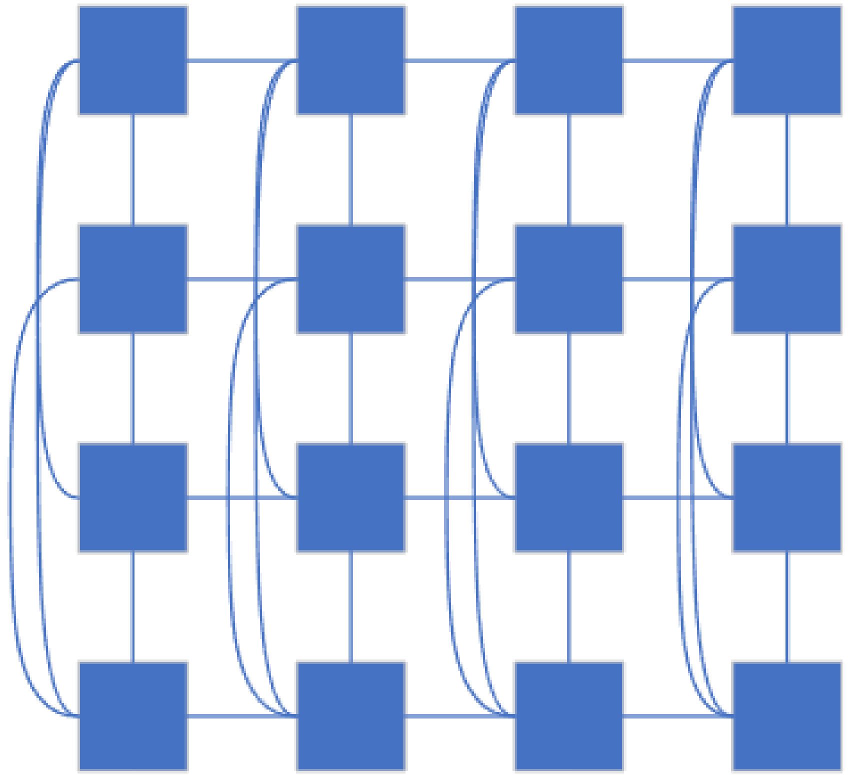
Tree Networks:

- Fat tree
- Fat tree2x
- K-ary n-tree
- eXtended Generalized Fat Trees (XGFT)
- Multi-Layer-Full-Mesh (MLFM)

Topologies

Mesh/Torus variants

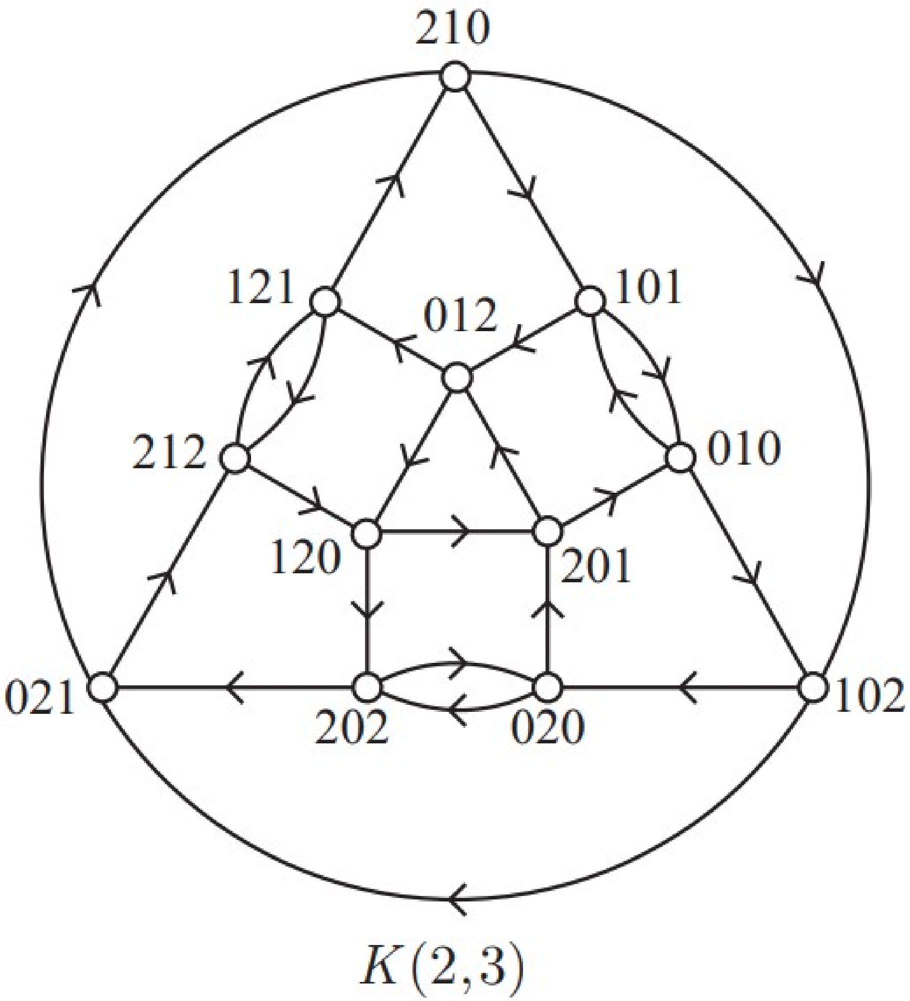
- Mesh
- Express Mesh
- Torus
- Tofu
- Hypercube
- HyperX
- Flattened Butterfly



Topologies

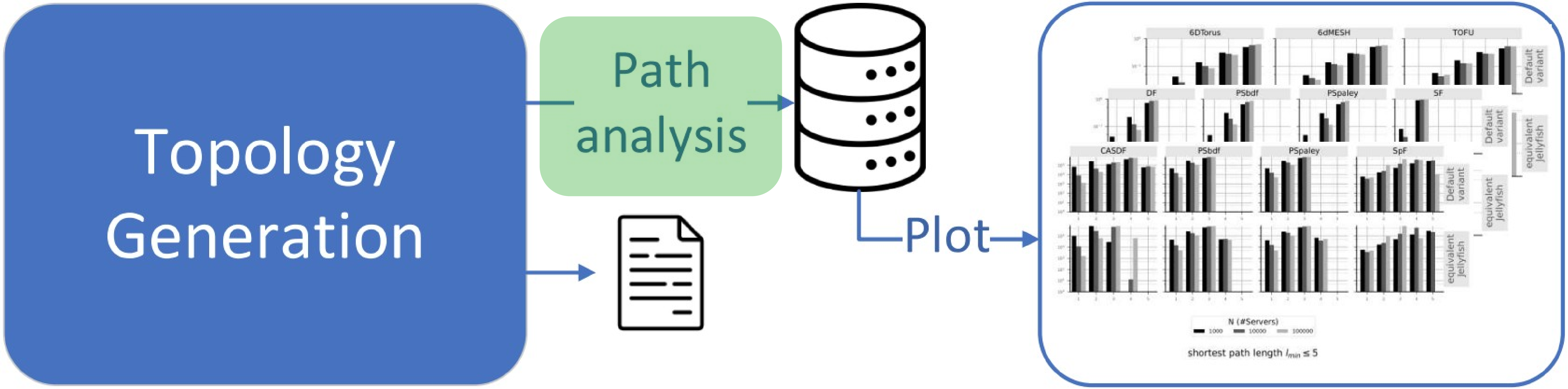
Kautz Graph:

- Kautz
- Arrangement graph



Source: The k-tuple twin domination in de Bruijn and Kautz digraphs. Toru Araki

Path Analysis



Topology Generation

Path analysis

Plot

Path Analysis – Shortest paths and multiplicity

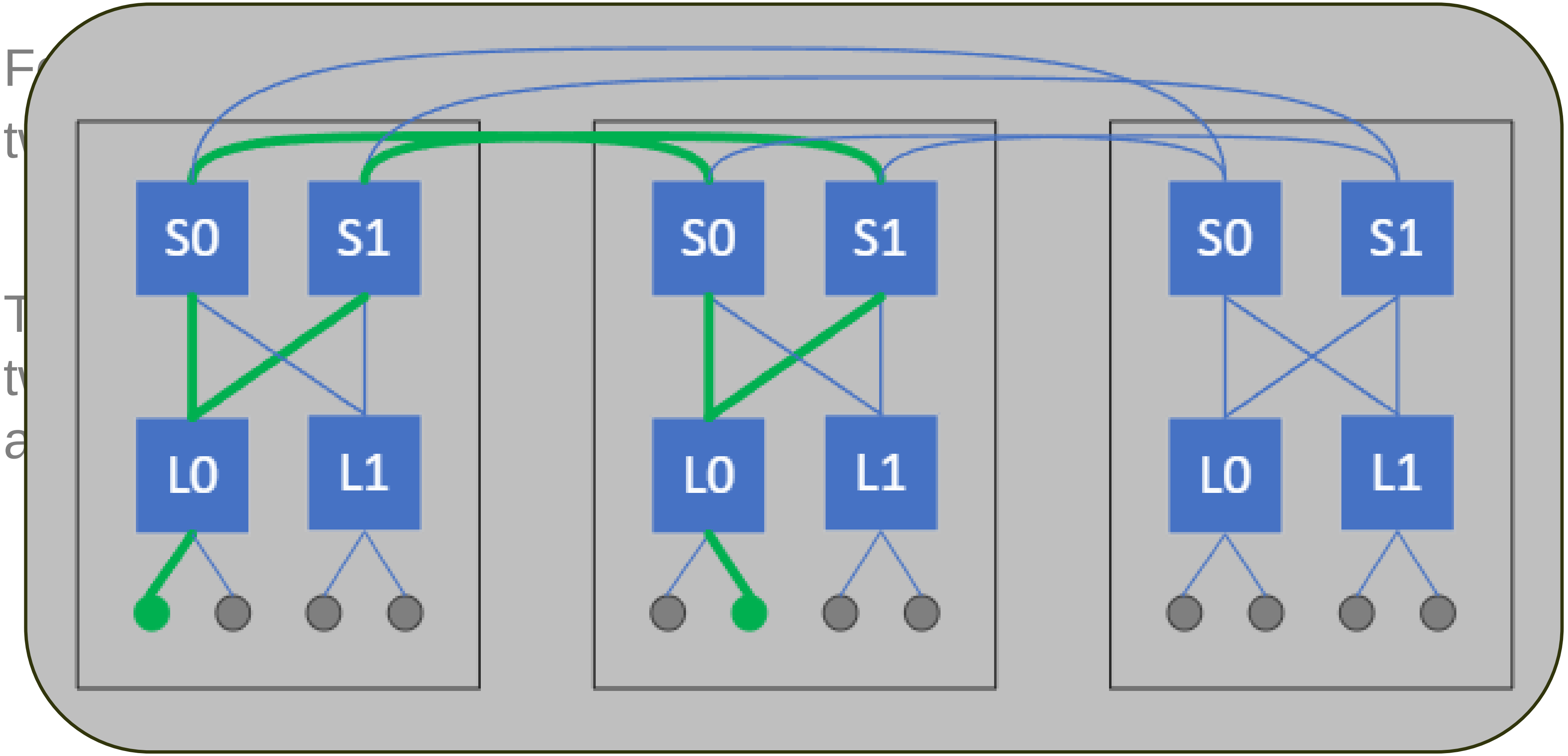
For s,t $\in$ V the length of the shortest path $l_{min}(s,t)$ connecting the two nodes is defined *as* $l_{min}(s,t) = min \{ i \in N : t \in h^i (\{s\})\}$

The shortest path multiplicity (or count of shortest paths) between two nodes s,t $\in$ V counts the number of shortest paths between s and t and can be defined as $n_{min}(s, t) = n_l(s, t)$ *with* $l = l_{min}(s,t)$

Path Analysis – Shortest paths and multiplicity

F
t

T
t
a

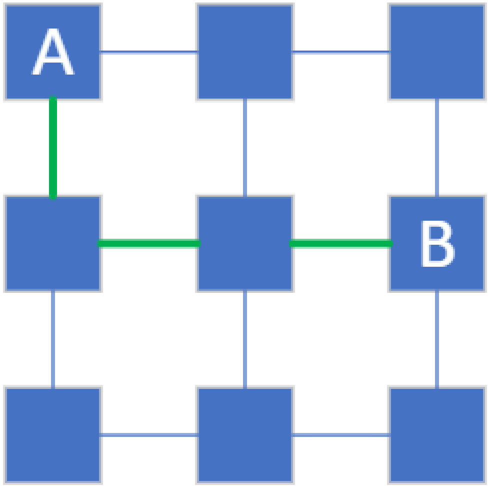


Path Analysis – Edge disjoint paths

Edge disjoint path is the smallest number of edges that can be removed between two sets of nodes, so that there is no longer a path between them of length l, denoted $c_l(A, B)$.

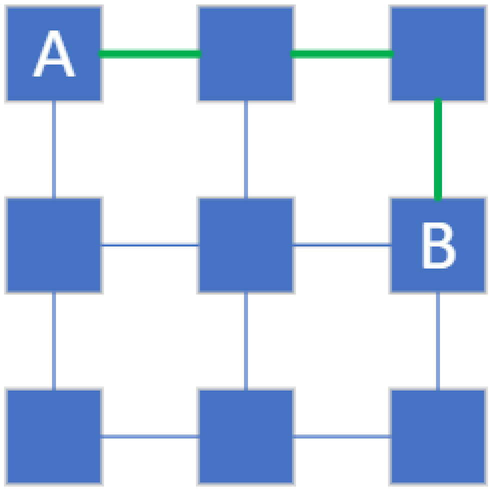
Path Analysis – Edge disjoint paths

Edge disjoint path is the smallest number of edges that can be removed between two sets of nodes, so that there is no longer a path between them of length l, denoted $c_l(A, B)$.



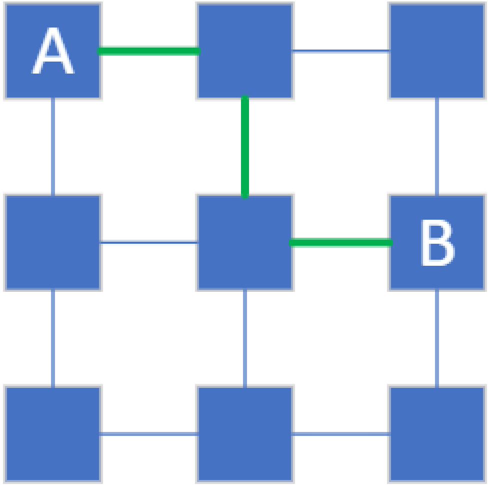
Path Analysis – Edge disjoint paths

Edge disjoint path is the smallest number of edges that can be removed between two sets of nodes, so that there is no longer a path between them of length l, denoted $c_l(A, B)$.



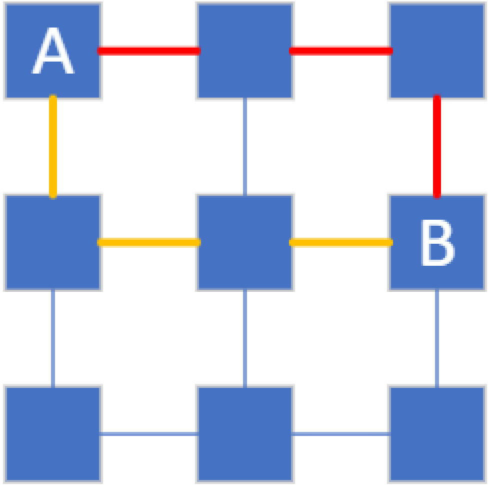
Path Analysis – Edge disjoint paths

Edge disjoint path is the smallest number of edges that can be removed between two sets of nodes, so that there is no longer a path between them of length l, denoted $c_l(A, B)$.



Path Analysis – Edge disjoint paths

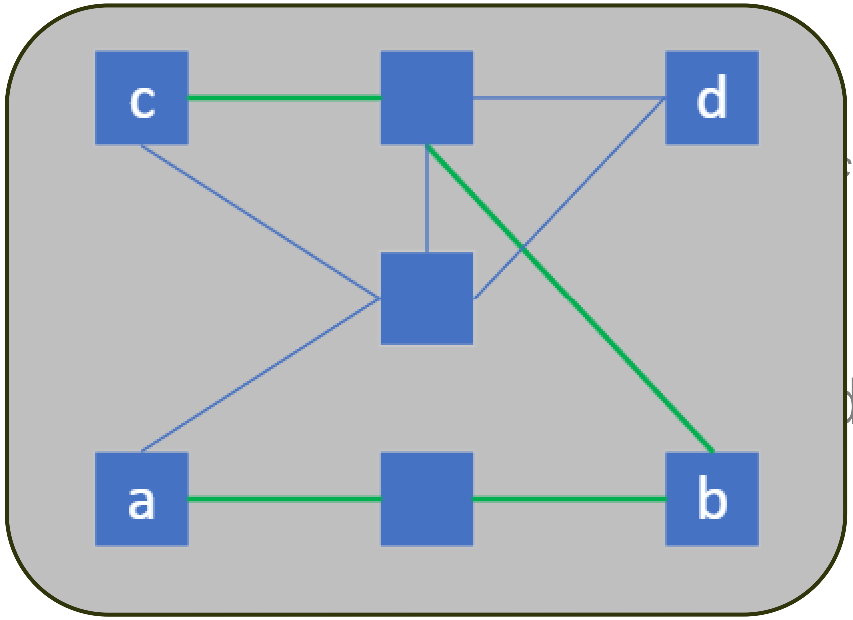
Edge disjoint path is the smallest number of edges that can be removed between two sets of nodes, so that there is no longer a path between them of length l, denoted c_l(A, B).



Path Analysis – Interference

Interference $I^{l}_{ab,cd}$ is defined as $c_l(\{a,c\}, \{b\}) + c_l(\{a,c\}, \{d\}) - c_l(\{a,c\}, \{b,d\})$, with $\mathrm{c_l}(\{\mathrm{s}\}, \{\mathrm{t}\})$ = edge disjoint paths between two nodes s and t of length l.

Path Analysis – Interference

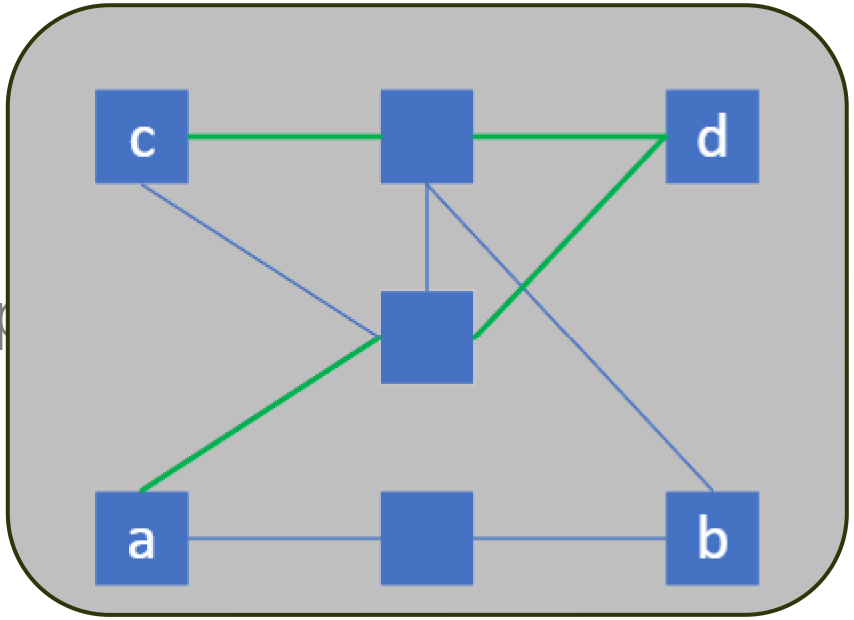


$_{d}$ is defined as

({a,c}, {b,d}),

edge disjoint

t of length *l*.



C_2({a,c},{b}) = 2

C_2({a,c},{d}) = 2

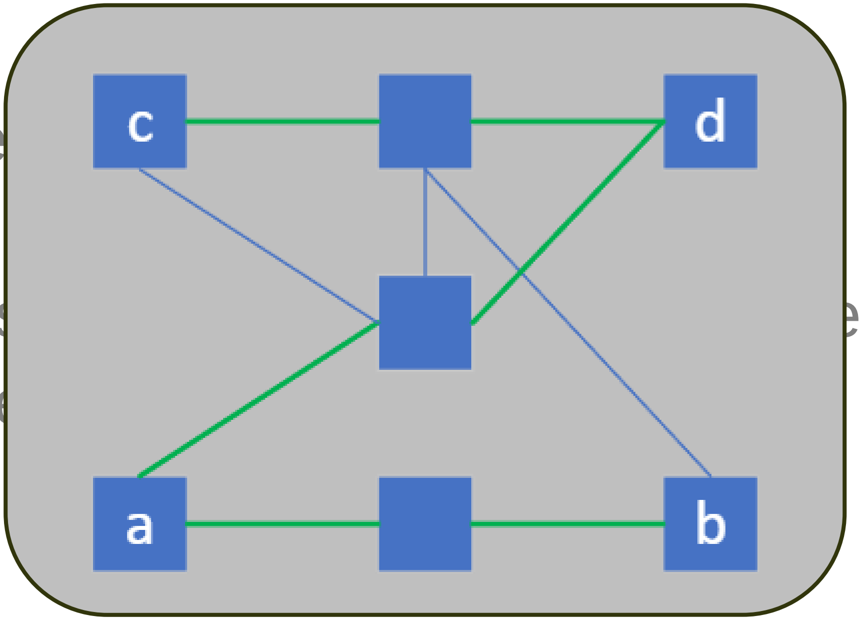
Path Analysis – Interference

Interfere *{b})* +

c_l({a,c},

with c_l ({s etween

two node



C_2({a,c},{b,d}) = 3

Path Analysis – Connectivity

For a given length *l*, connectivity is defined as

$$\frac{c_l(\{s\},\{t\})}{r'}$$

where *r'* is the network radix.

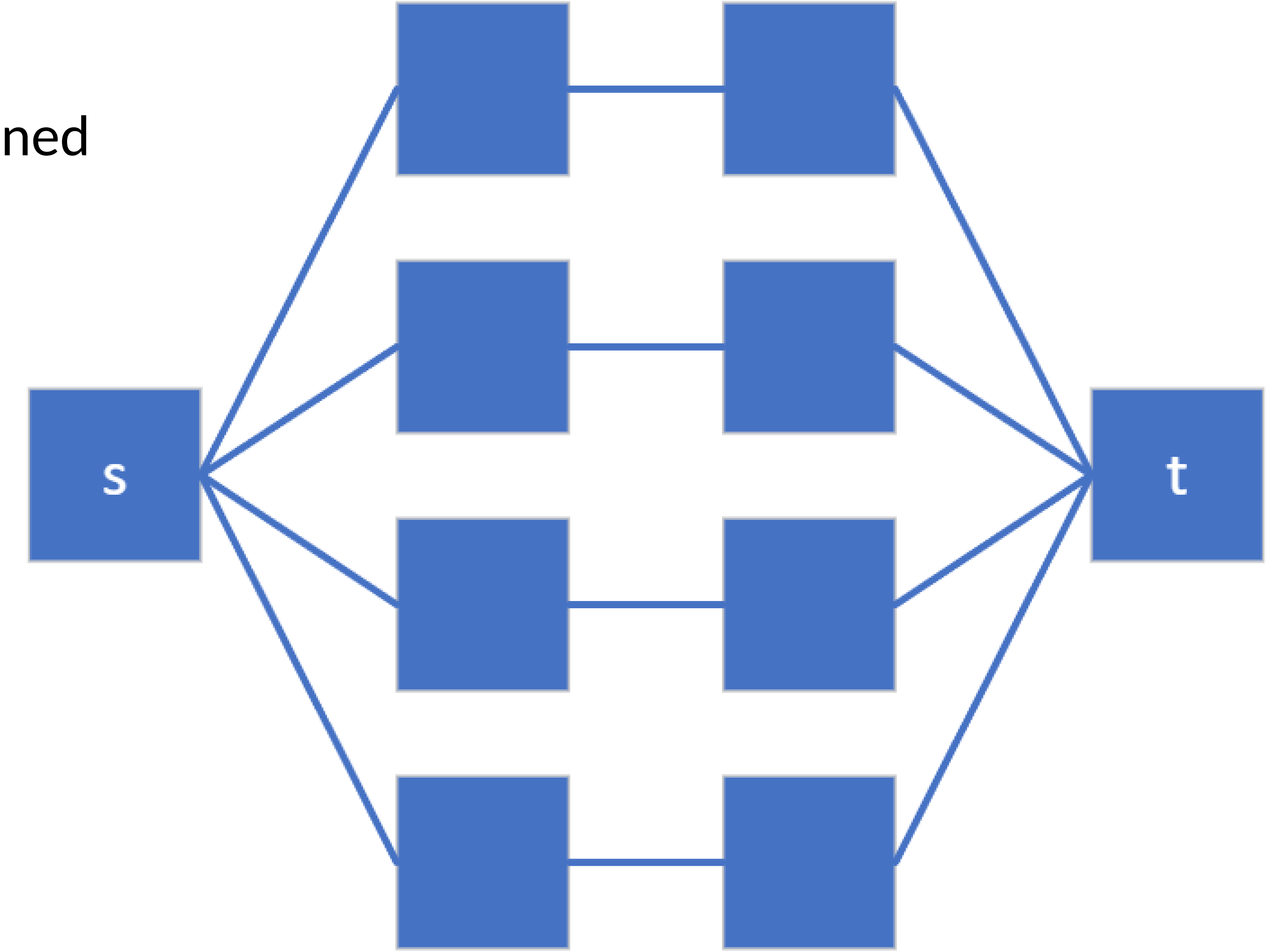
Path Analysis – Connectivity

For a given length *l*, connectivity is defined as

$$\frac{c_l(\{s\},\{t\})}{r'}$$

where *r'* is the network radix.

$$\frac{c_l(\{s\},\{t\})}{r'}=\frac{4}{4}=100\,\%$$



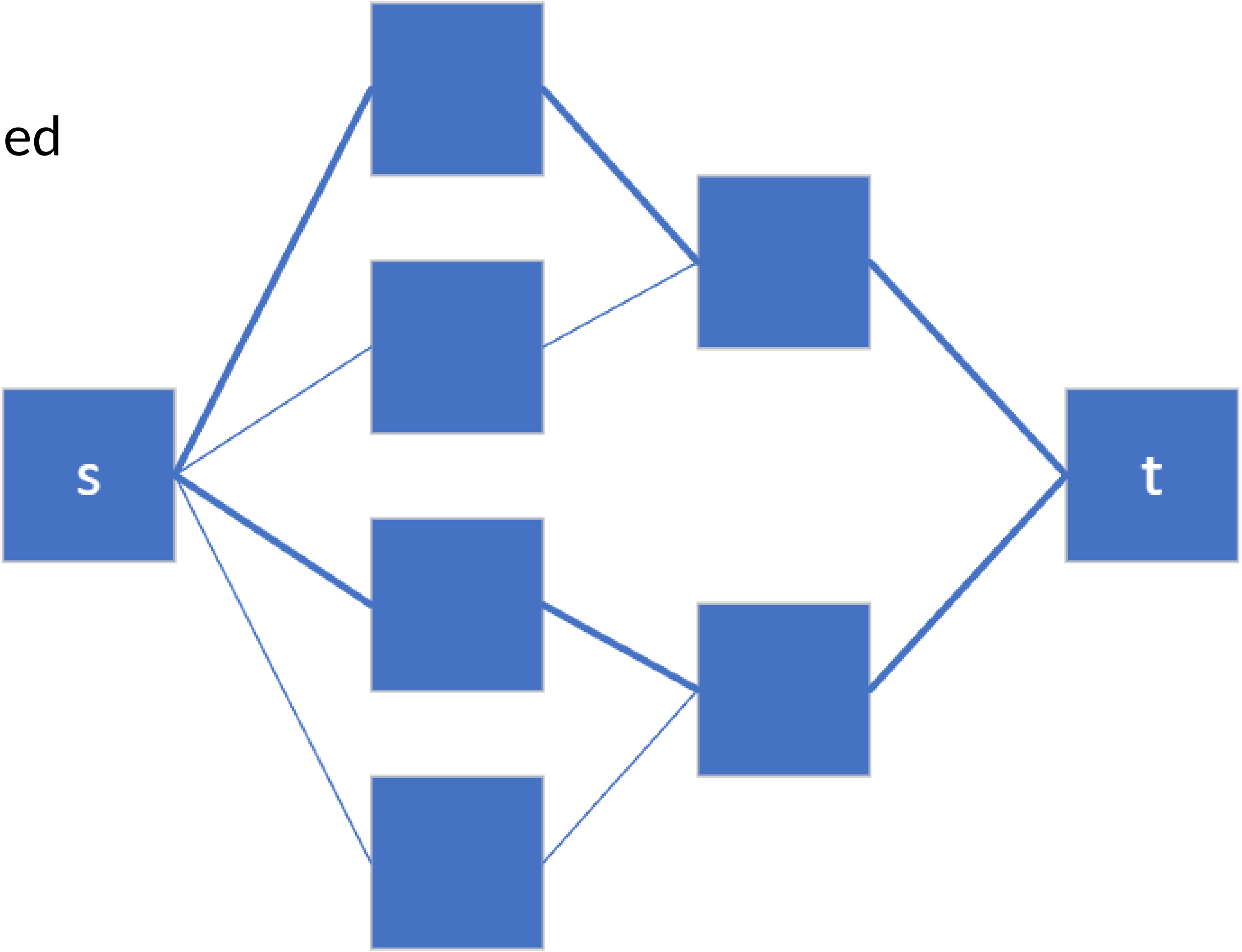
Path Analysis – Connectivity

For a given length *l*, connectivity is defined as

$$\frac{c_l(\{s\},\{t\})}{r'}$$

where *r'* is the network radix.

$$\frac{c_l(\{s\},\{t\})}{r'}=\frac{2}{4}=50\,\%$$



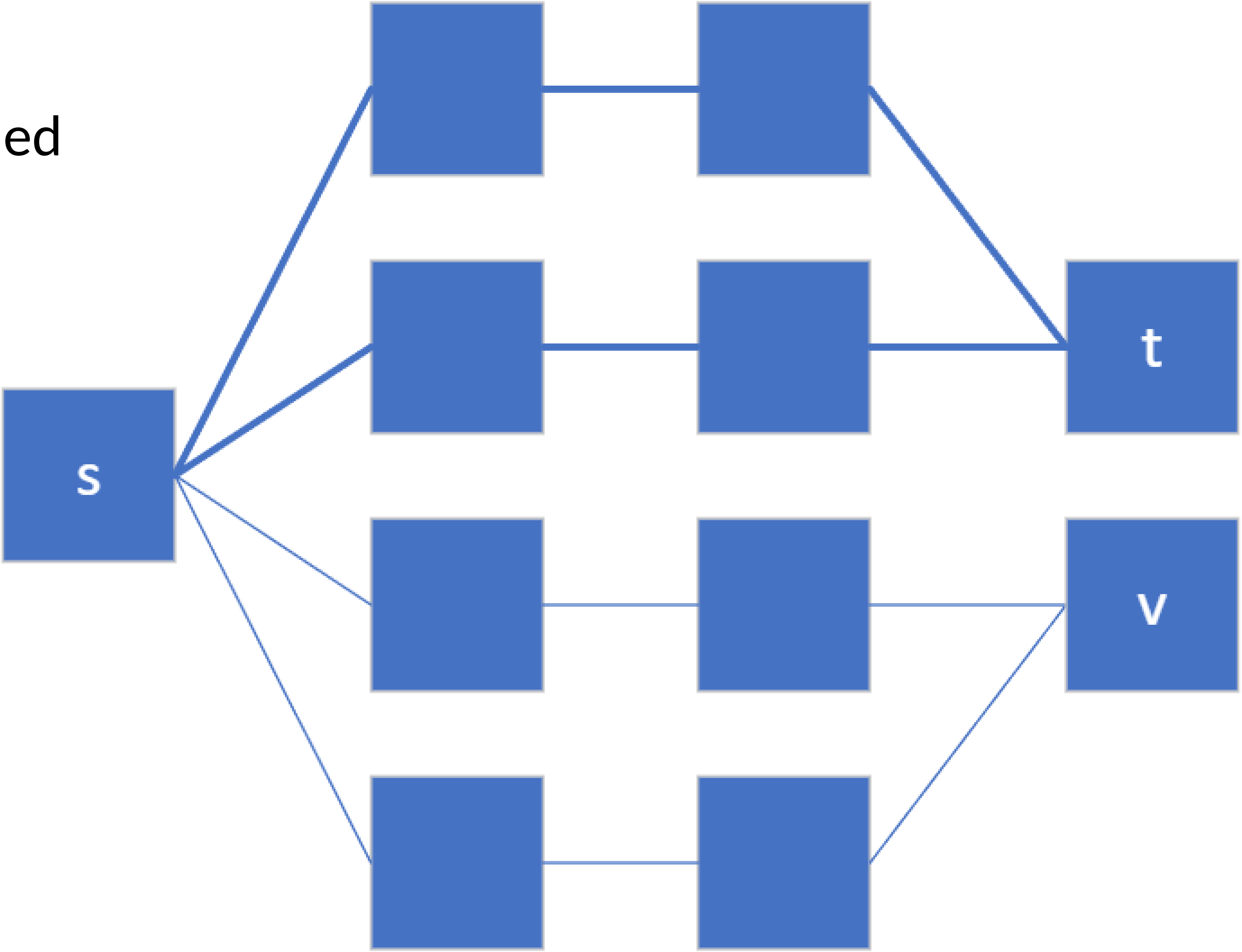
Path Analysis – Connectivity

For a given length *l*, connectivity is defined as

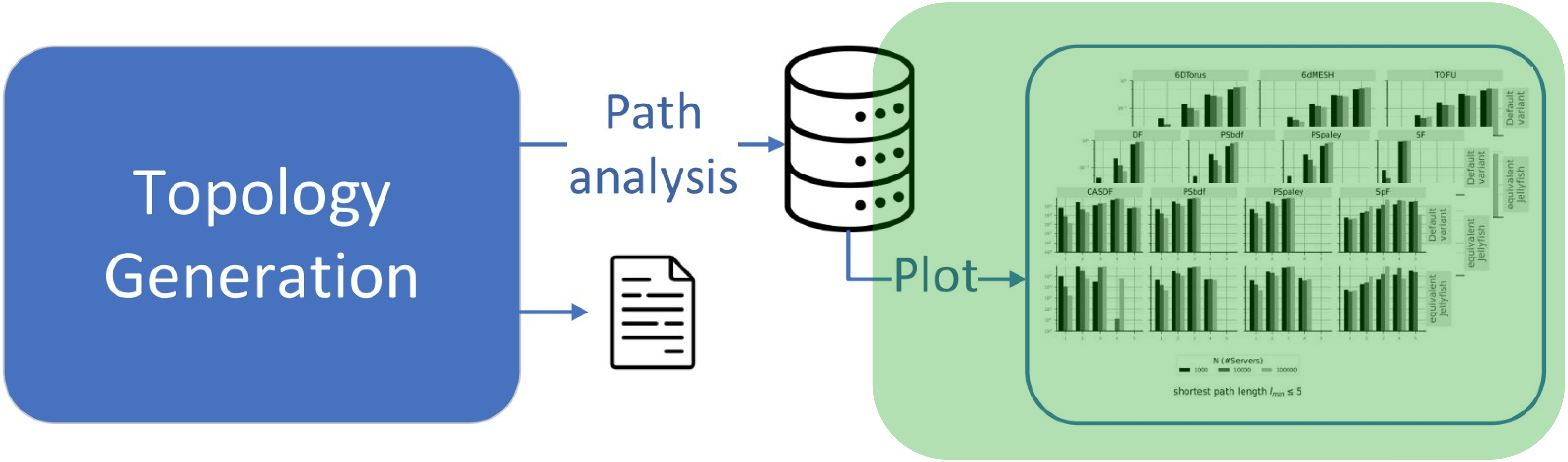
$$\frac{c_l(\{s\},\{t\})}{r'}$$

where *r’* is the network radix.

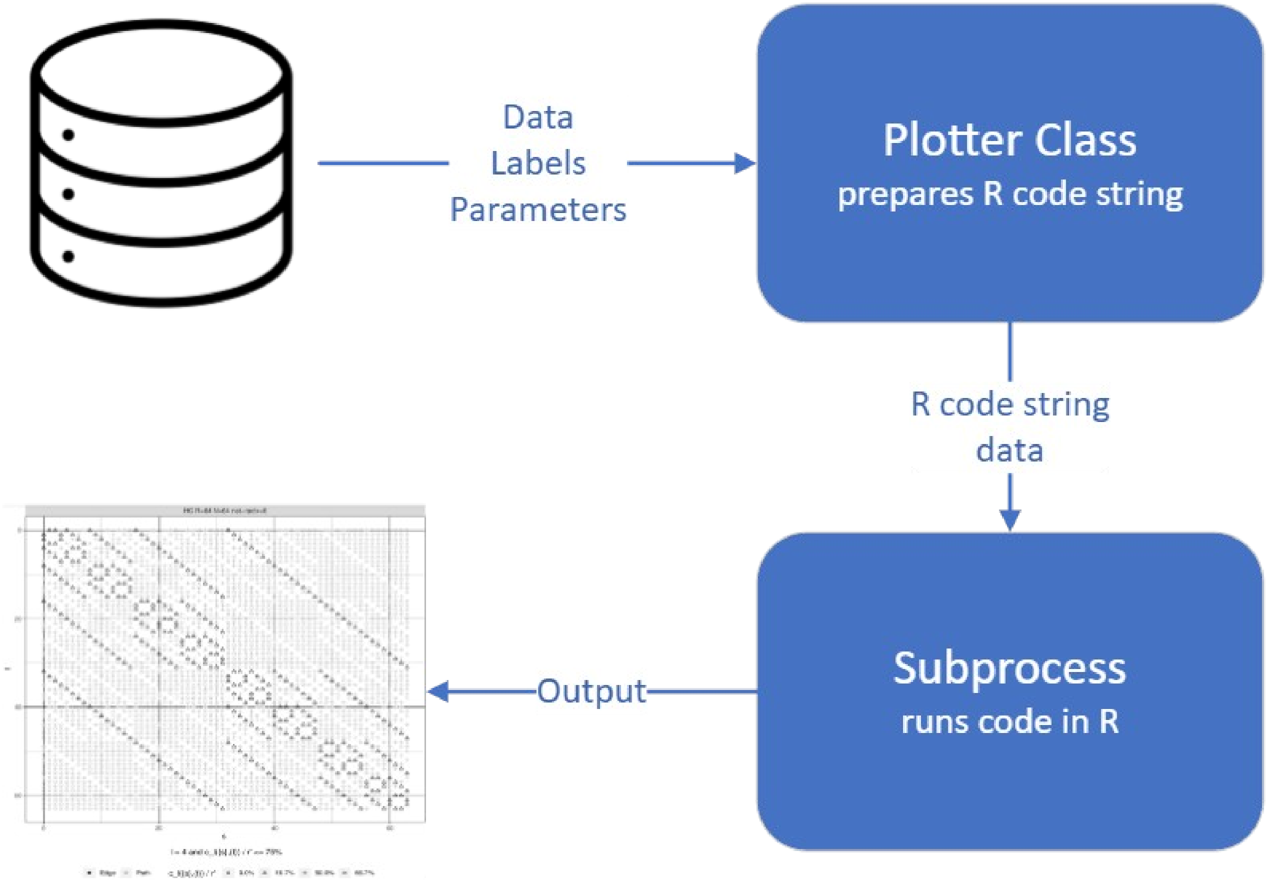
$$\frac{c_l(\{s\},\{t\})}{r'}=\frac{2}{4}=50\,\%$$



The Visualization Module



The Visualization Module

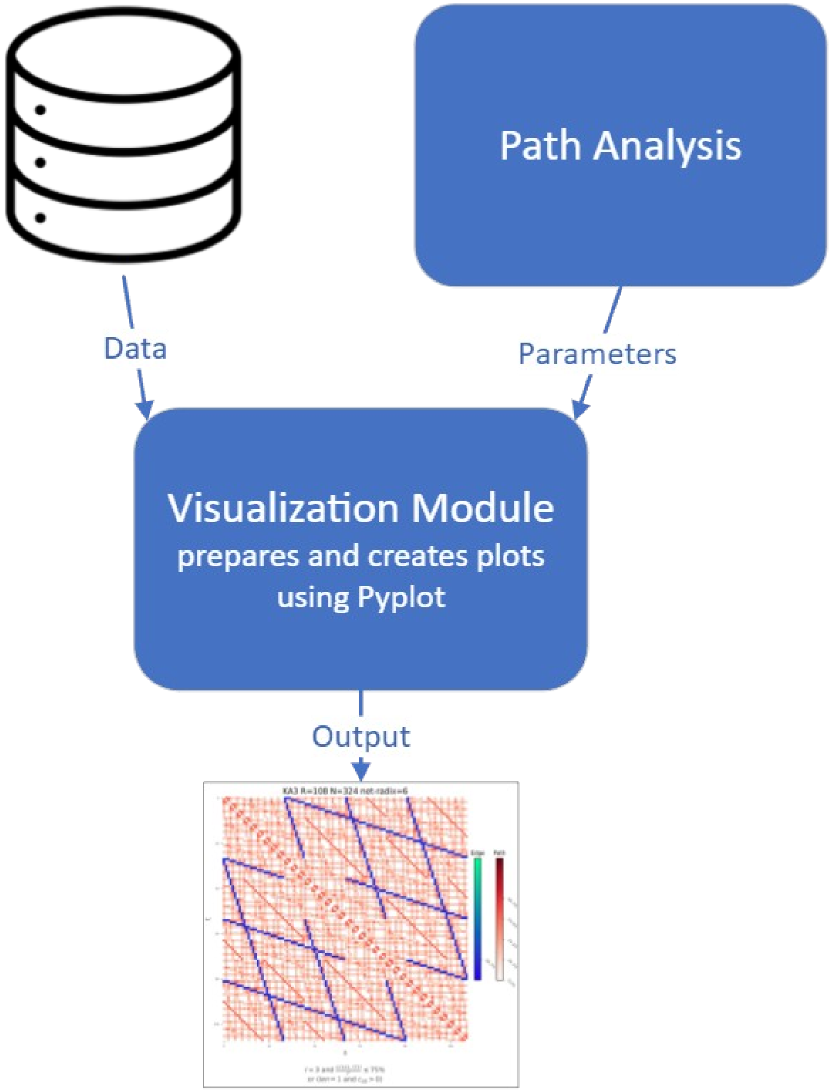


Plotter function of previous toolchain

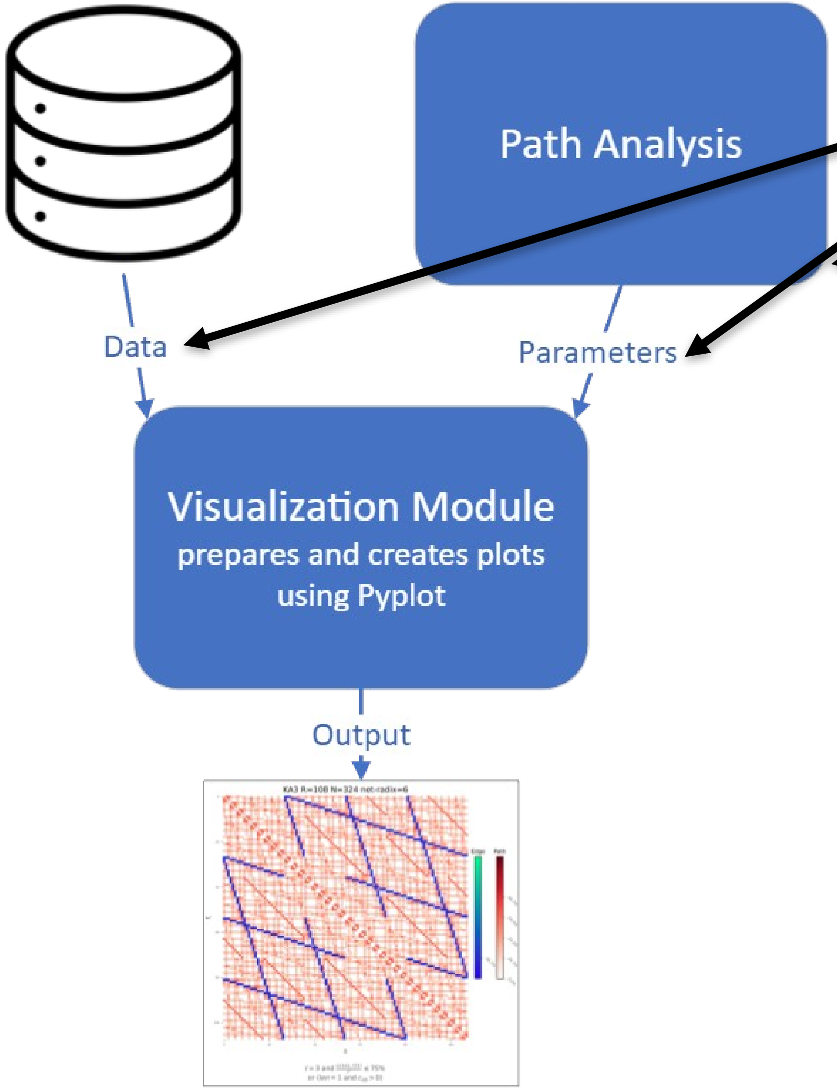
Issues:

- Parameters for design are coupled with data
- Difficult to expand current plotting tools
- No way to interact with data R code
- Hard to debug

The Visualization Module



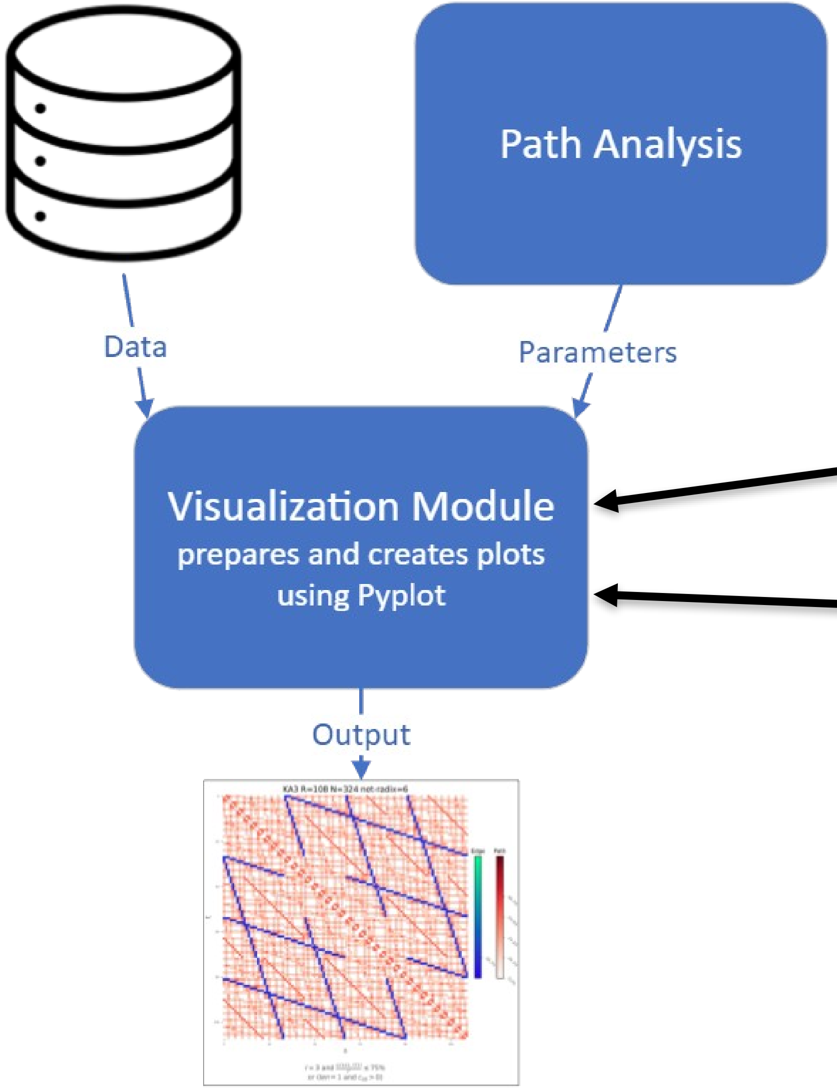
The Visualization Module


Path Analysis
Data
Parameters
Visualization Module
prepares and creates plots using Pyplot
Output
KA3 R=108 N=324 net-radix=6

Clear distinction between data and parameters/design elements

The same (and more) design parameters are given by the specific Analysis class

The Visualization Module


Path Analysis
Data
Parameters
Visualization Module
prepares and creates plots using Pyplot
Output
KA3 R=108 N=324 net-radix=6
Edge
Path

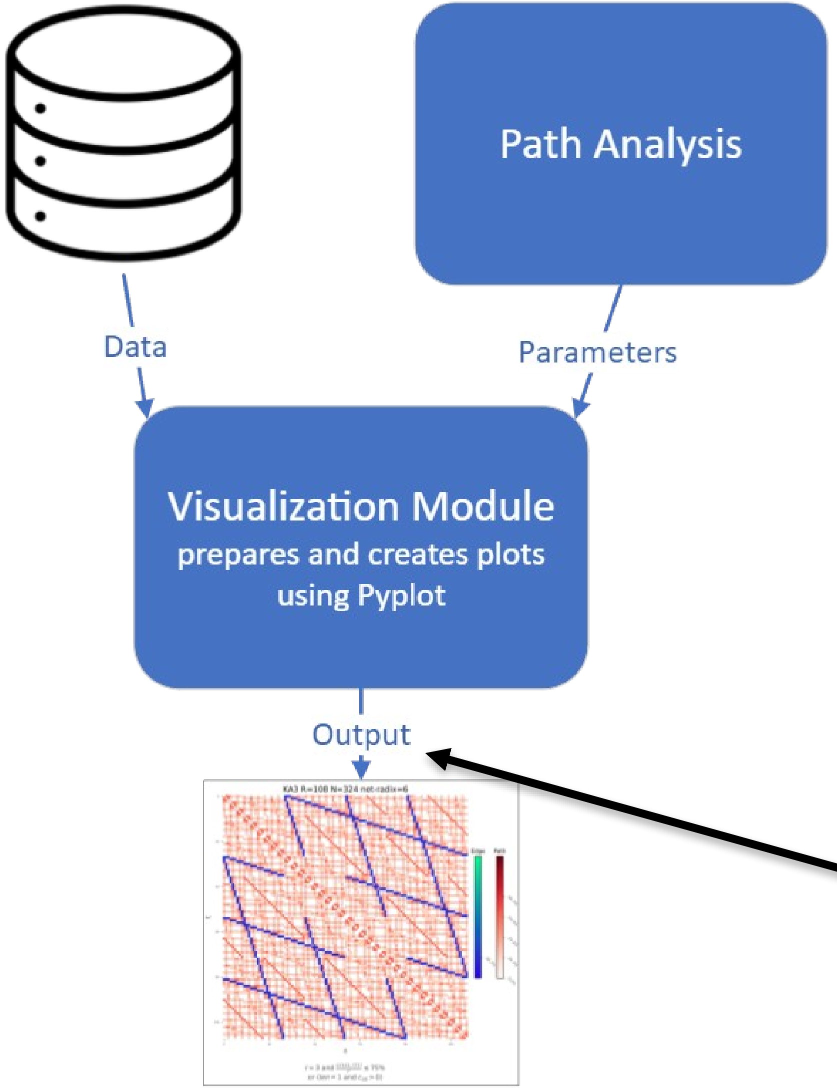
Clear distinction between data and parameters/design elements

The same (and more) design parameters are given by the specific Analysis class

Some design elements for all types of analysis (directly coded in the Plotter class)

Easily readable code (each analysis type is handled separately)

The Visualization Module


Path Analysis
Data
Parameters
Visualization Module
prepares and creates plots using Pyplot
Output
KA3 R=108 N=324 net-radix=6
Edge
Path

Clear distinction between data and parameters/design elements

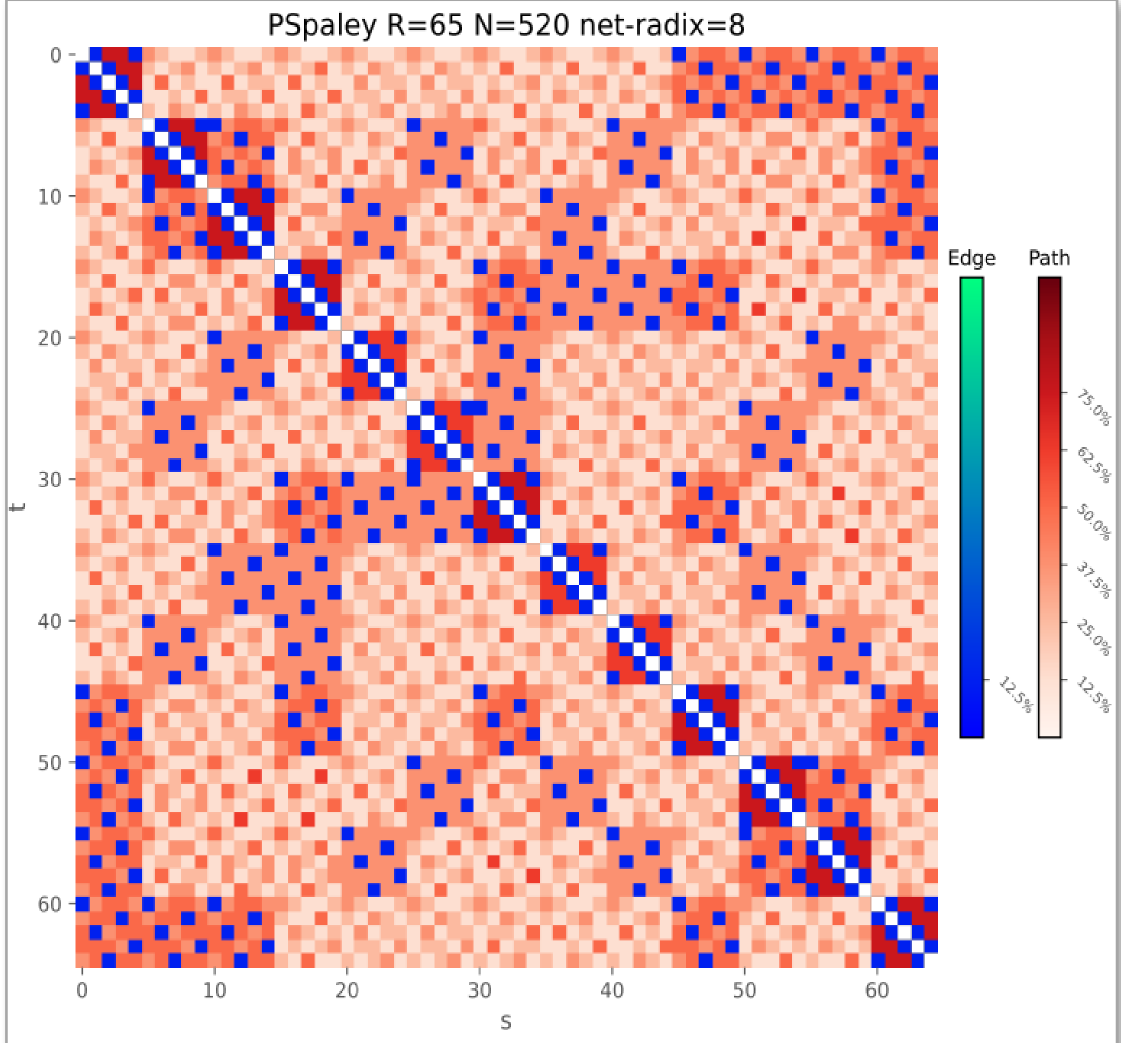
The same (and more) design parameters are given by the specific Analysis class

Some design elements for all types of analysis (directly coded in the Plotter class)

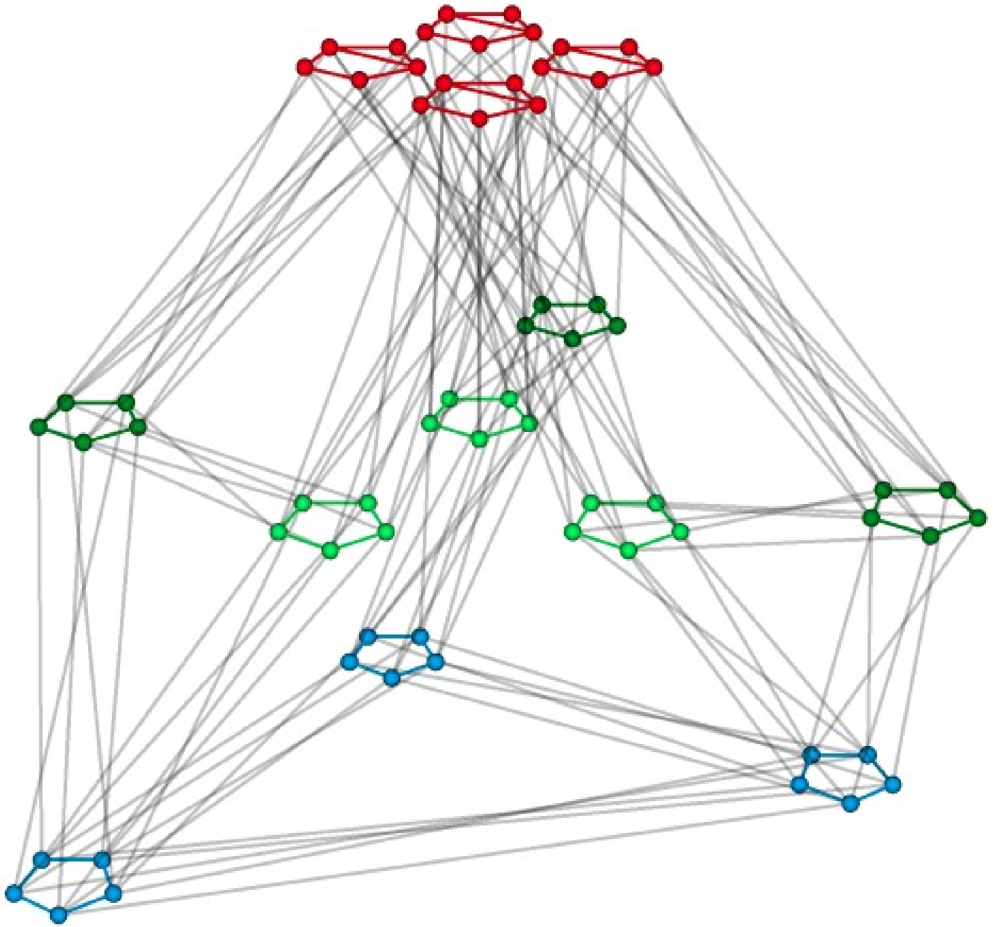
Easily readable code (each analysis type is handled separately)

Possibility to show plots and adjust some Pyplot parameters

Low Connectivity - Polarstar



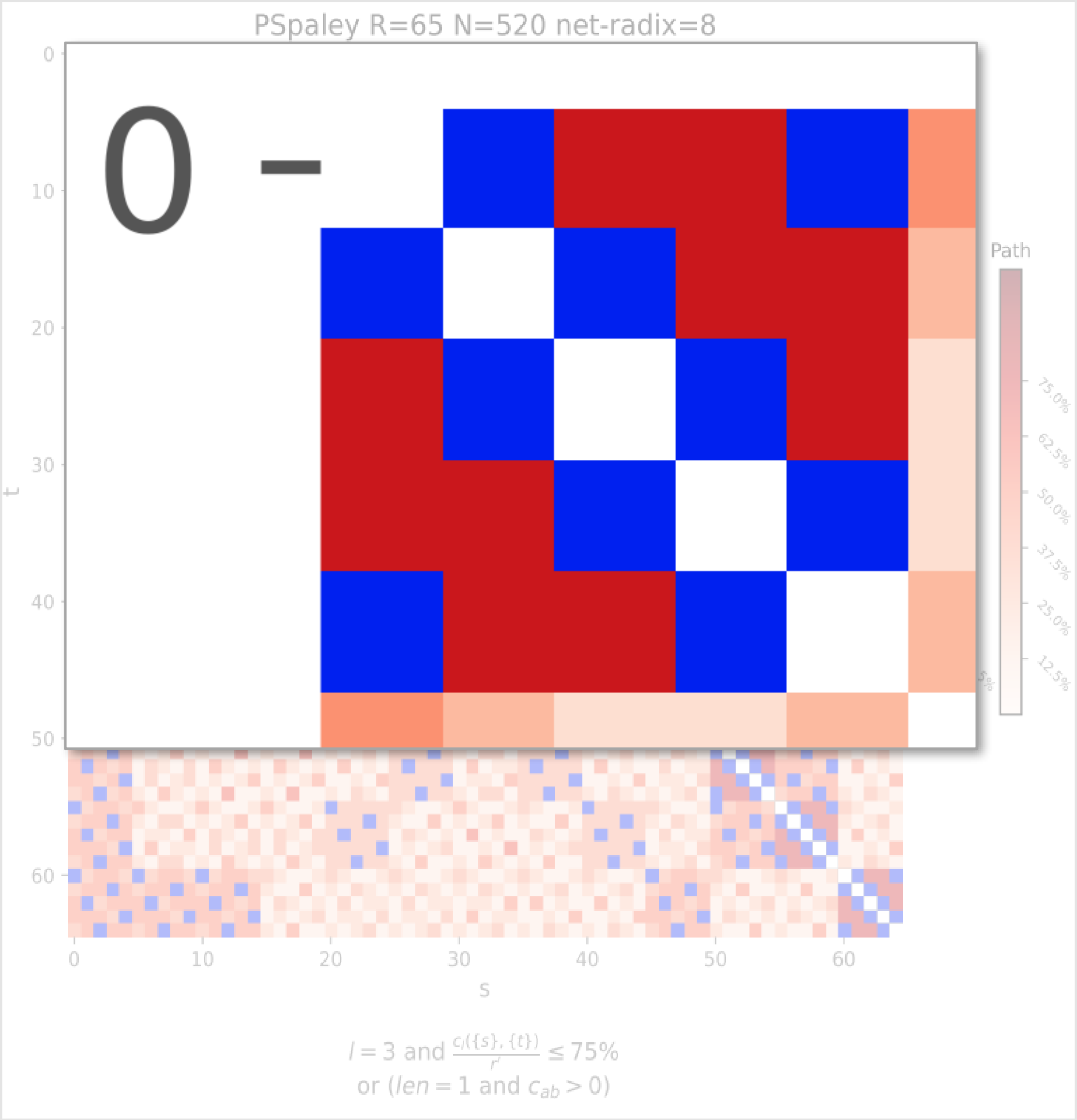
$l = 3$ and $\frac{c_l(\{s\},\{t\})}{r'} \leq 75\%$
or $(len = 1$ and $c_{ab} > 0)$

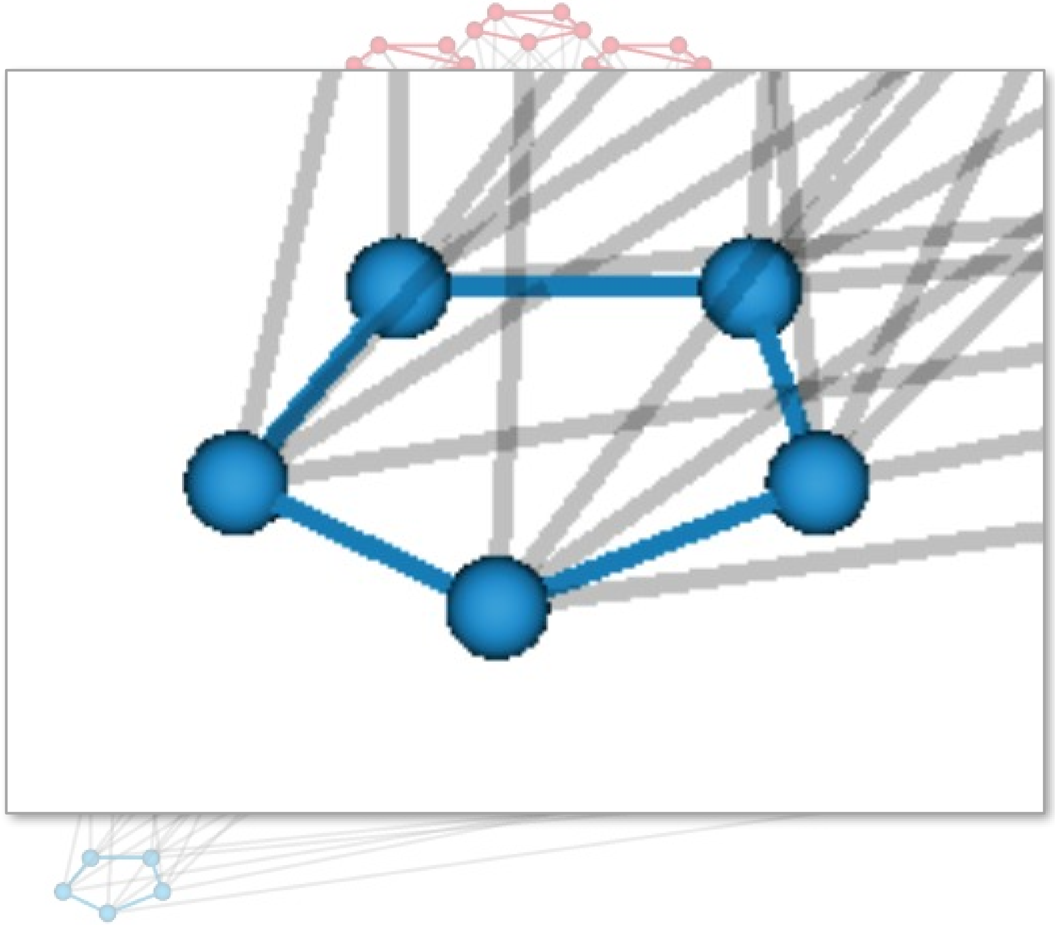


$G{*}G'\colon ER_3 * Paley(5)$

Source: PolarStar: Expanding the Scalability Horizon of Diameter-3 Networks. K. Lakhotia, L. Monroe, K. Isham, M. Besta, N. Blach, T. Hoefler, F. Petrini

Low Connectivity - Polarstar

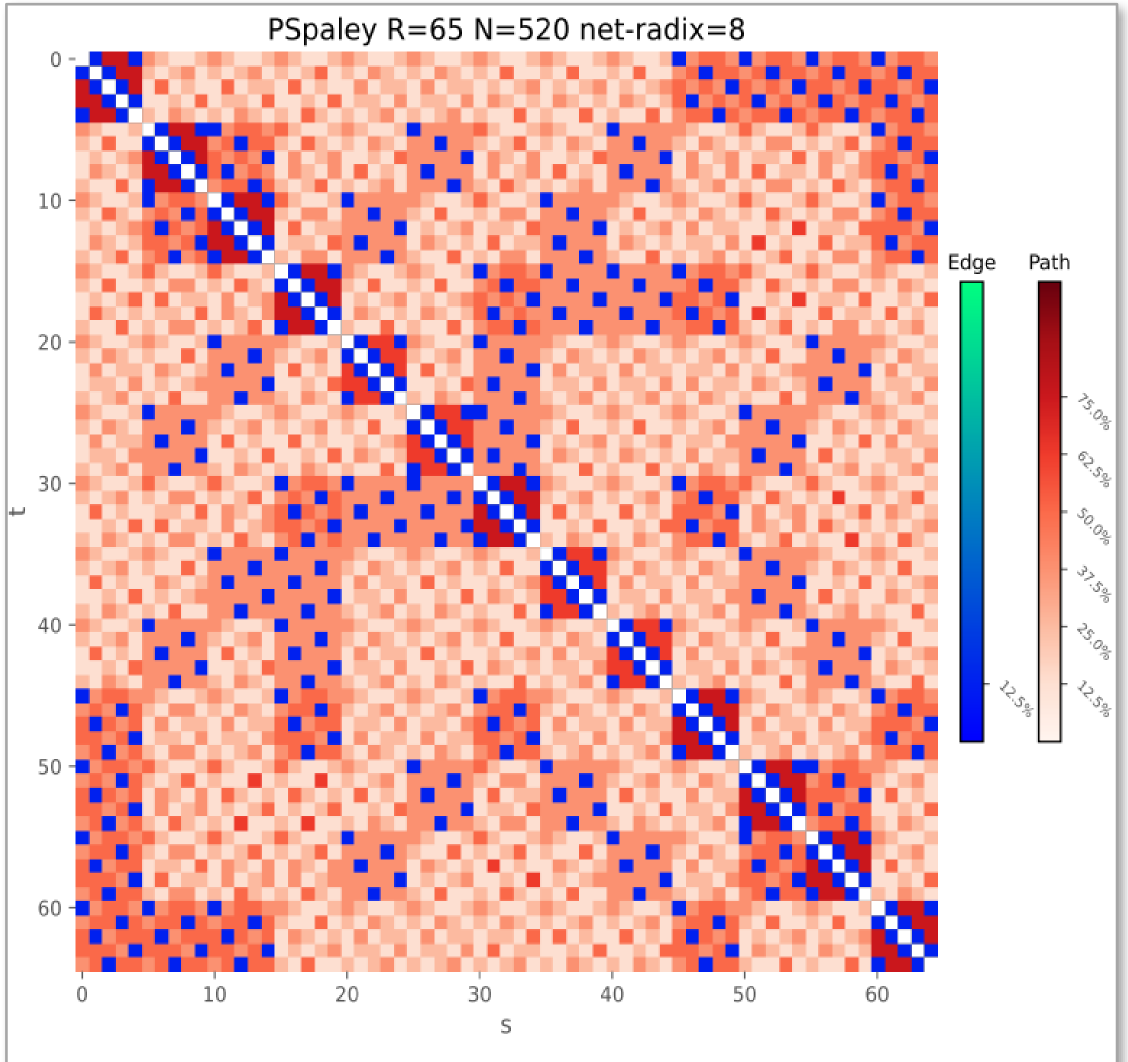




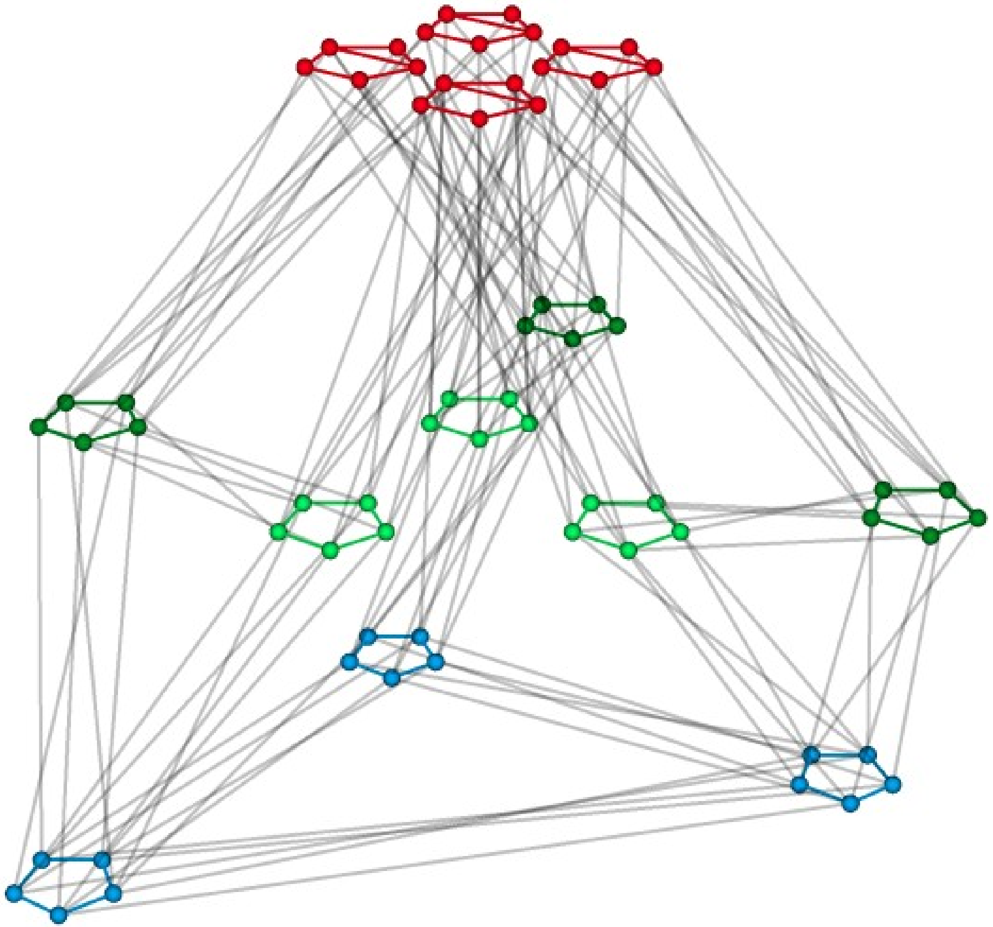
$G{*}G'$: $ER_3 * Paley(5)$

Source: PolarStar: Expanding the Scalability Horizon of Diameter-3 Networks. K. Lakhotia, L. Monroe, K. Isham, M. Besta, N. Blach, T. Hoefler, F. Petrini

Low Connectivity - Polarstar



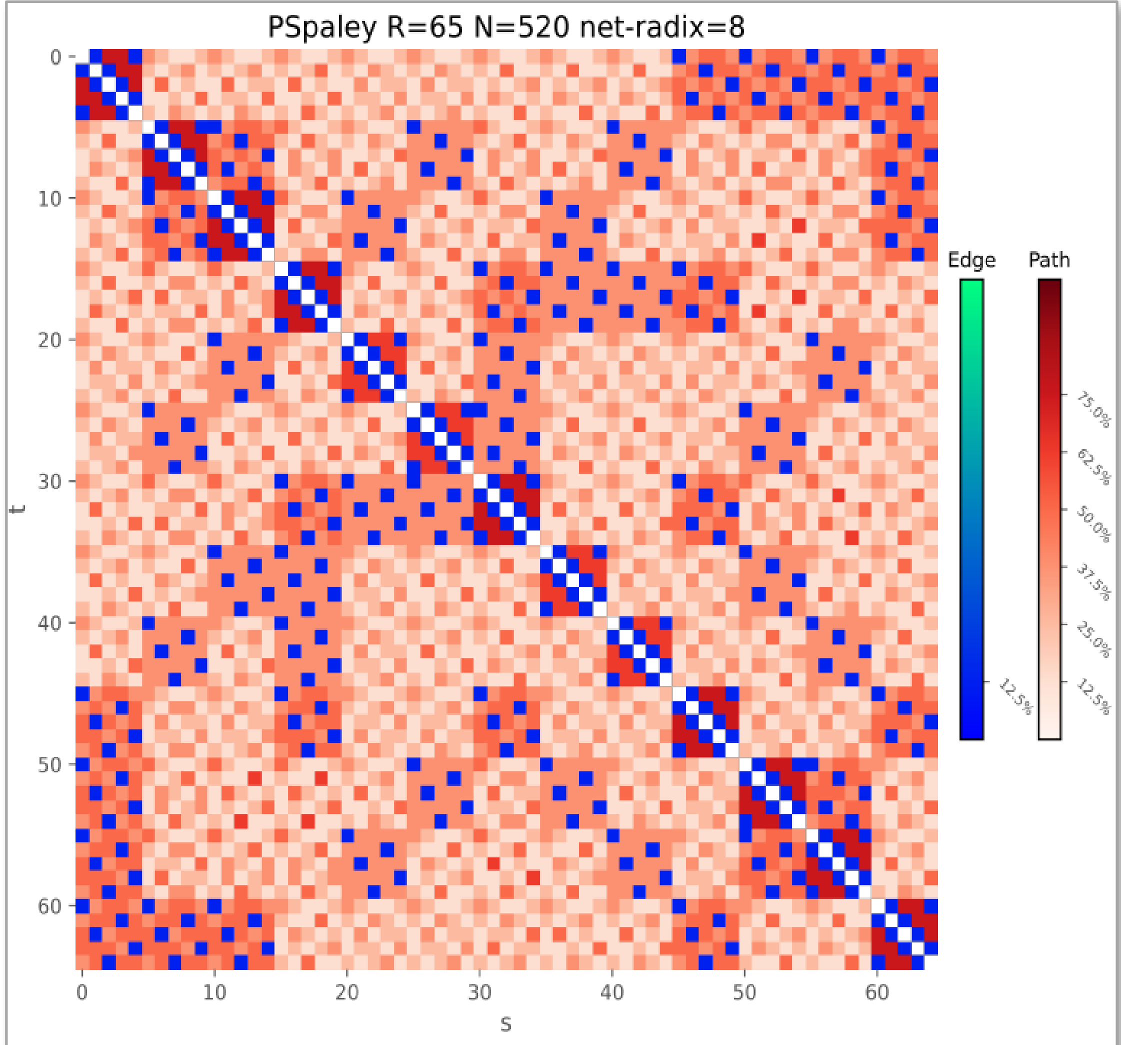
$l = 3$ and $\frac{c_l(\{s\},\{t\})}{r'} \leq 75\%$
or ($len = 1$ and $c_{ab} > 0$)



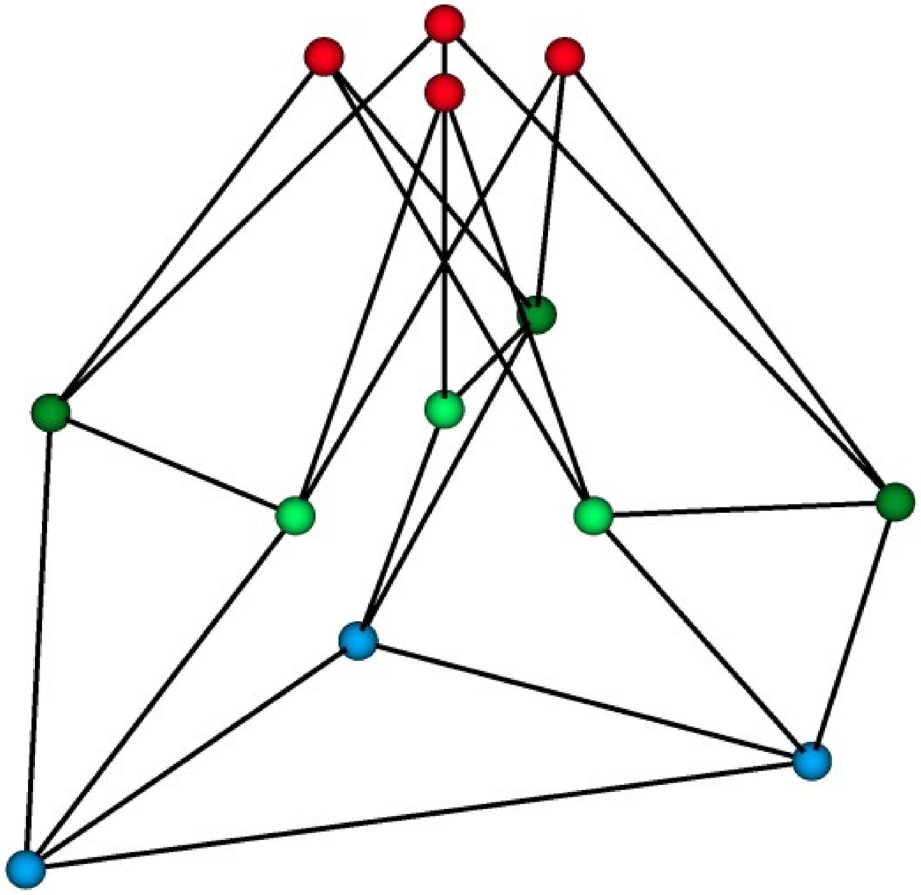
$G{*}G'\colon ER_3 * Paley(5)$

Source: PolarStar: Expanding the Scalability Horizon of Diameter-3 Networks. K. Lakhotia, L. Monroe, K. Isham, M. Besta, N. Blach, T. Hoefler, F. Petrini

Low Connectivity - Polarstar



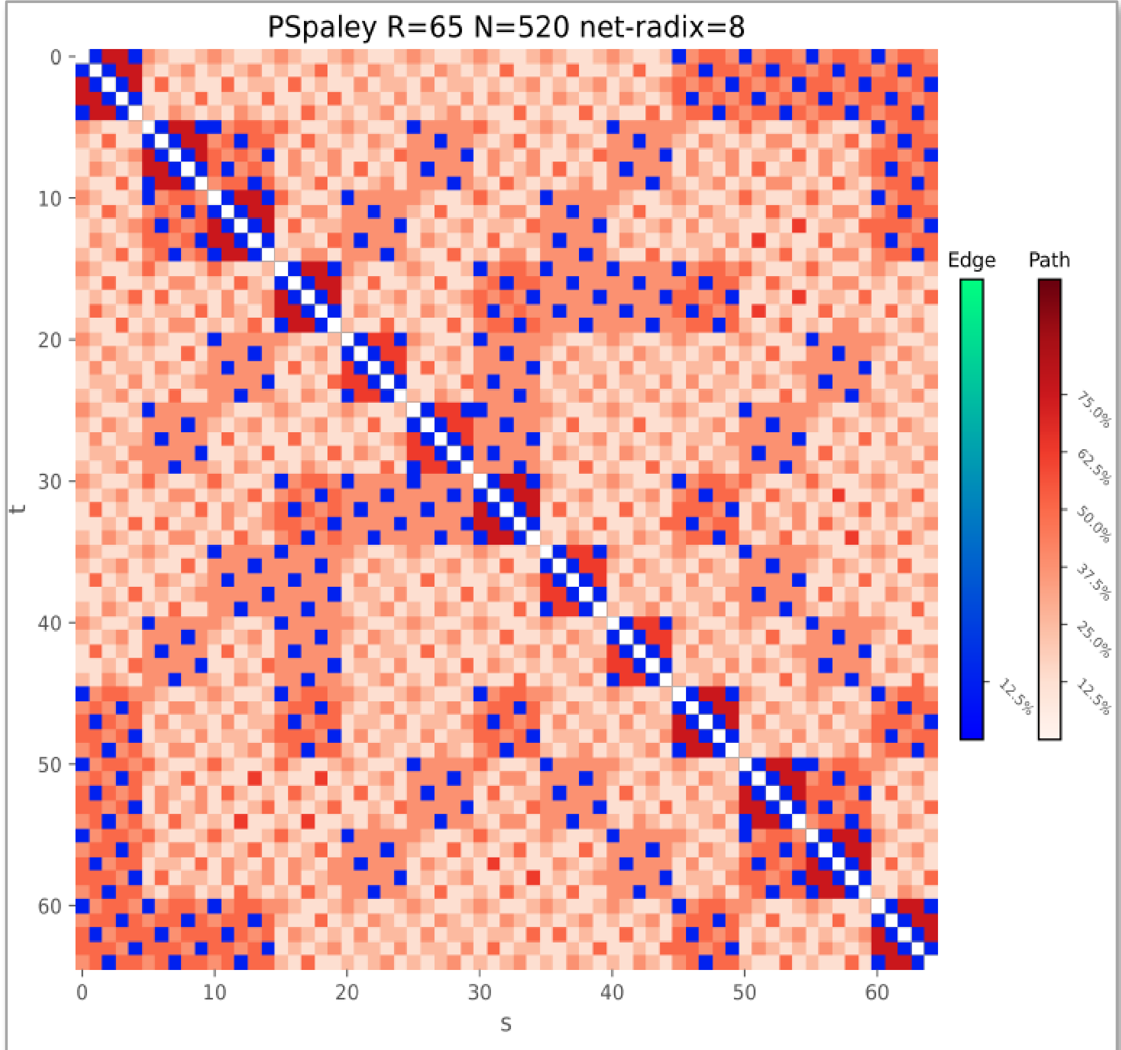
$l = 3$ and $\frac{c_l(\{s\},\{t\})}{r'} \leq 75\%$
or ($len = 1$ and $c_{ab} > 0$)



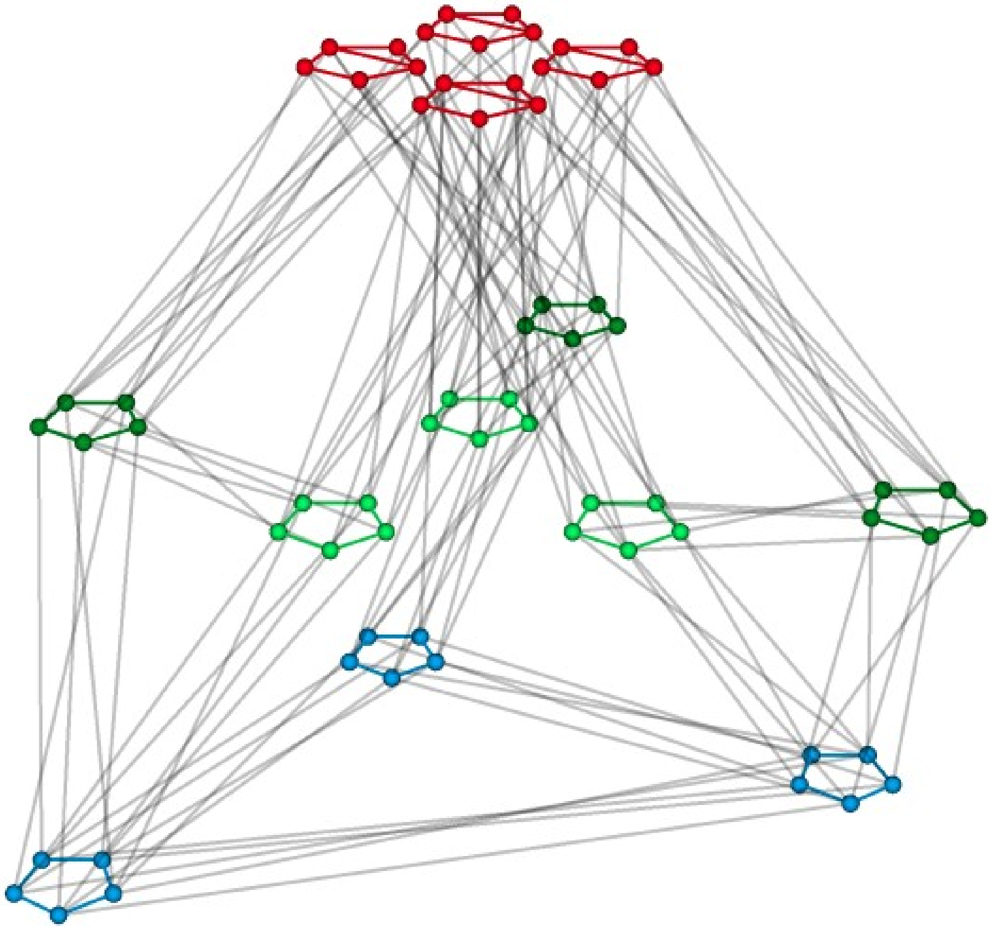
Structure graph G: ER_3

Source: PolarStar: Expanding the Scalability Horizon of Diameter-3 Networks. K. Lakhotia, L. Monroe, K. Isham, M. Besta, N. Blach, T. Hoefler, F. Petrini

Low Connectivity - Polarstar



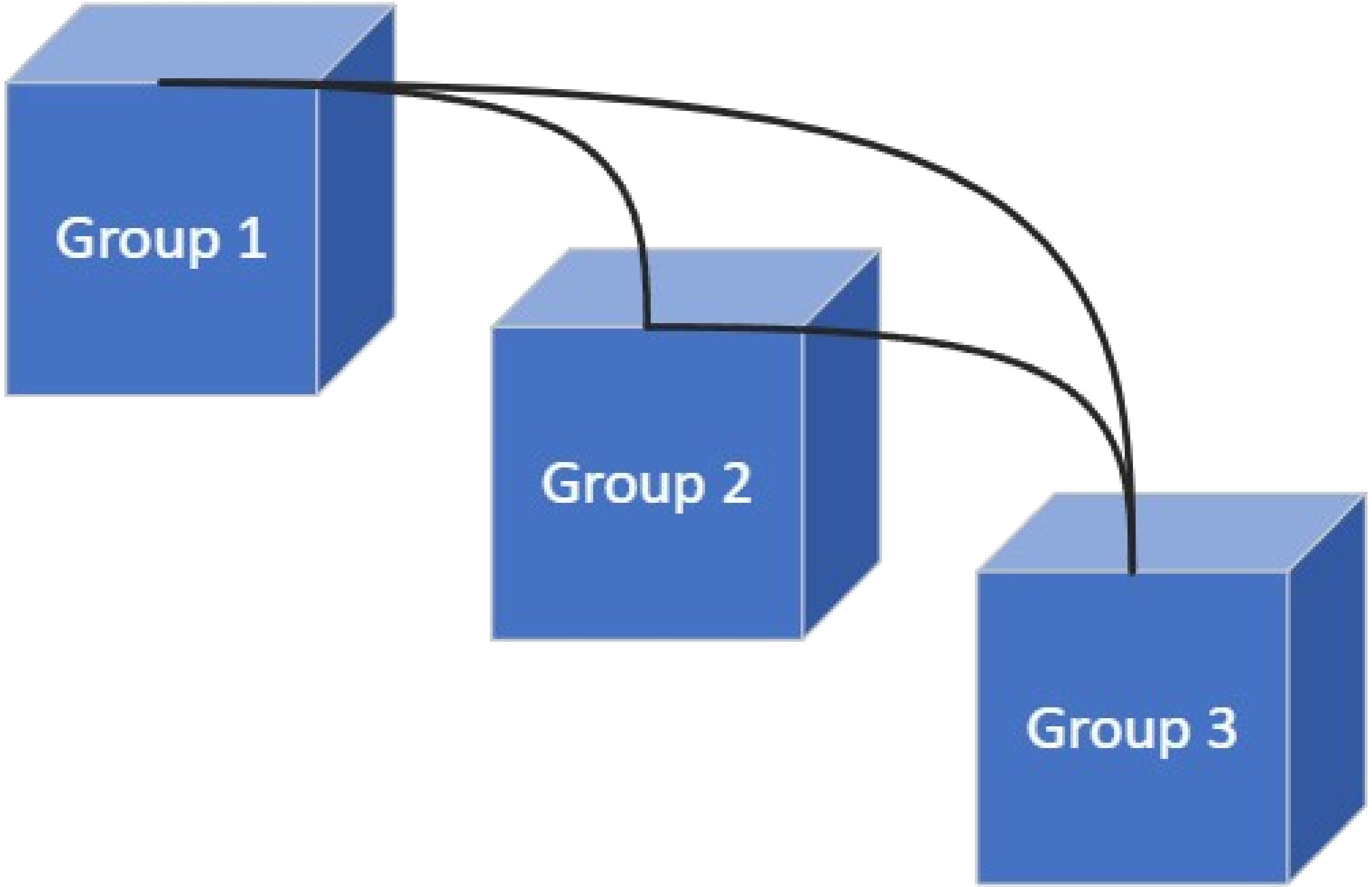
$l = 3$ and $\frac{c_l(\{s\},\{t\})}{r'} \le 75\%$
or ($len = 1$ and $c_{ab} > 0$)



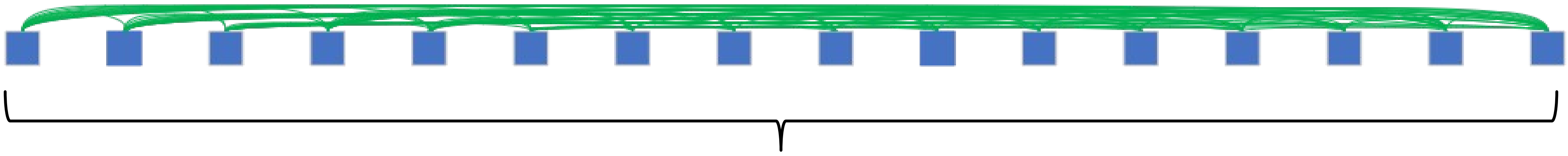
$G{*}G'$: $ER_3 * Paley(5)$

Source: PolarStar: Expanding the Scalability Horizon of Diameter-3 Networks. K. Lakhotia, L. Monroe, K. Isham, M. Besta, N. Blach, T. Hoefler, F. Petrini

Low Connectivity - Cascade Dragonfly

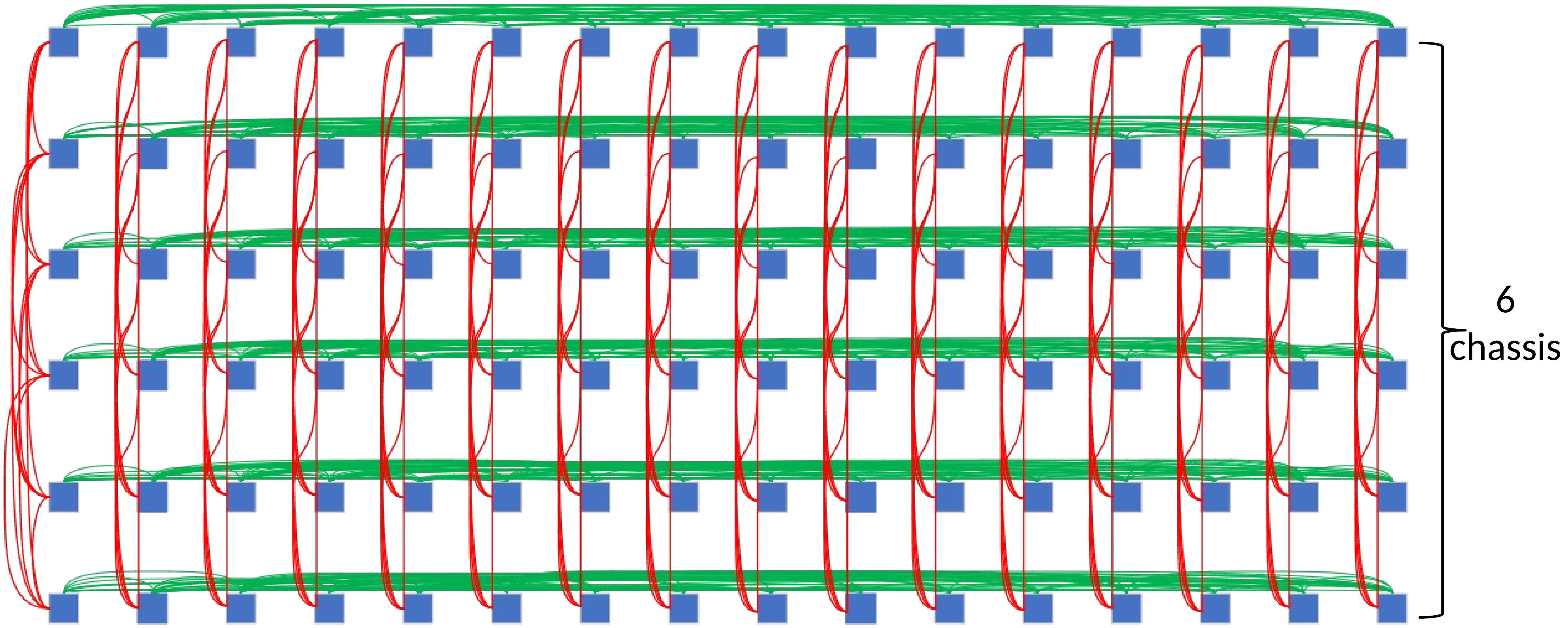


Low Connectivity - Cascade Dragonfly

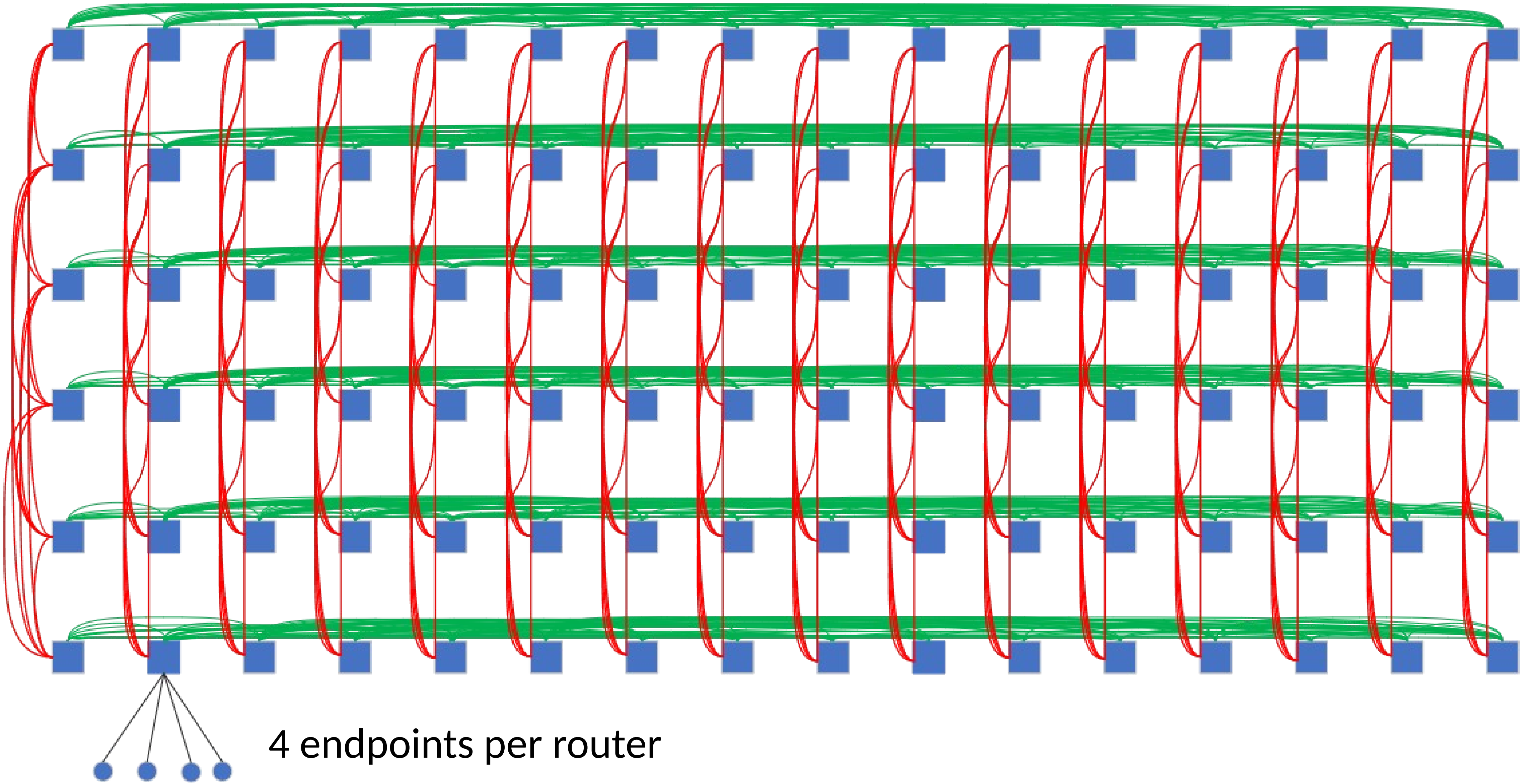


16 Aries router per chassis

Low Connectivity - Cascade Dragonfly

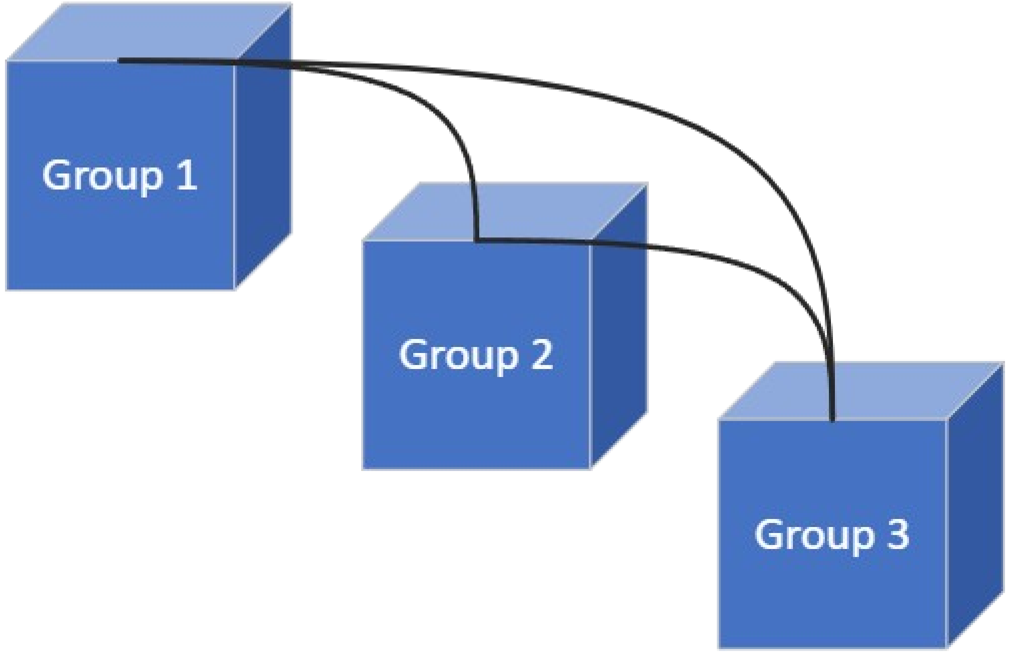


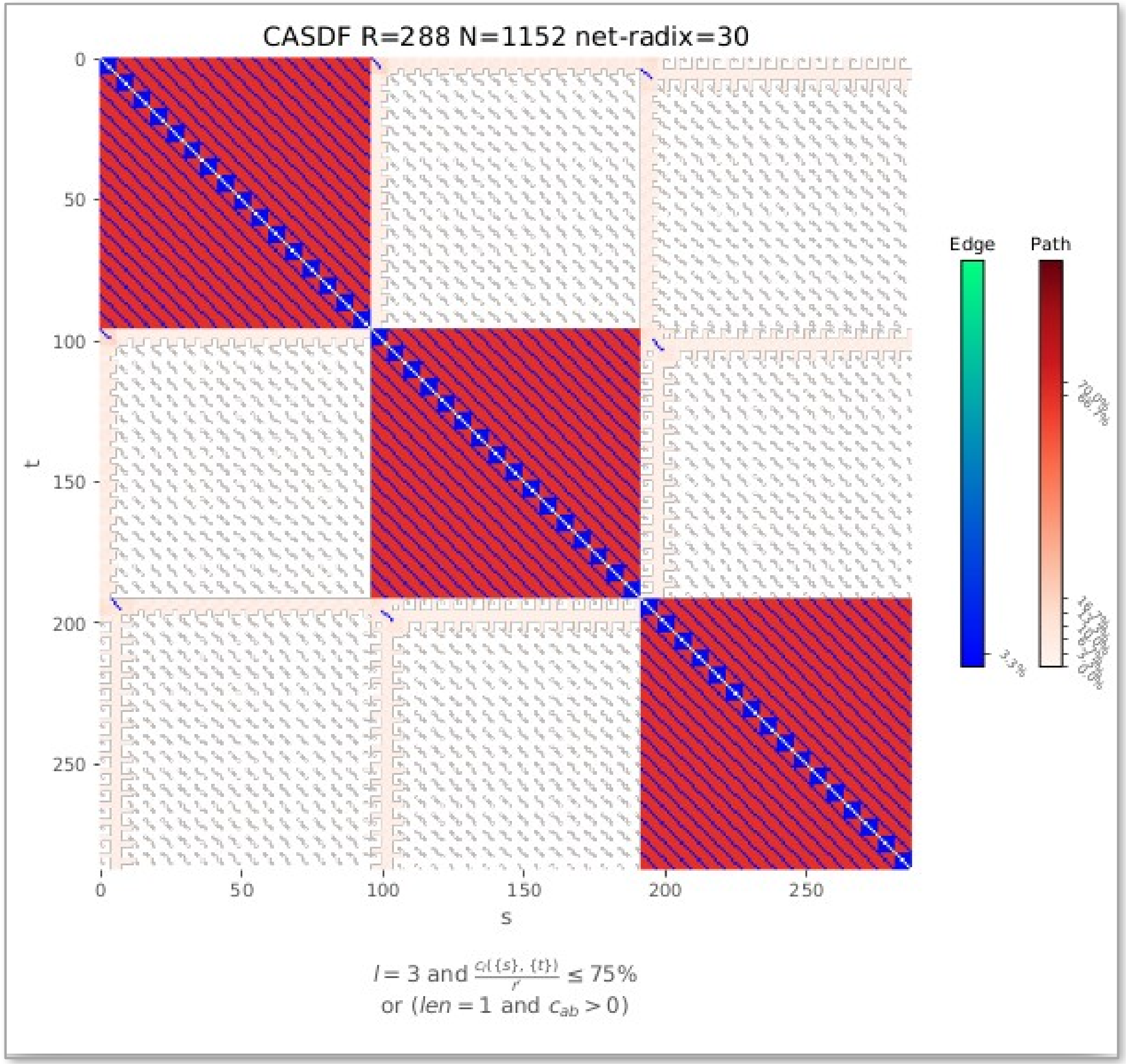
Low Connectivity - Cascade Dragonfly



4 endpoints per router

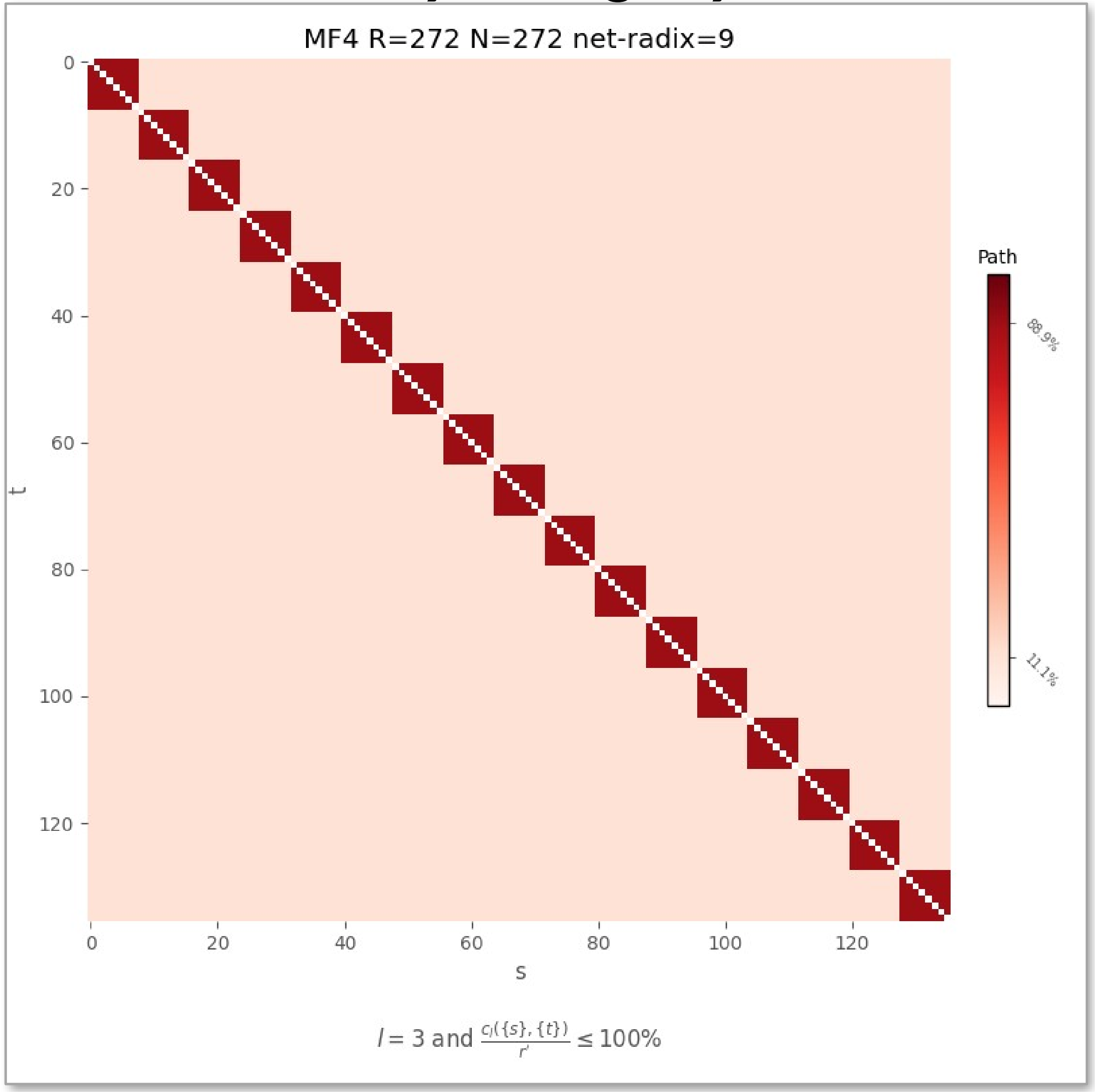
Low Connectivity - Cascade Dragonfly



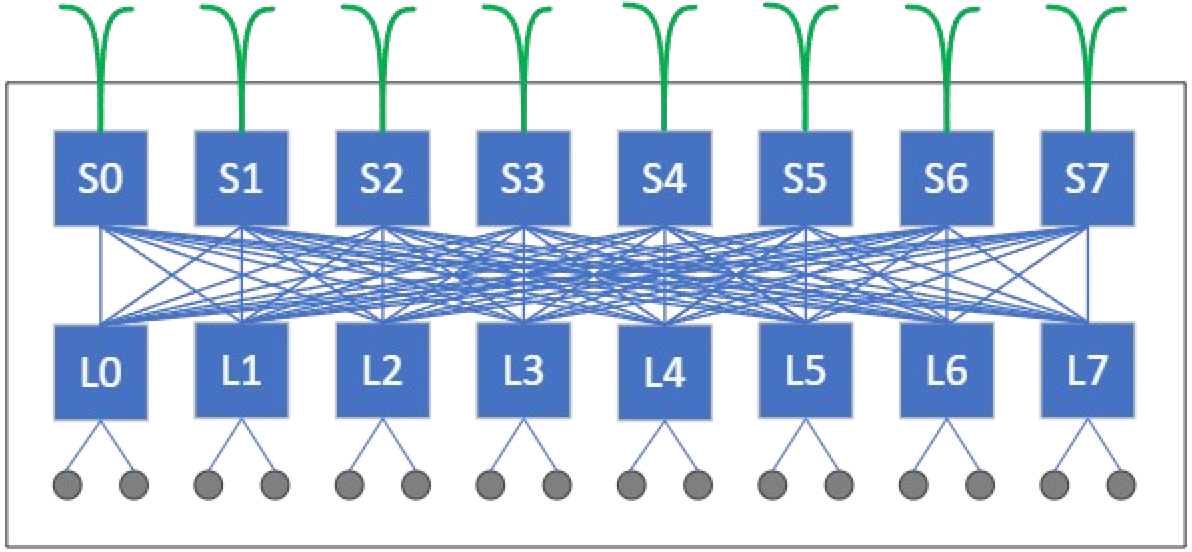


$l = 3$ and $\frac{c(\{s\},\{t\})}{r'} \leq 75\%$
or ($len = 1$ and $c_{ab} > 0$)

Low connectivity - Megafly

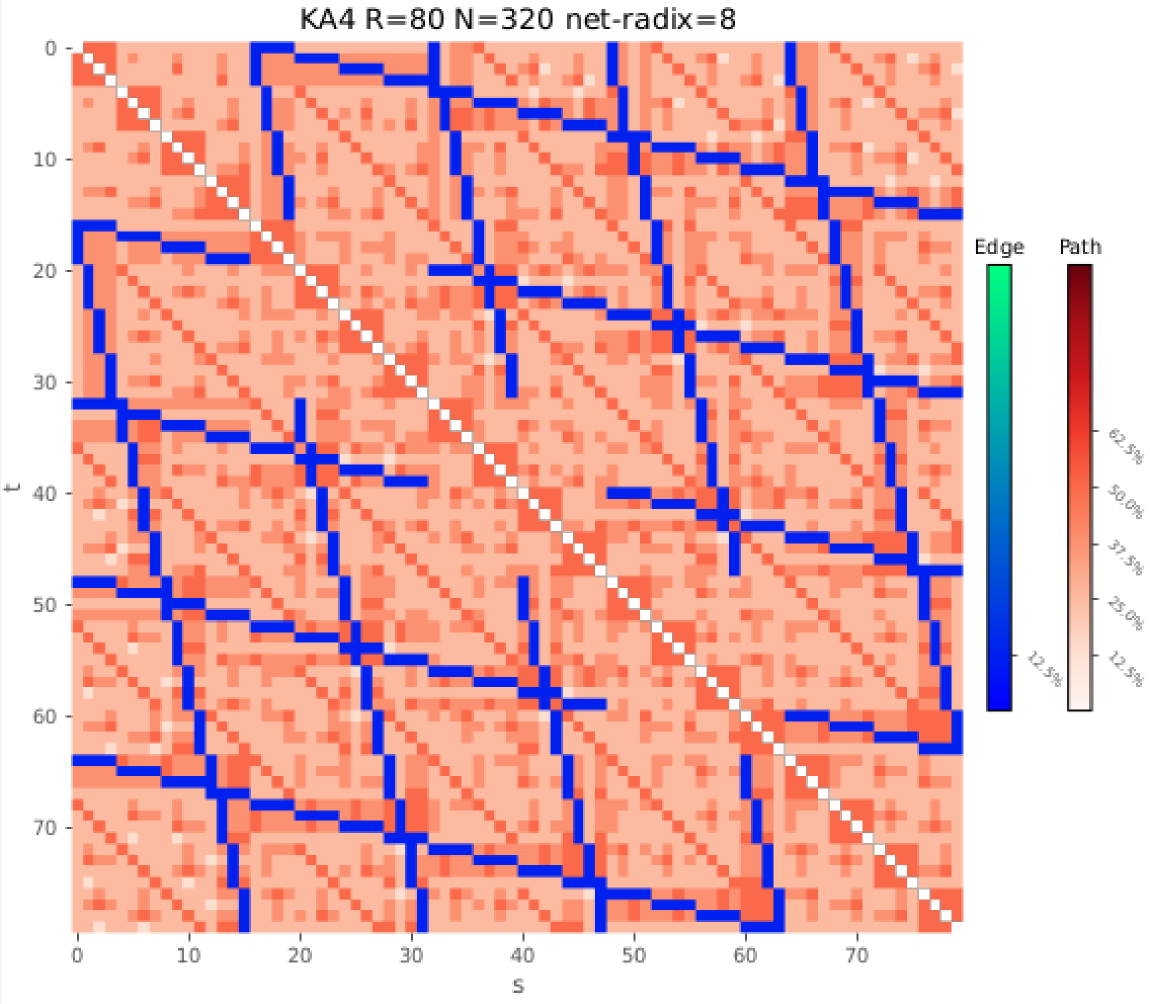


$l = 3$ and $\frac{c_l(\{s\},\{t\})}{r^l} \leq 100\%$

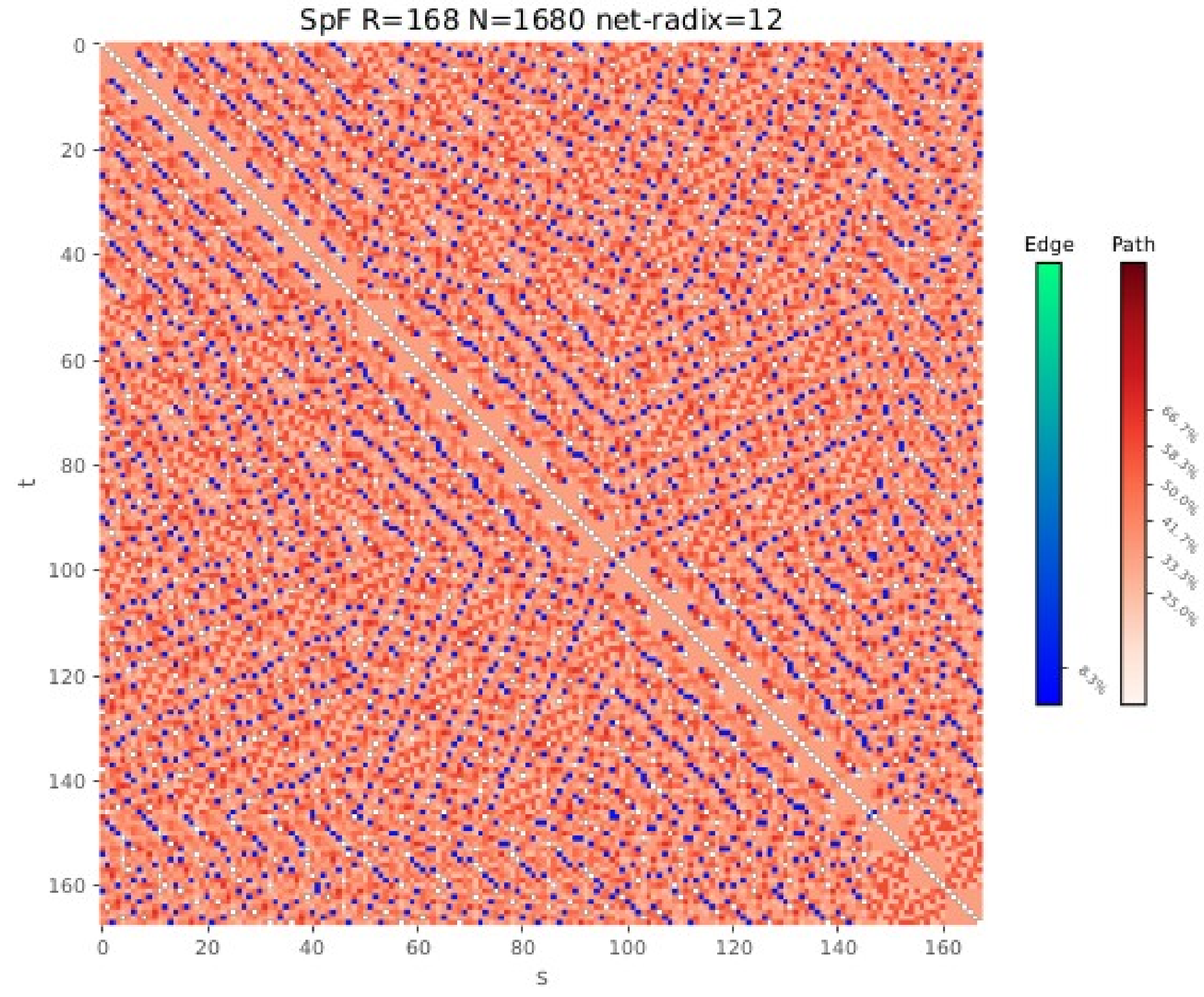


One of the 17 groups

Low connectivity

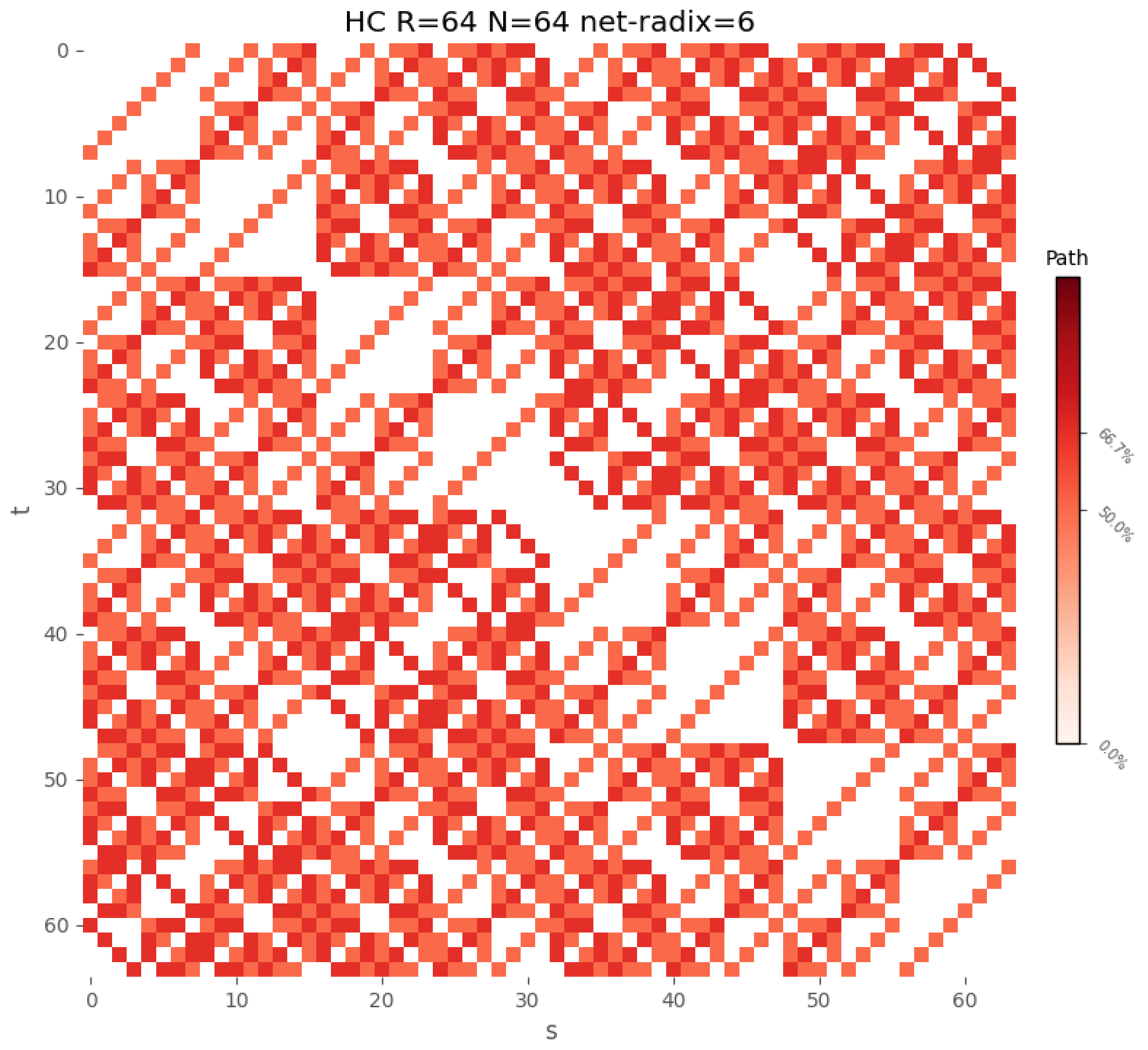


$l = 3$ and $\frac{c_l(\{s\},\{t\})}{r^l} \leq 75\%$
or ($len = 1$ and $c_{ab} > 0$)



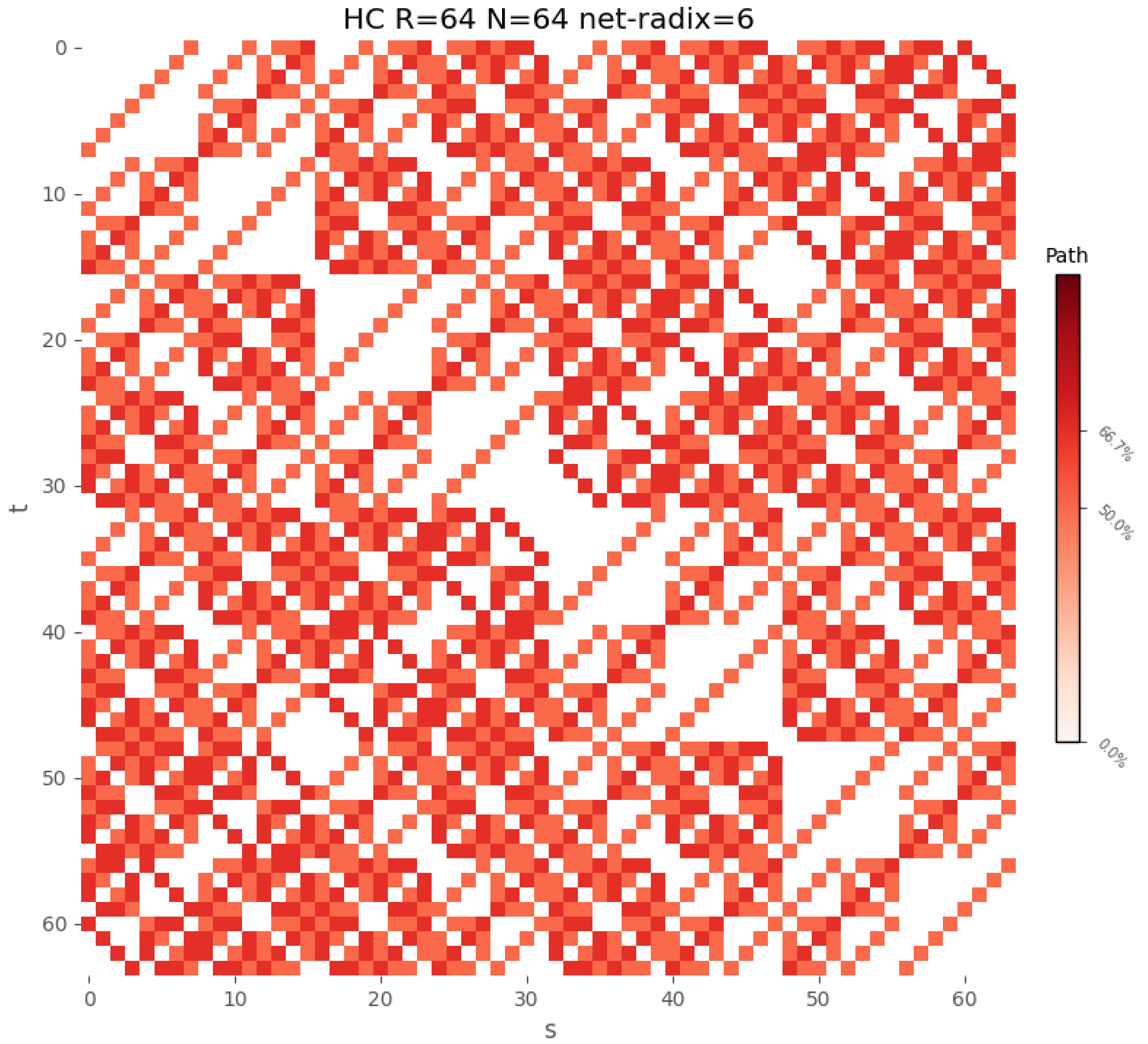
$l = 3$ and $\frac{c_l(\{s\},\{t\})}{r^l} \leq 75\%$
or ($len = 1$ and $c_{ab} > 0$)

Low connectivity

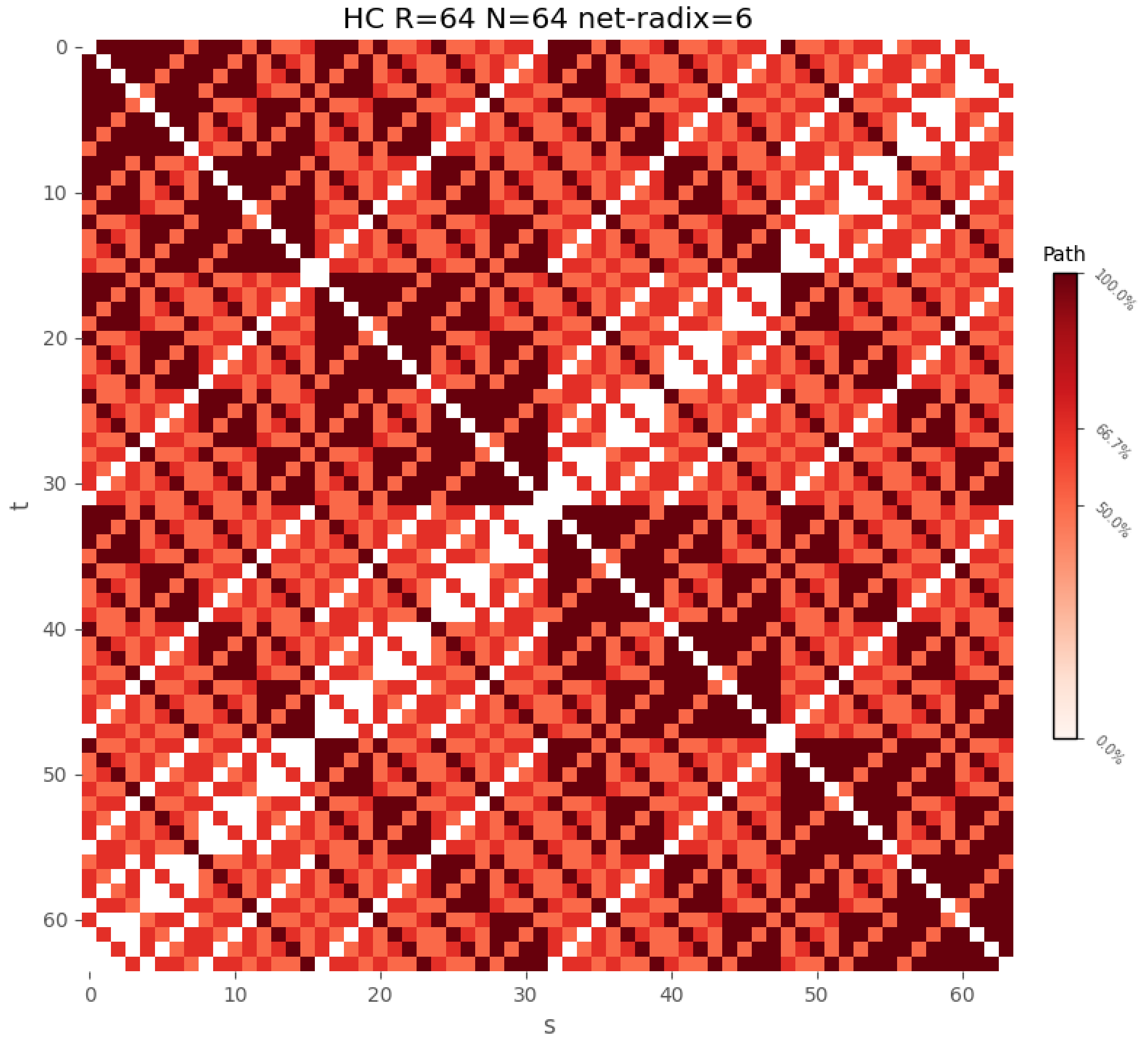


$l = 4$ and $\frac{c_l(\{s\},\{t\})}{r'} \leq 75\%$

Low connectivity

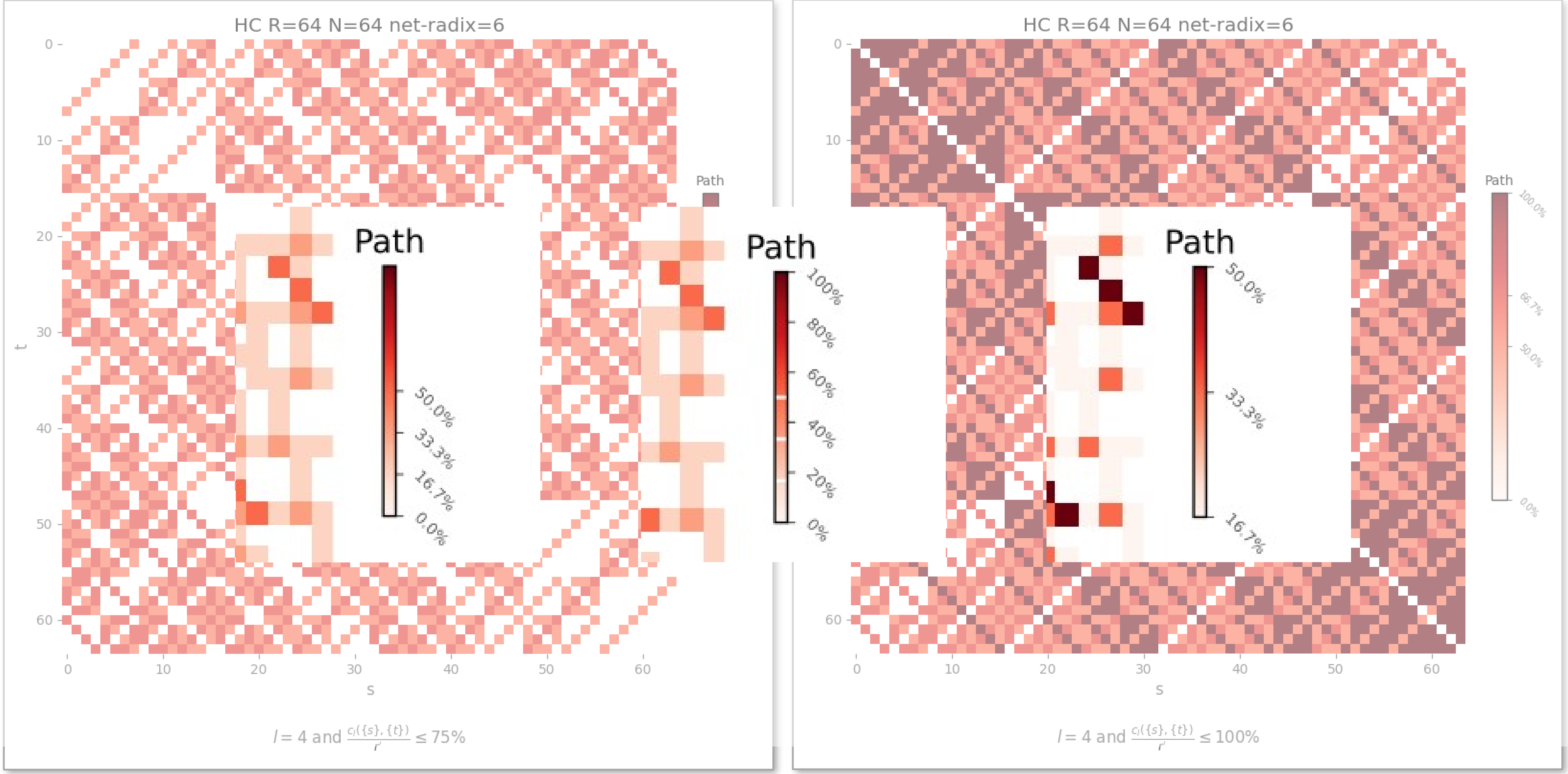


$l = 4$ and $\frac{c_l(\{s\},\{t\})}{r'} \leq 75\%$



$l = 4$ and $\frac{c_l(\{s\},\{t\})}{r'} \leq 100\%$

Low connectivity



$l = 4$ and $\frac{c_l(\{s\},\{t\})}{r'} \leq 75\%$

$l = 4$ and $\frac{c_l(\{s\},\{t\})}{r'} \leq 100\%$

Conclusion

Conclusion

Torus

Hypercube

Slimfly

Fat tree2x

Dragonfly

Kautz

Expander

Mesh

MLFM

Polarfly

Tofu

Fat tree

HyperX

Spectralfly

Cascade
Dragonfly

Megafly

Arrangement
graph

Express
Mesh

Polarstar

XGFT

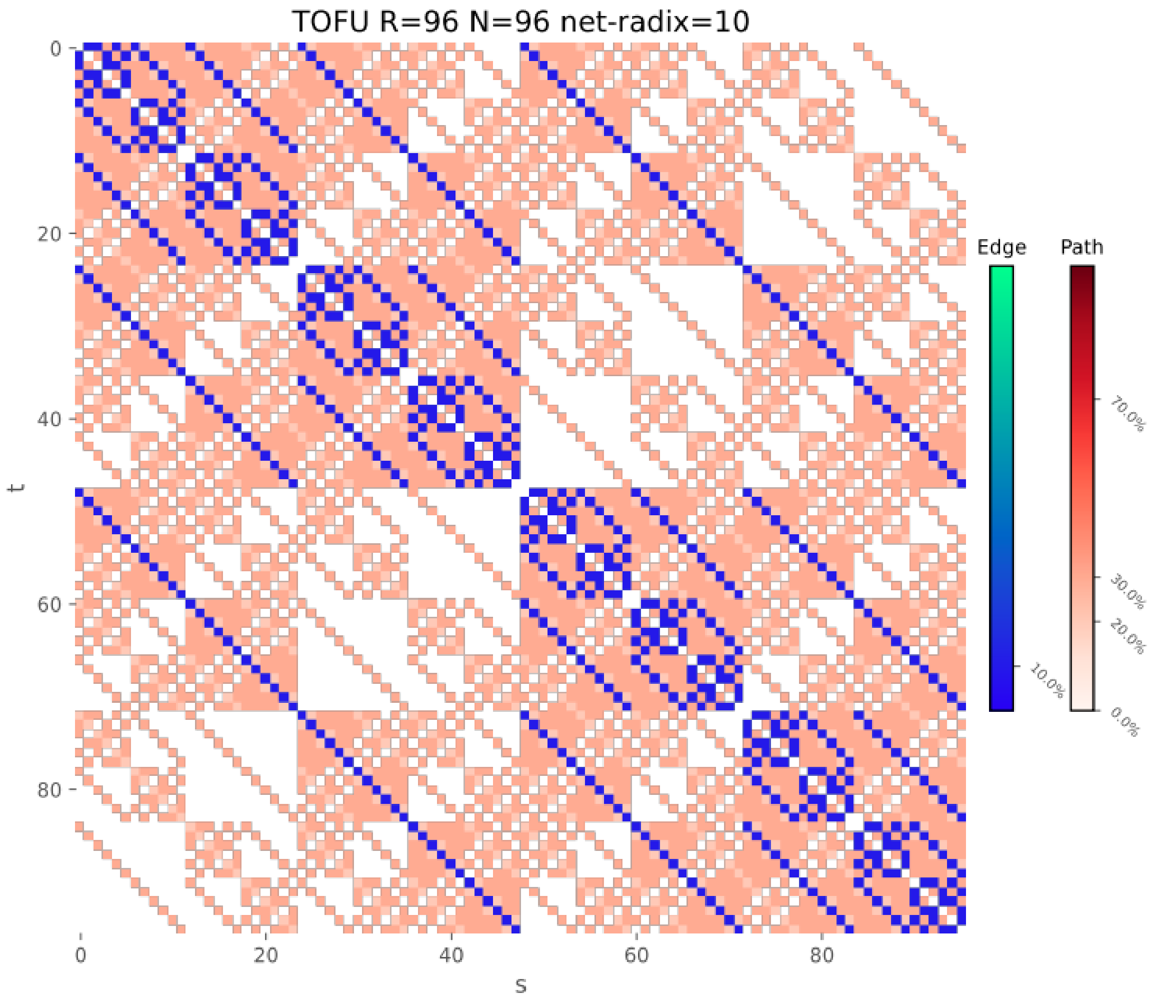
Flattened
Butterfly

K-ary n-tree

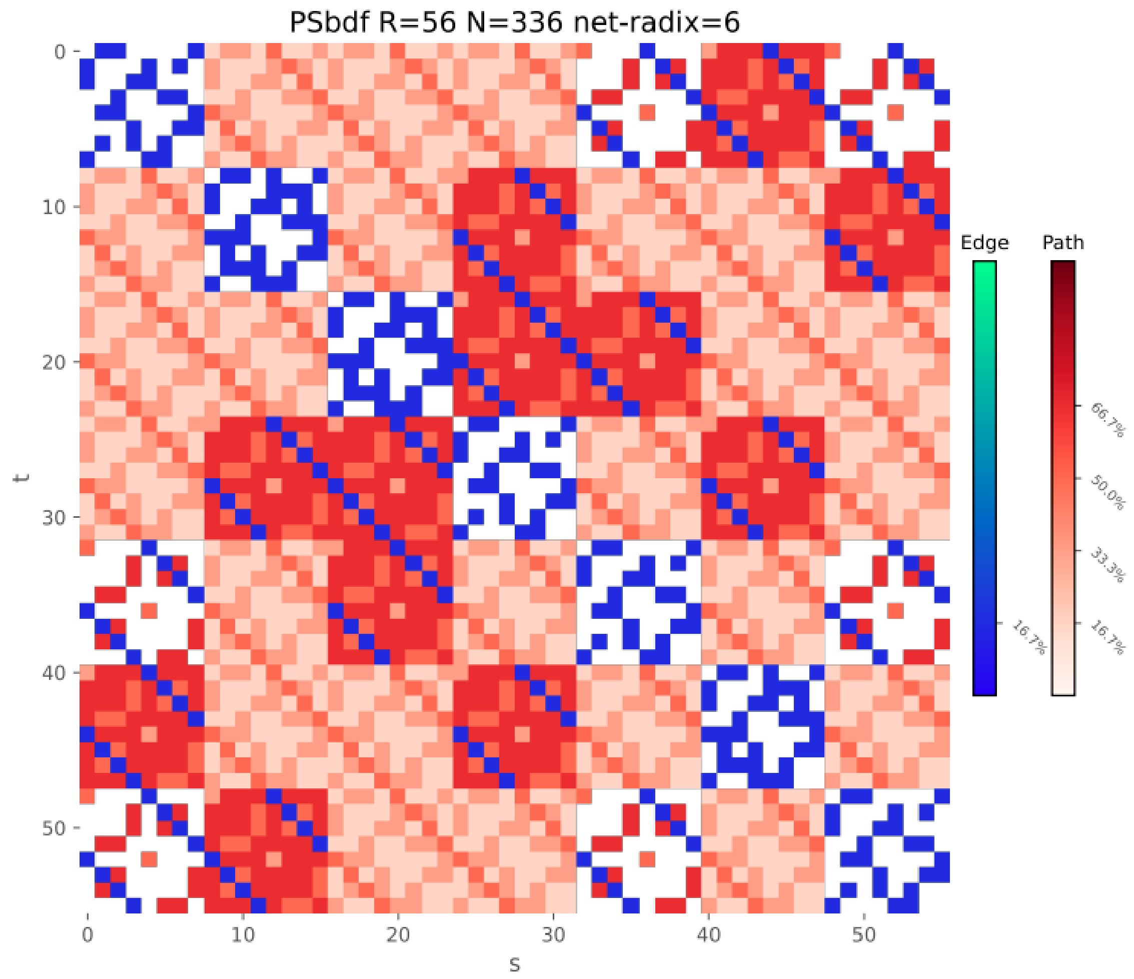
Random
(Jellyfish)

Conclusion

Conclusion



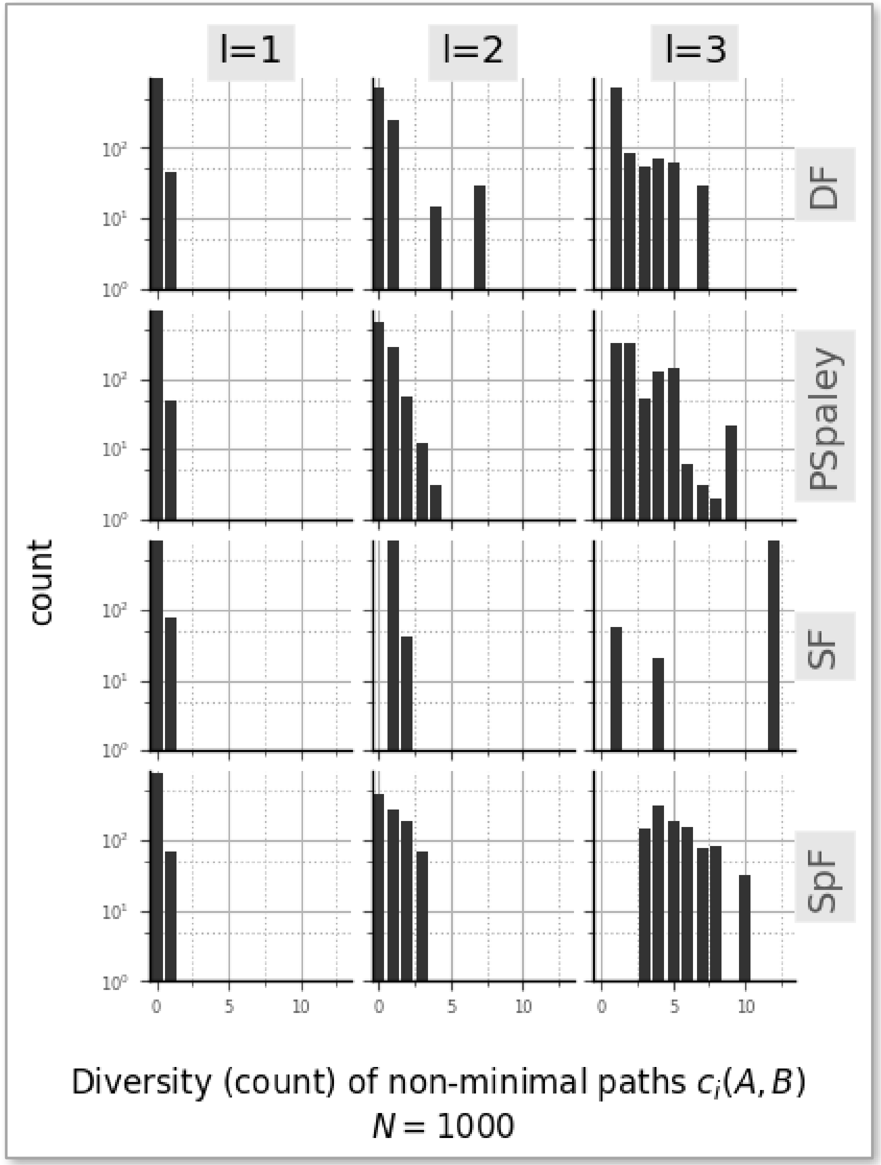
$l = 3$ and $\frac{c_l(\{s\},\{t\})}{r'} \leq 100\%$
or ($len = 1$ and $c_{ab} > 0$)



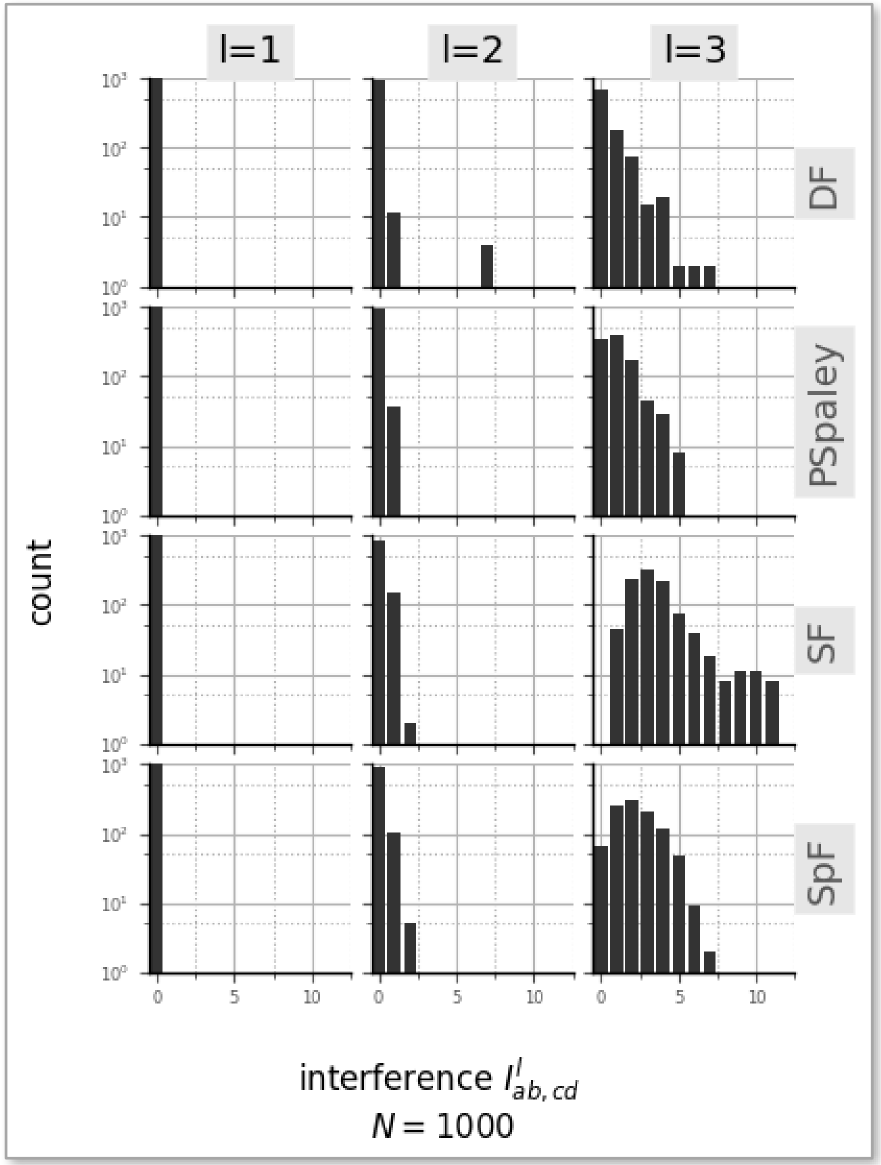
$l = 3$ and $\frac{c_l(\{s\},\{t\})}{r'} \leq 75\%$
or ($len = 1$ and $c_{ab} > 0$)

SPCL

Results

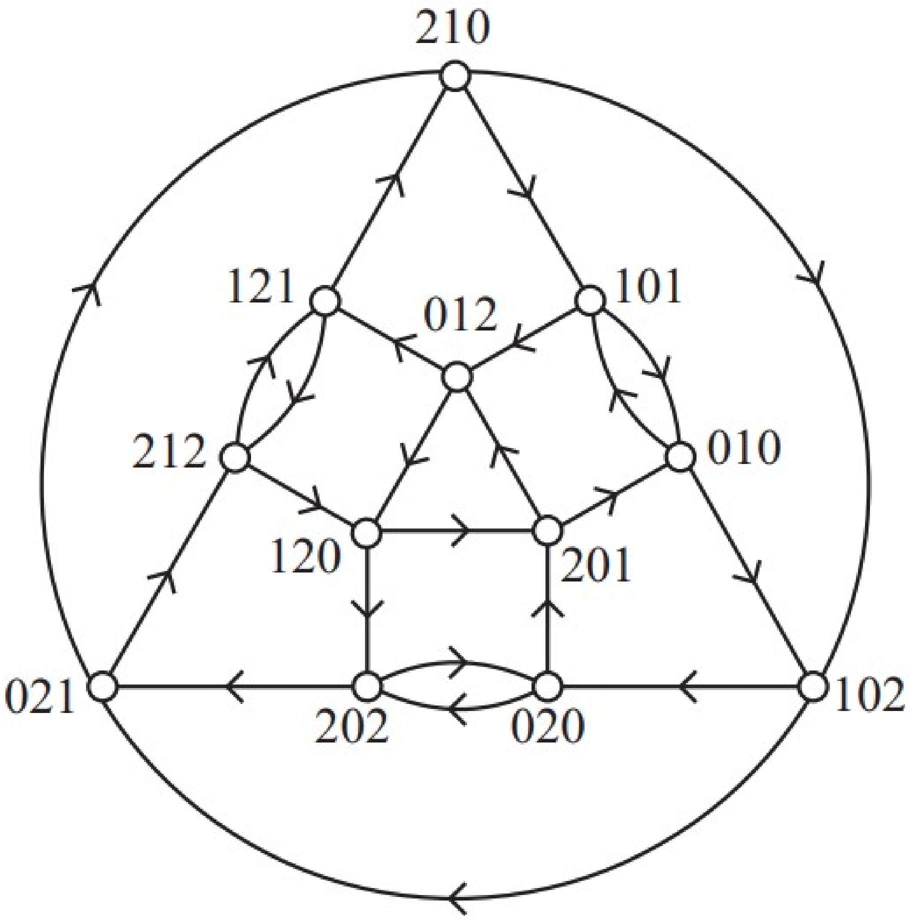


Diversity (count) of non-minimal paths $c_i(A,B)$
$N = 1000$



interference $I^l_{ab,cd}$
$N = 1000$

Kautz



$K(2,3)$

Source: The k-tuple twin domination in de Bruijn and Kautz digraphs. Toru Araki

$\{x_1,x_2,\ldots,x_n\}$ with $x_i \neq x_{i+1}$ defines a node.

A node $\{x_1,x_2,\ldots,x_n\}$ is connected to $\{x_2,\ldots,x_n, \alpha\}$ for all $\alpha \neq x_n$

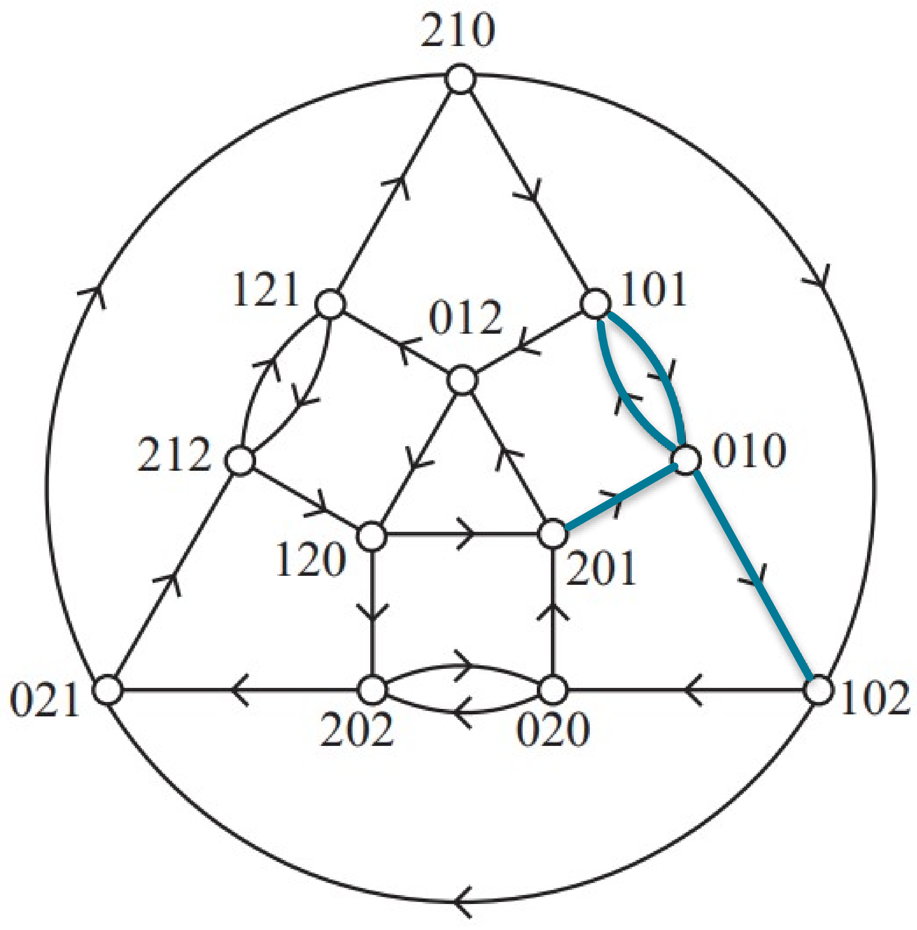
{010} → {101}

→ {102}

{101} → {010}

{201} → {010}

Kautz



$K(2,3)$

Source: The k-tuple twin domination in de Bruijn and Kautz digraphs. Toru Araki

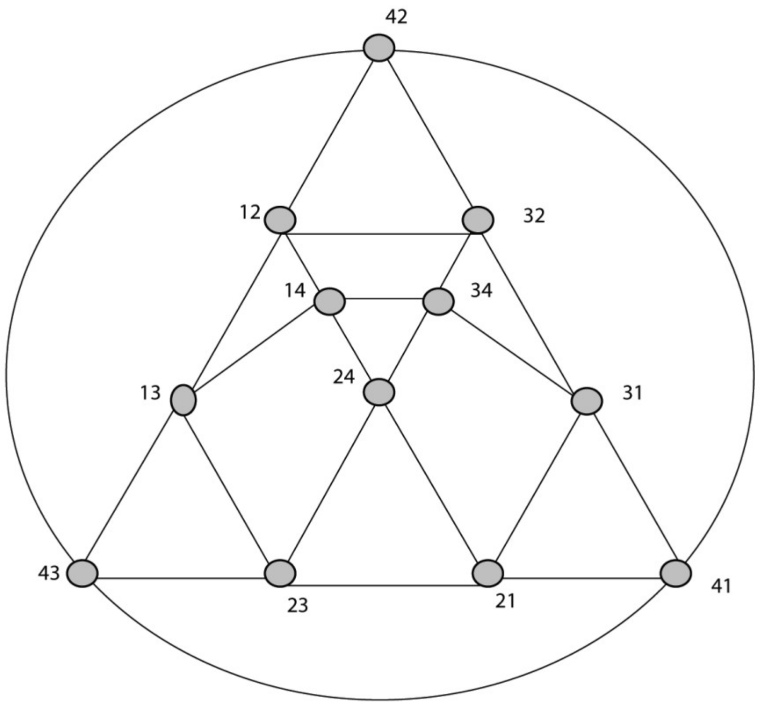
$\{x_1,x_2,\ldots,x_n\}$ with $x_i \neq x_{i+1}$ defines a node.

A node $\{x_1,x_2,\ldots,x_n\}$ is connected to $\{x_2,\ldots,x_n, \alpha\}$ for all $\alpha \neq x_n$

{010} → {101}
→ {102}

{101} → {010}
{201} → {010}

Arrangement



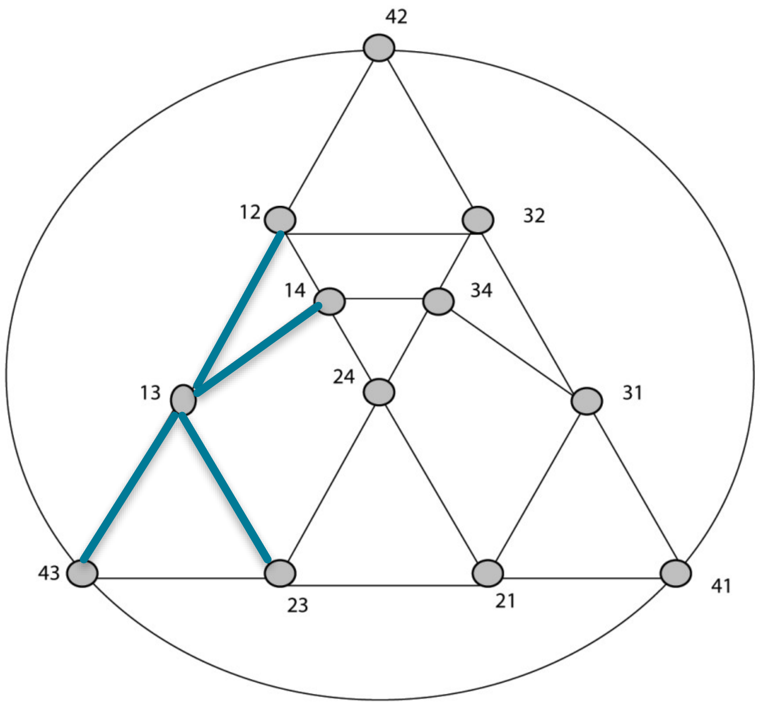
$A_{4,2}$

Source: Structural Outlooks for the OTIS-Arrangement Network.
A. M. Awwad, J. Al-Sadi, B. Haddad, A. Kayed

$\{x_1,x_2,\ldots,x_n\}$ with $x_i \neq x_j$ for $i \neq j$ defines a node.

A node $\{x_1,x_2,\ldots,x_n\}$ is connected to $\{x_1, x_2,\ldots,x_n\}$ if they differ in exactly one position.

Arrangement



$A_{4,2}$

Source: Structural Outlooks for the OTIS-Arrangement Network.
A. M. Awwad, J. Al-Sadi, B. Haddad, A. Kayed

$\{x_1,x_2,\dots,x_n\}$ with $x_i \neq x_j$ for $i \neq j$ defines a node.

A node $\{x_1,x_2,\dots,x_n\}$ is connected to $\{x_1, x_2,\dots,x_n\}$ if they differ in exactly one position.

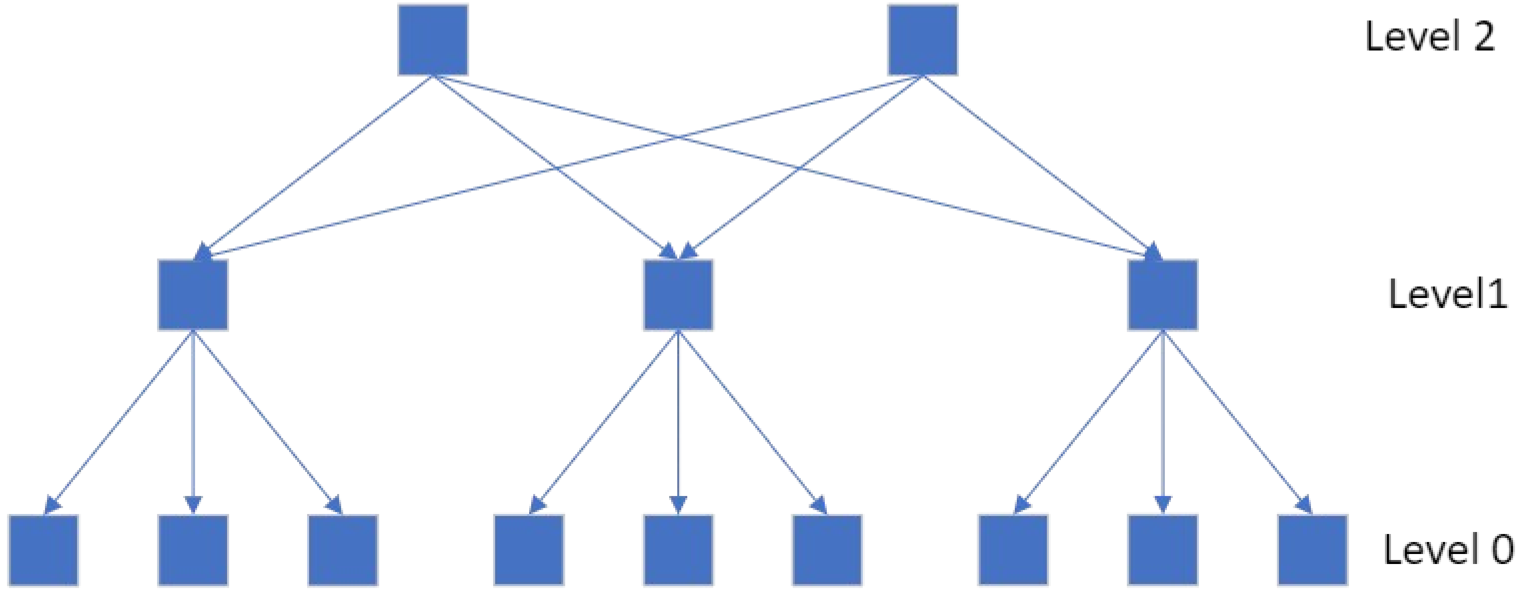
$\{13\} \rightarrow \{14\}$

$\rightarrow \{12\}$

$\{13\} \rightarrow \{23\}$

$\rightarrow \{42\}$

XGFT



XGFT $(h,m_1,\ldots,m_h,w_1,\ldots,w_h)$

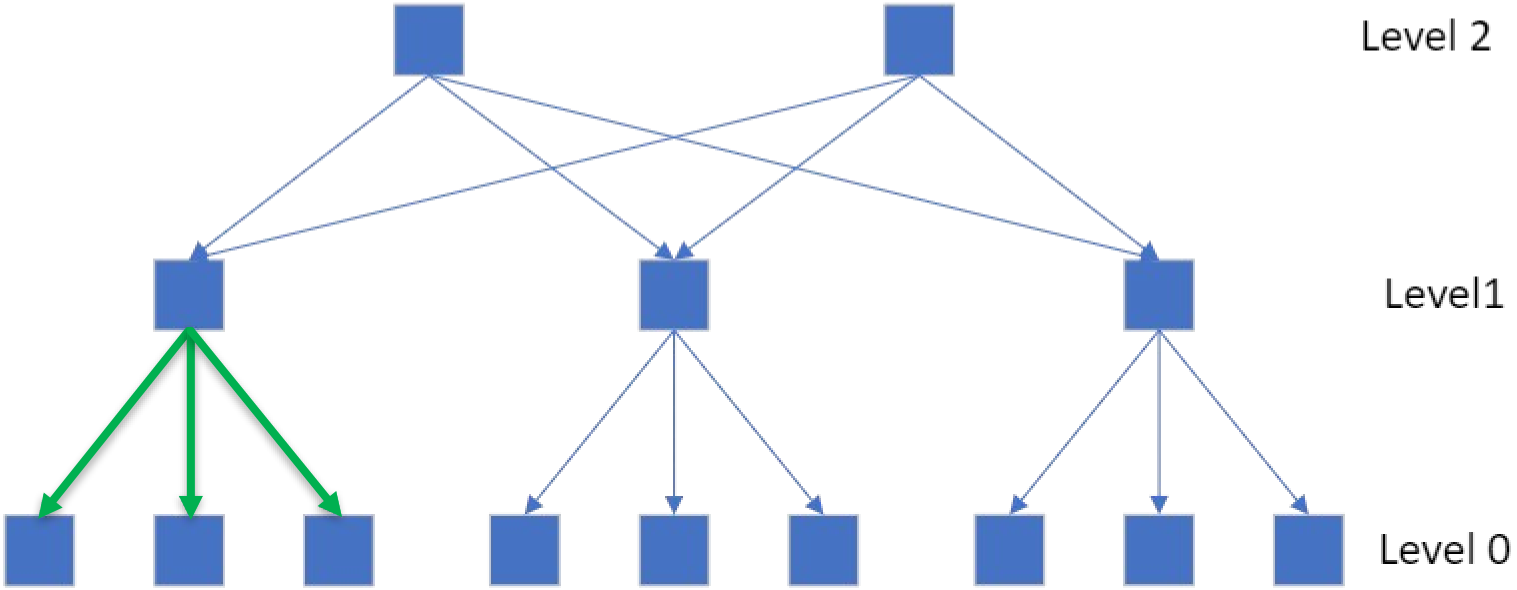
h: Height of the fat tree

$m_1,\ldots,m_{h:}$ Number of children at level i

$w_1,\ldots,w_{h:}$ Number of parents at level i

XGFT(2,3,3,1,2)

XGFT



XGFT (h,m_1,…,m_h,w_1,…,w_h)

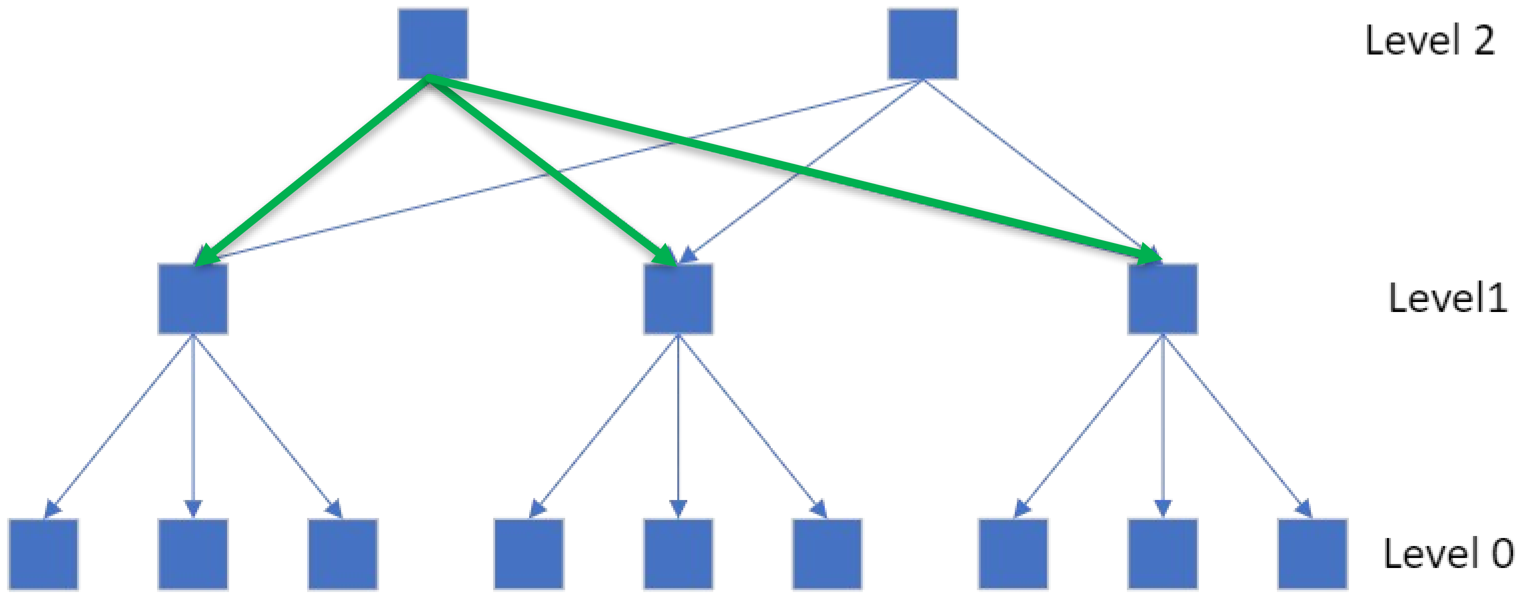
h: Height of the fat tree

m_1,…,$m_{h:}$ Number of children at level i

w_1,…,$w_{h:}$ Number of parents at level i

XGFT(2,3,3,1,2)

XGFT



XGFT (h,m_1,…,m_h,w_1,…,w_h)

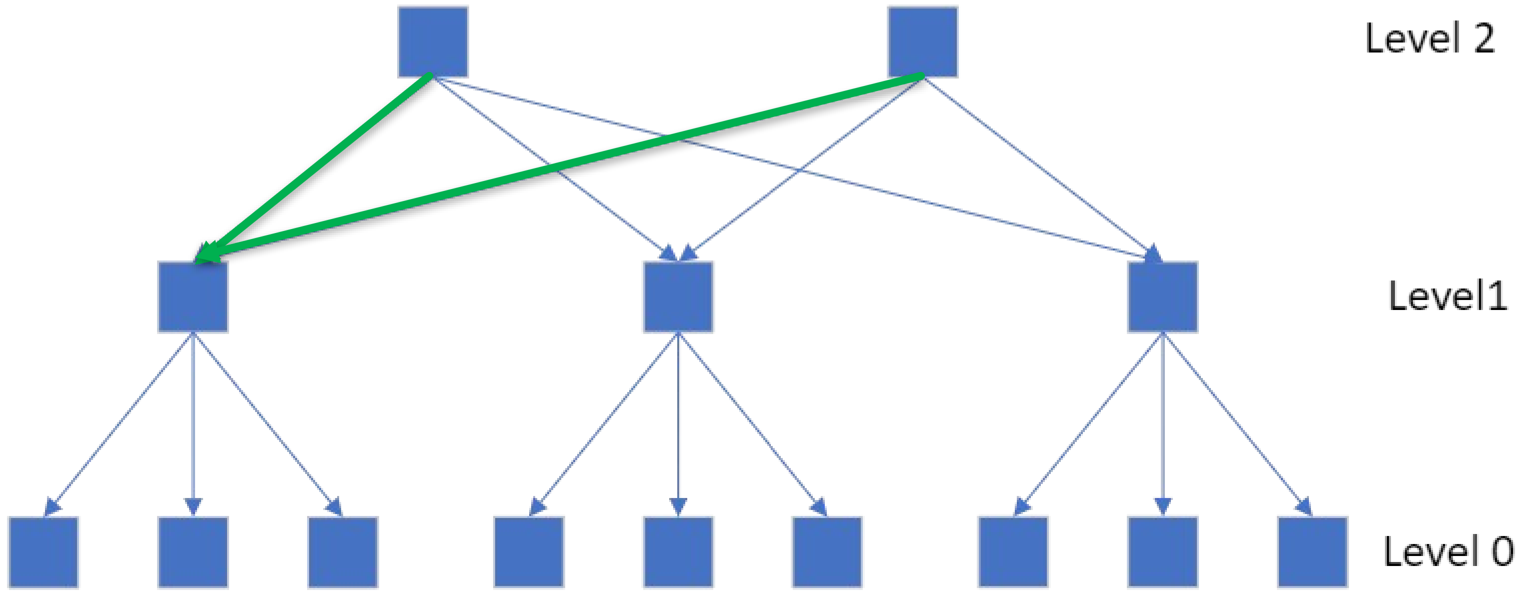
h: Height of the fat tree

m_1,…,$m_{h:}$ Number of children at level i

w_1,…,$w_{h:}$ Number of parents at level i

XGFT(2,3,3,1,2)

XGFT



XGFT (h,m_1,…,m_h,w_1,…,w_h)

h: Height of the fat tree

m_1,…,$m_{h:}$ Number of children at level i

w_1,…,$w_{h:}$ Number of parents at level i

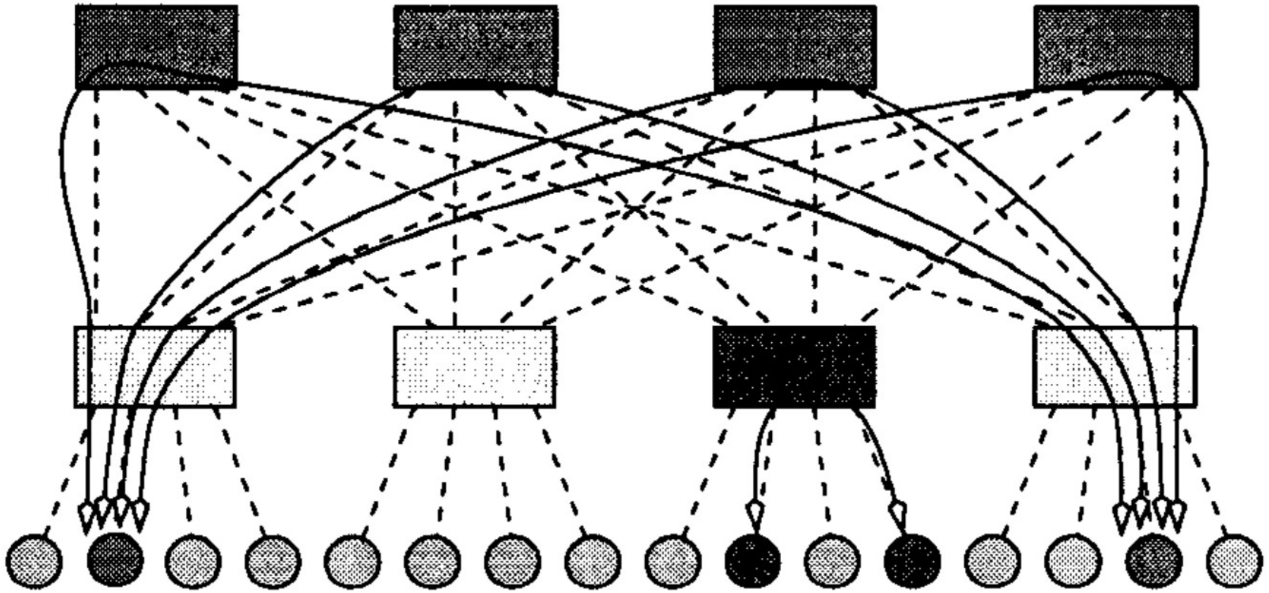
XGFT(2,3,3,1,2)

K-ary n-tree

Each end node is a unique n-tuple {0,1,…,k-1} ^{n}
A router is defined as (w,l). w is a (n-1)-tuple {0,1,…,k-1}$^{n-1}$. l = {0,1,…,n-1}.

Two routers (w^a,l^a) and (w^b,l^b) are connected if l^b=l^a+1 and w_i^a=w_i^b for i ≠ l^a.

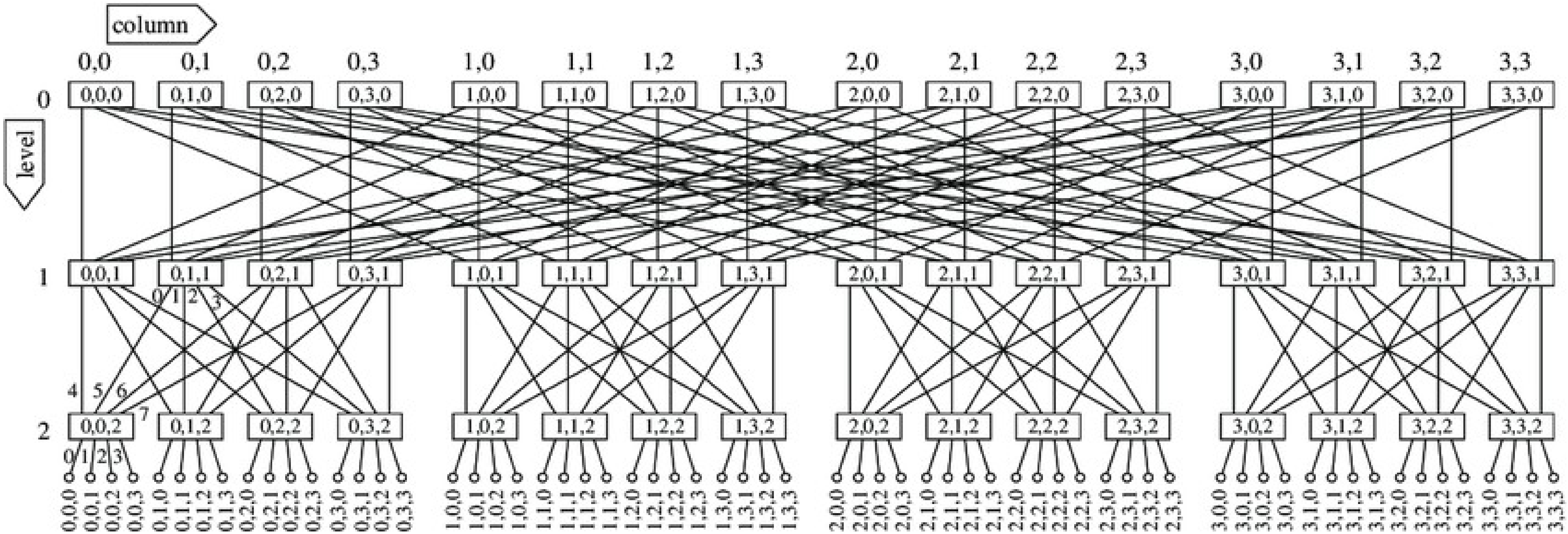
An endpoint is connected to a router (w,n-1), if x_i=w_i.



4-ary 2-tree

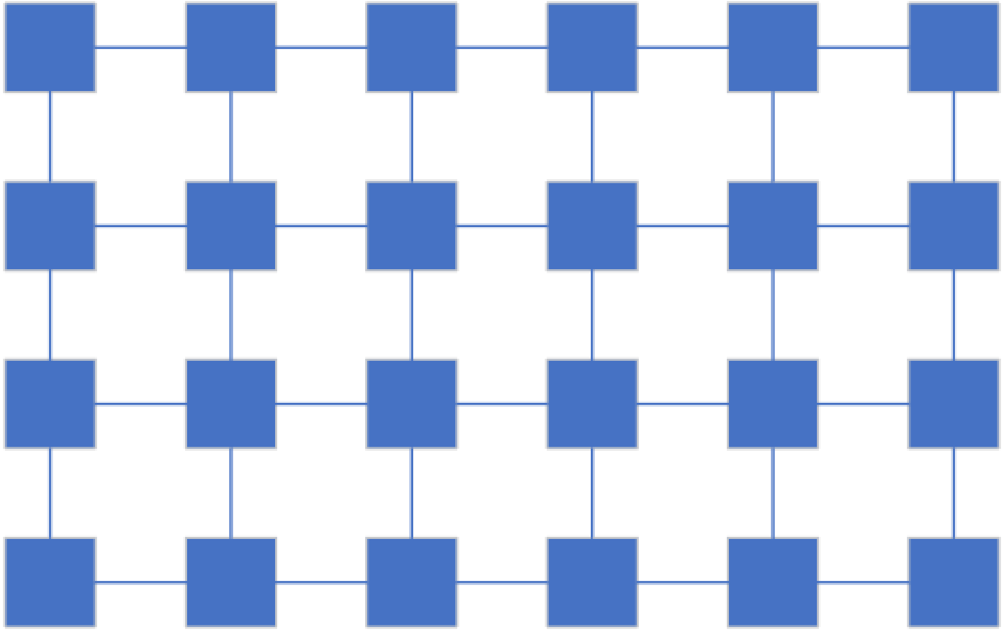
Source: k-ary n-trees: high performance networks for massively parallel architectures. F. Petrini; M. Vanneschi

4-ary 3-tree



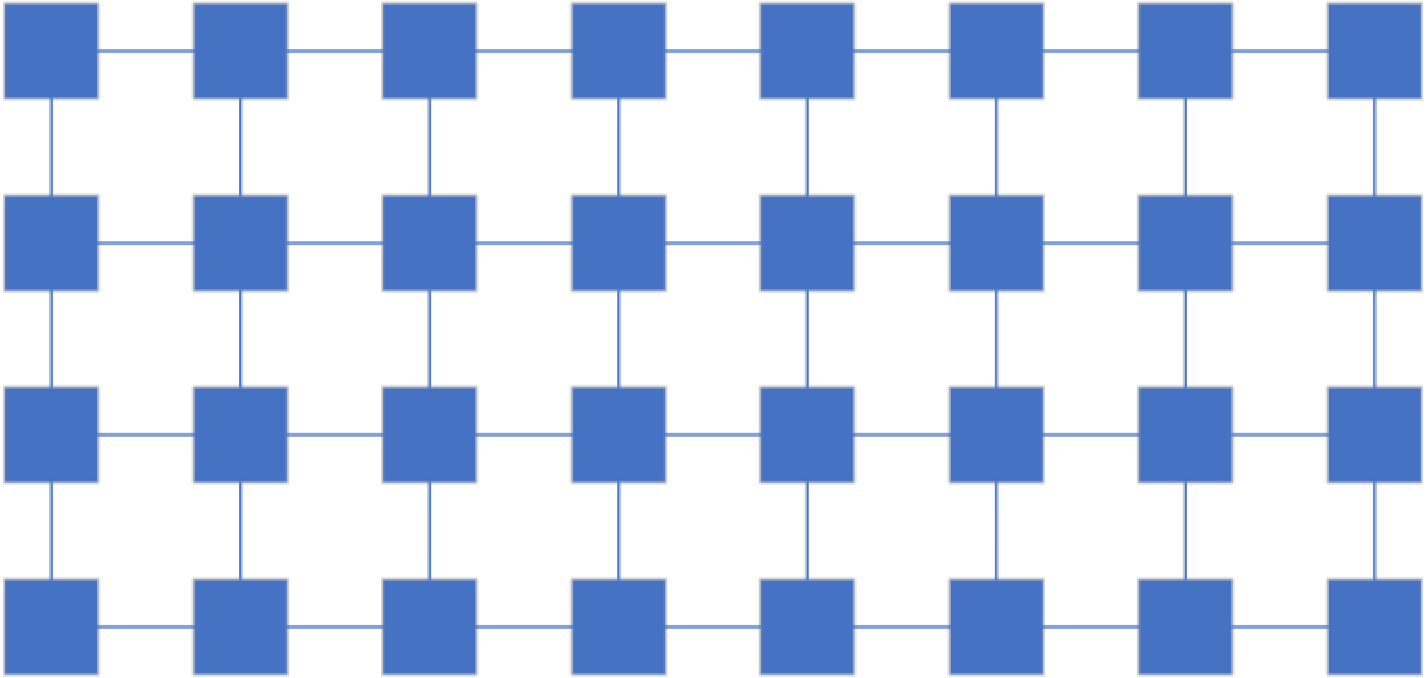
Source: Dynamic power saving in fat-tree interconnection networks using on/off links. Alonso, Marina and Coll, Salvador and Martínez, Juan and Santonja, Vicente and López, Pedro and Duato, José

Mesh



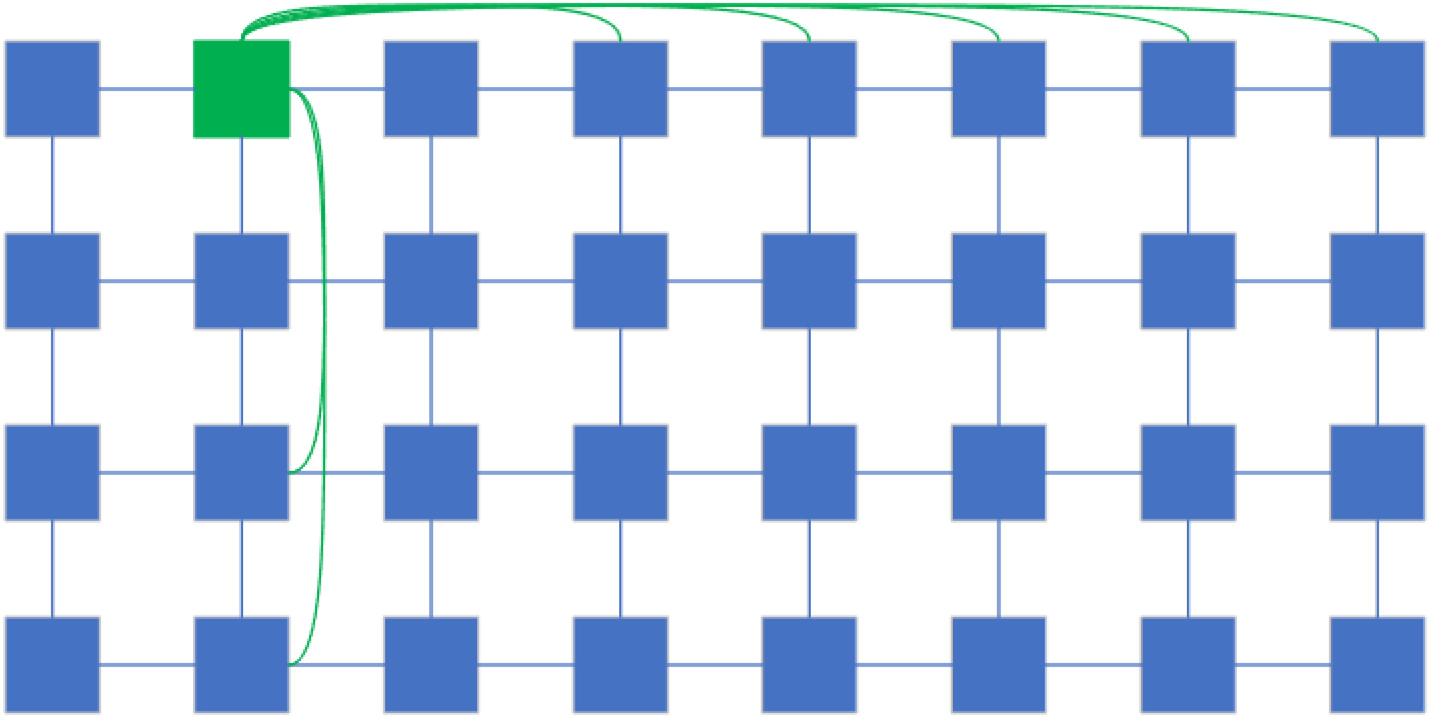
A d-dimensional mesh.
Each node is uniquely defined as $\{x_0,\ldots,x_{d-1}\}$, with $x_i < n$.

Express Mesh



Express connections are added to nodes of a multiple of g distance to original neighbors.

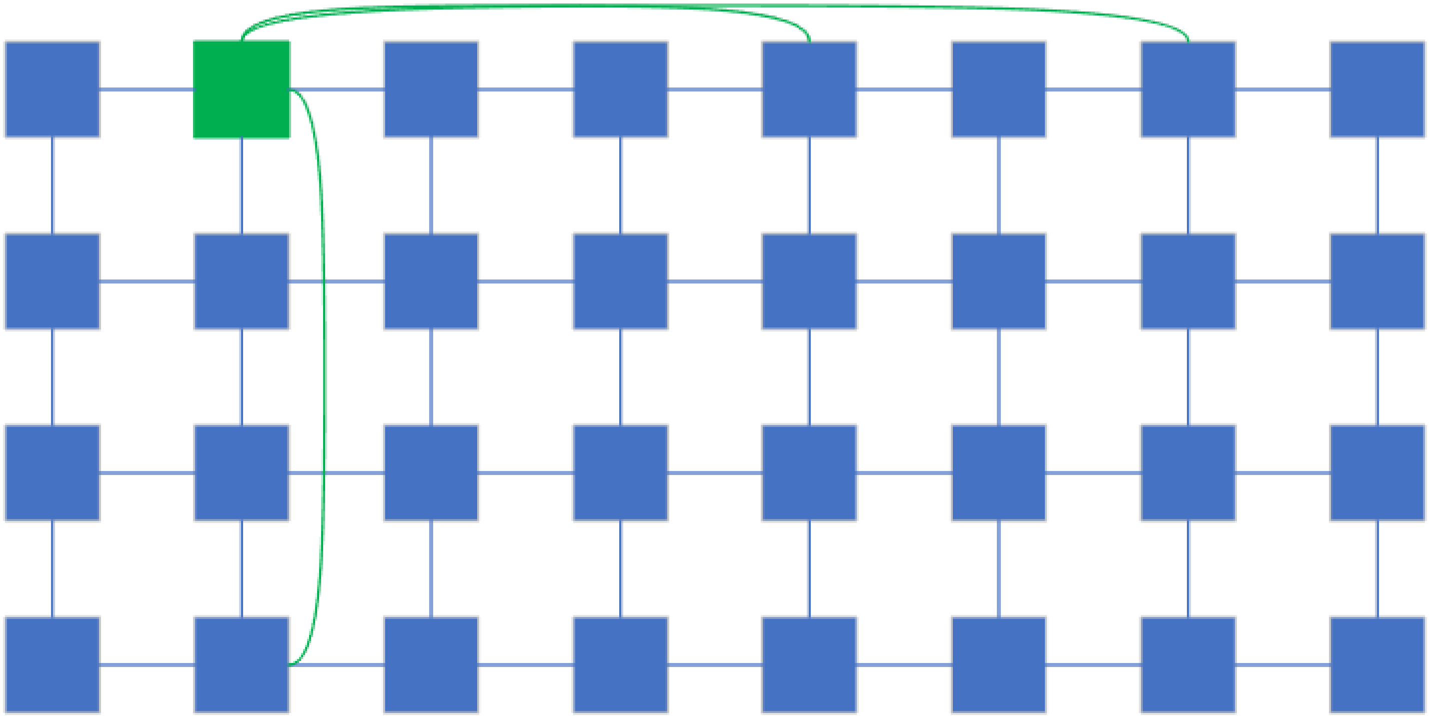
Express Mesh



Express connections are added to nodes of a multiple of g distance to original neighbors.

g = 1

Express Mesh



Express connections are added to nodes of a multiple of g distance to original neighbors.

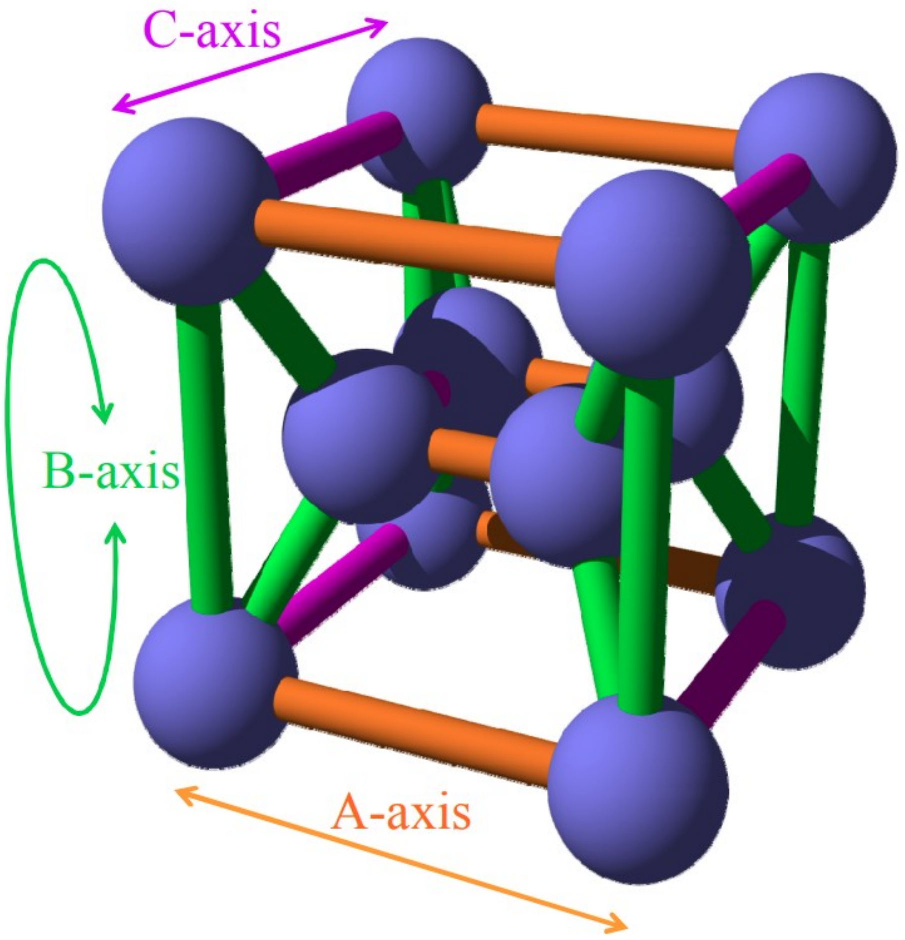
g = 2

Tofu

Variant of a 6-dimensional Torus.
Each Tofu cluster consists of 12 nodes

3-dimensional Torus containing multiple clusters.

Each node is connected its equivalent in a neighboring Tofu cluster



3-dimensional cluster

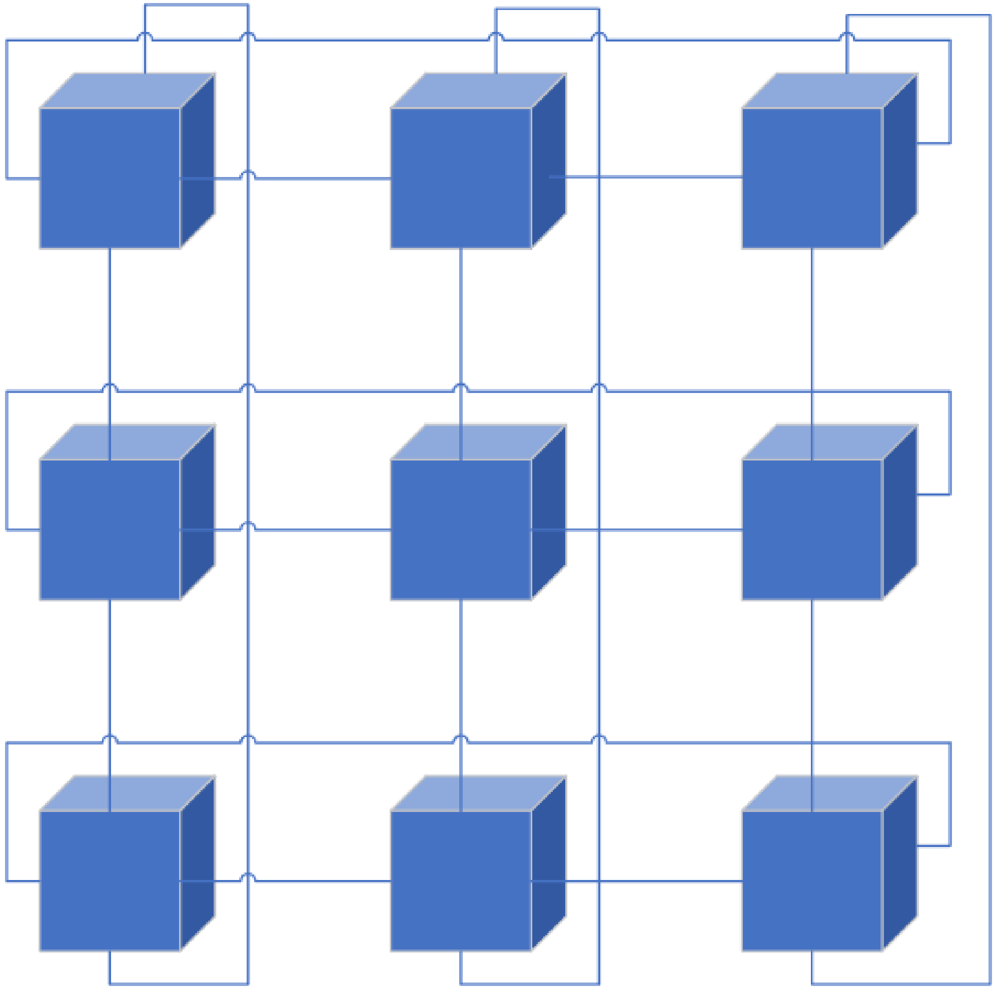
Source: The Tofu Interconnect. Y. Ajima, Y. Takagi, T. Inoue, S. Hiramoto, T. Shimizu

Tofu

Variant of a 6-dimensional Torus.
Each Tofu cluster consists of 12 nodes

3-dimensional Torus containing multiple clusters.

Each node is connected its equivalent in a neighboring Tofu cluster

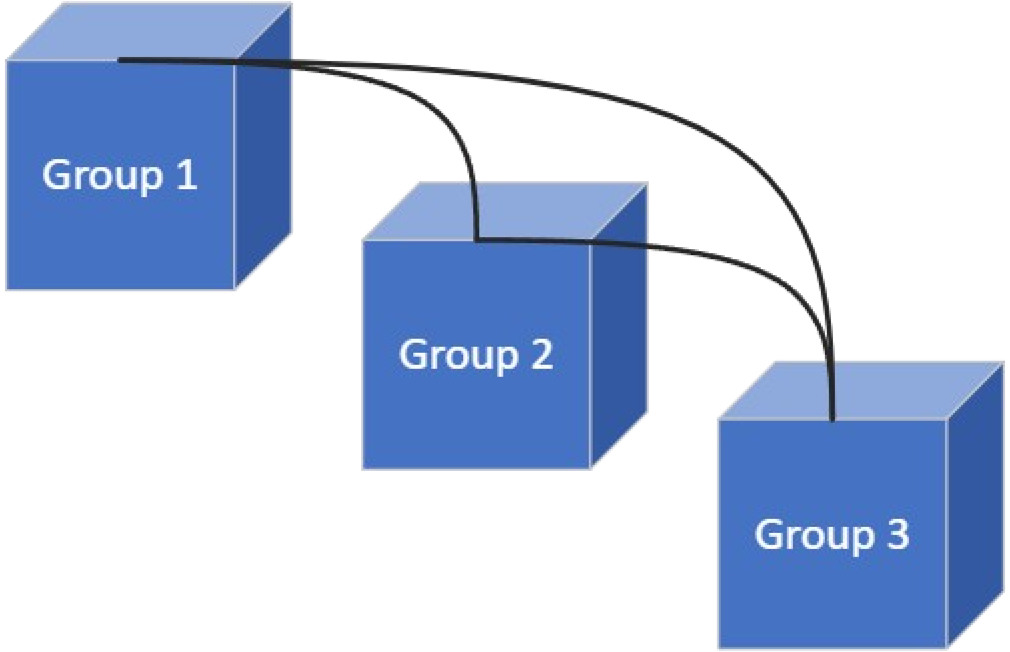


Cascade Dragonfly

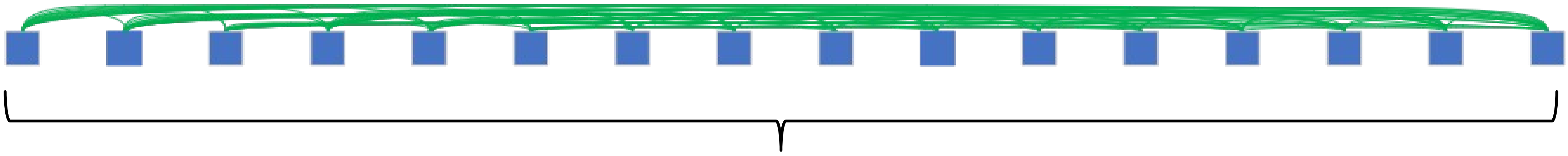
A group is built as a Dragonfly network

Each group is connected via 4 nodes to any other cluster

6 chassis with each 16 Aries routers → 96 routers per group

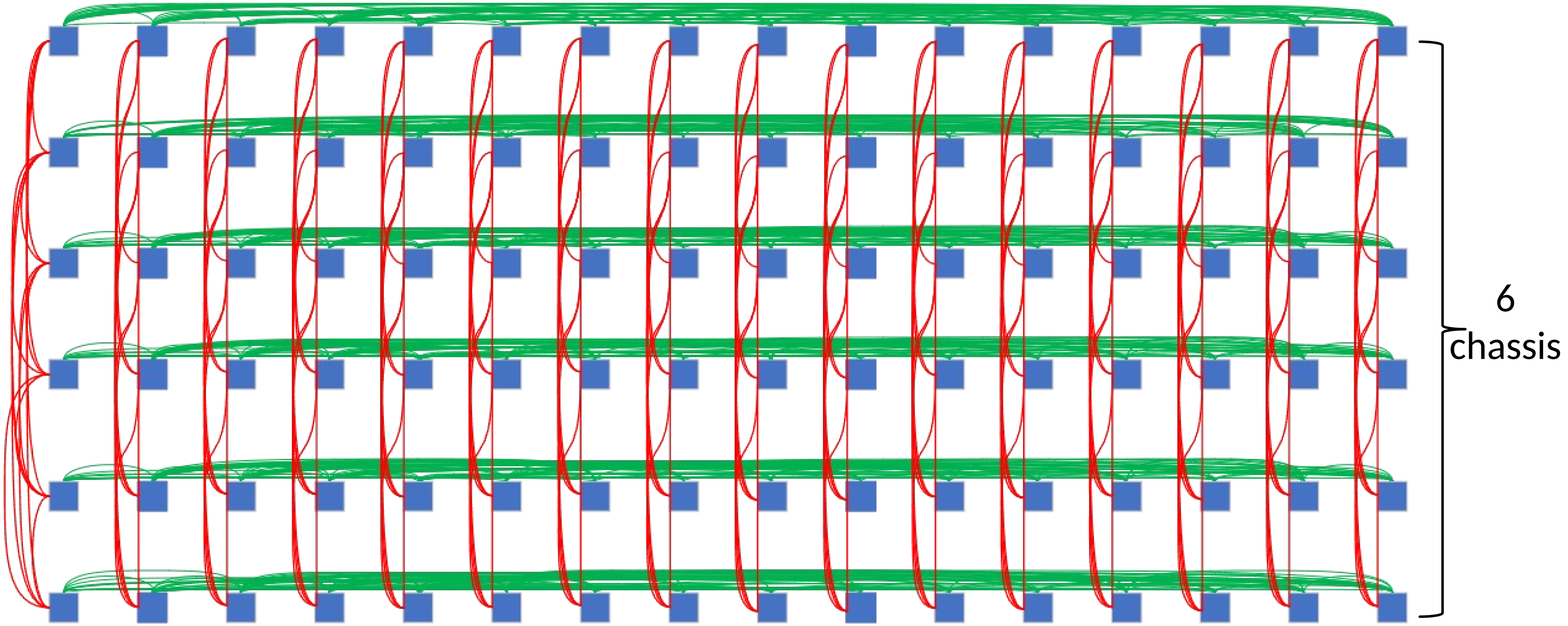


Cascade Dragonfly

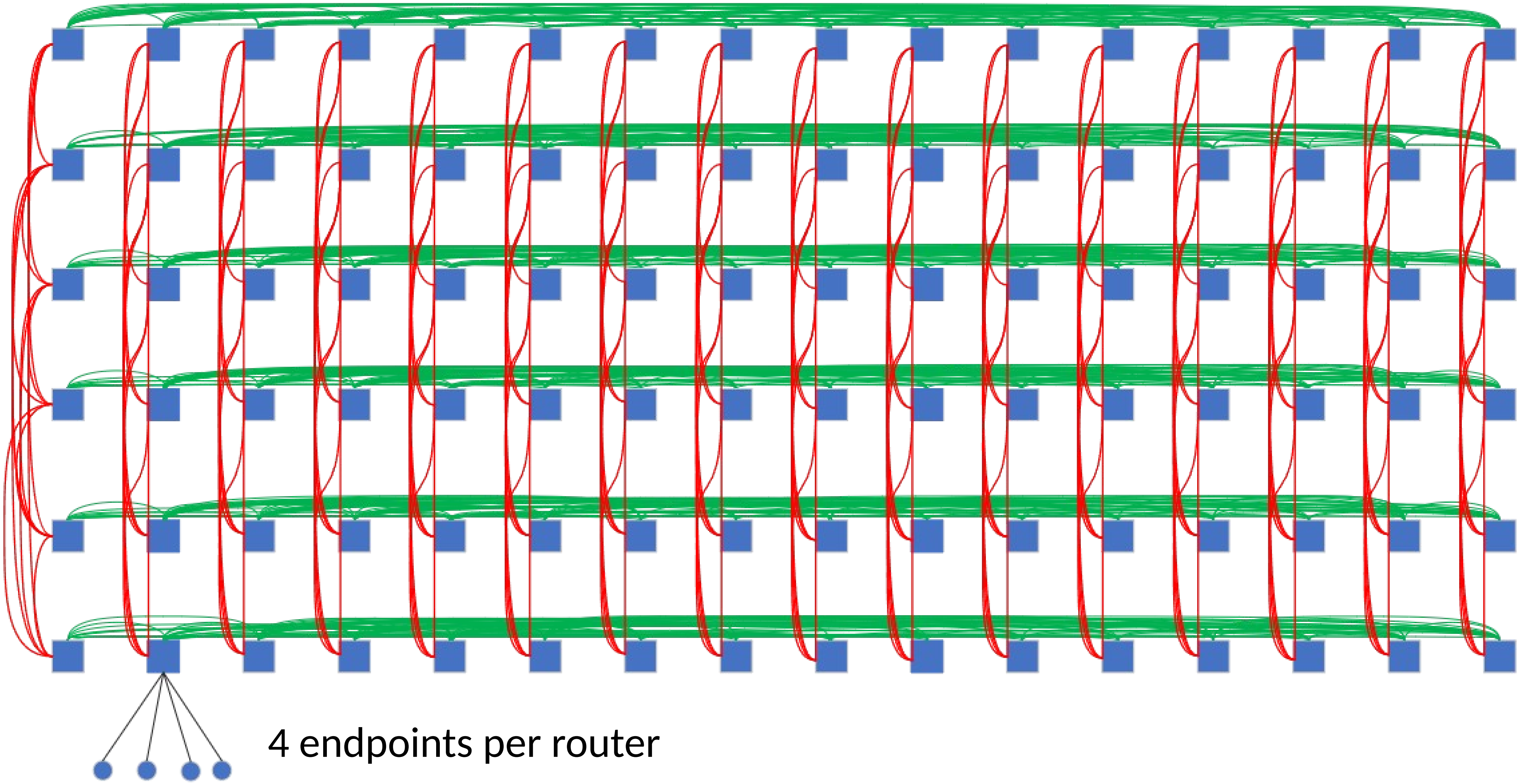


16 Aries router per chassis

Cascade Dragonfly



Cascade Dragonfly



4 endpoints per router

Spectralfly

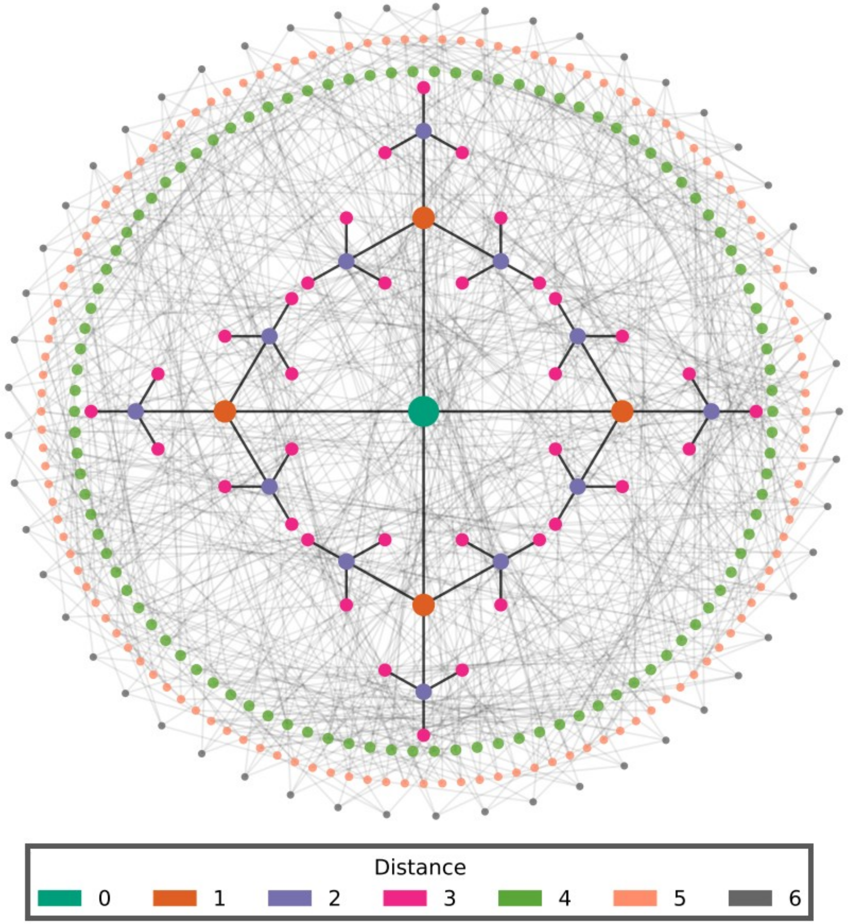
Construction:

v, w are primes

x, y be solutions to $x^2 + y^2 + 1 \equiv_w 0$

$\alpha_0^2 + \alpha_1^2 + \alpha_2^2 + \alpha_3^2 = v$

- $\alpha_0 > 0$ is odd, if $v \equiv_4 1$
- $\alpha_0 > 0$ is even, or $\alpha_0 = 0$ and $\alpha_1 > 0$, if $v \equiv_4 3$



$\text{SpF}_{3,7}$

Source: SpectralFly: Ramanujan Graphs as Flexible and Efficient Interconnection Networks. S. Young, S. Aksoy, J. Firoz, R Gioiosa, T. Hagge, M. Kempton, J. Escobedo, M. Raugas

Spectralfly

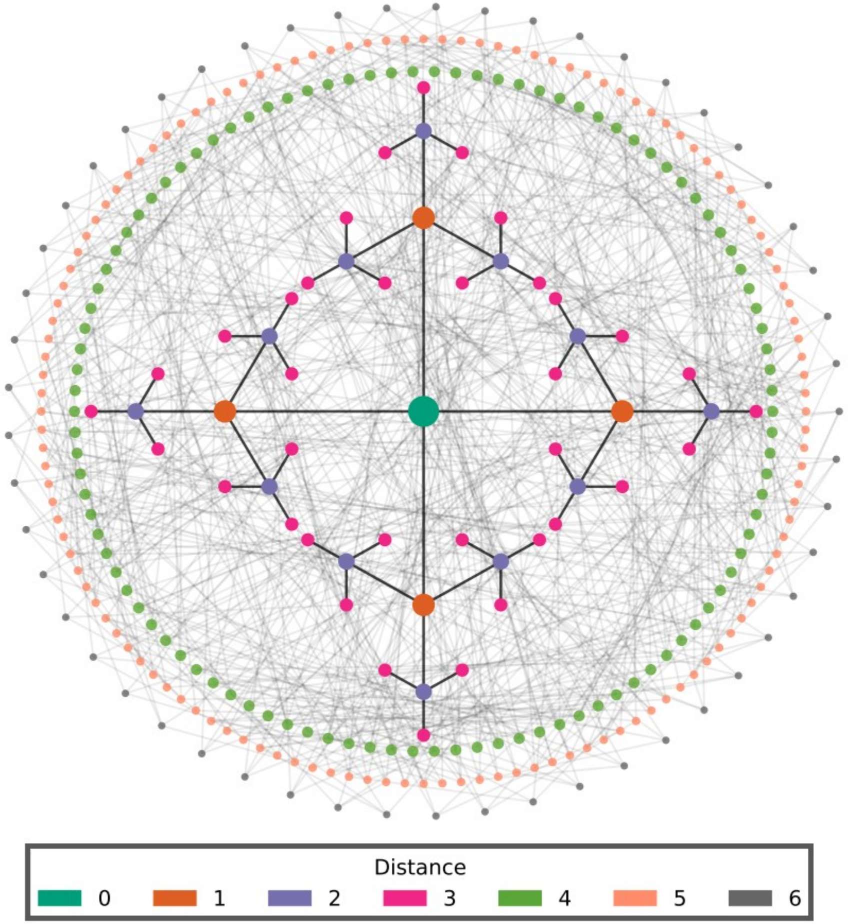
Construction:

Generating set S of SpF(v,w):

$$\begin{bmatrix} a_0 + xa_1 + ya_3 & -ya_1 + a_2 + xa_3 \\ -ya_1 - a_2 + xa_3 & a_0 - xa_1 - ya_3 \end{bmatrix}$$

There is an edge {u,v} if $u^{-1}v$ in S

etc…



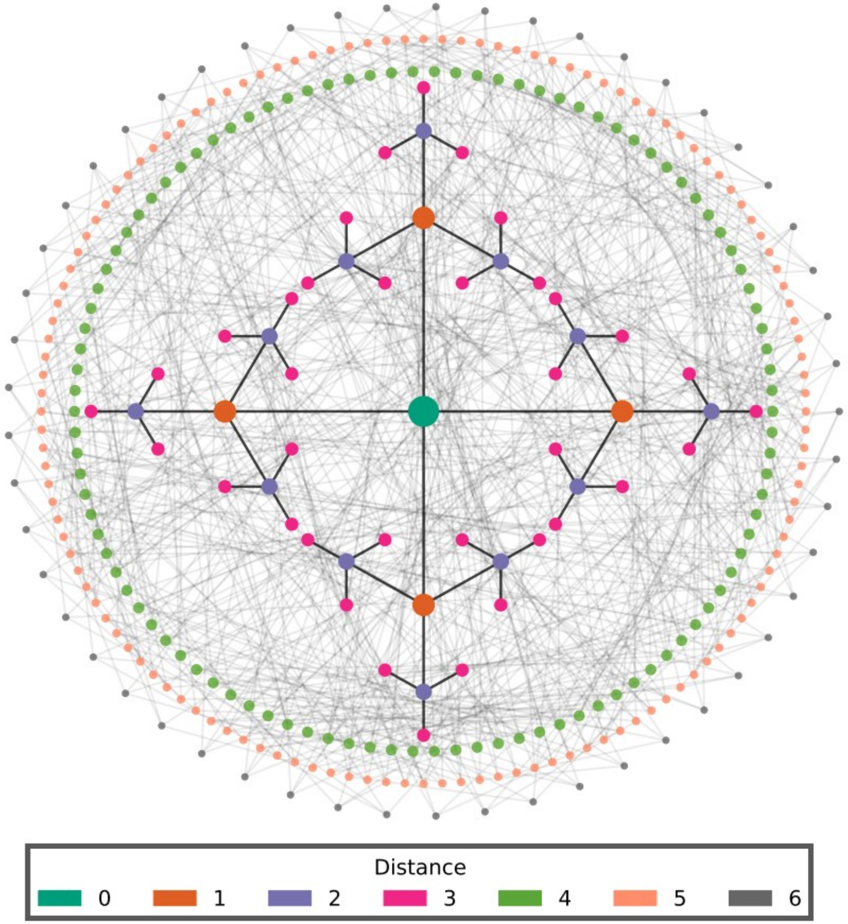
SpF$_{3,7}$

Source: SpectralFly: Ramanujan Graphs as Flexible and Efficient Interconnection Networks. S. Young, S. Aksoy, J. Firoz, R Gioiosa, T. Hagge, M. Kempton, J. Escobedo, M. Raugas

Spectralfly

SpectralFly: Ramanujan Graphs as Flexible and Efficient Interconnection Networks
S. Young, S. Aksoy, J. Firoz, R Gioiosa, T. Hagge, M. Kempton, J. Escobedo, M. Raugas

Elementary number theory, group theory and Ramanujan graphs
G. Davidoff, P. Sarnak, and A. Valette



SpF$_{3,7}$

Source: SpectralFly: Ramanujan Graphs as Flexible and Efficient Interconnection Networks. S. Young, S. Aksoy, J. Firoz, R Gioiosa, T. Hagge, M. Kempton, J. Escobedo, M. Raugas

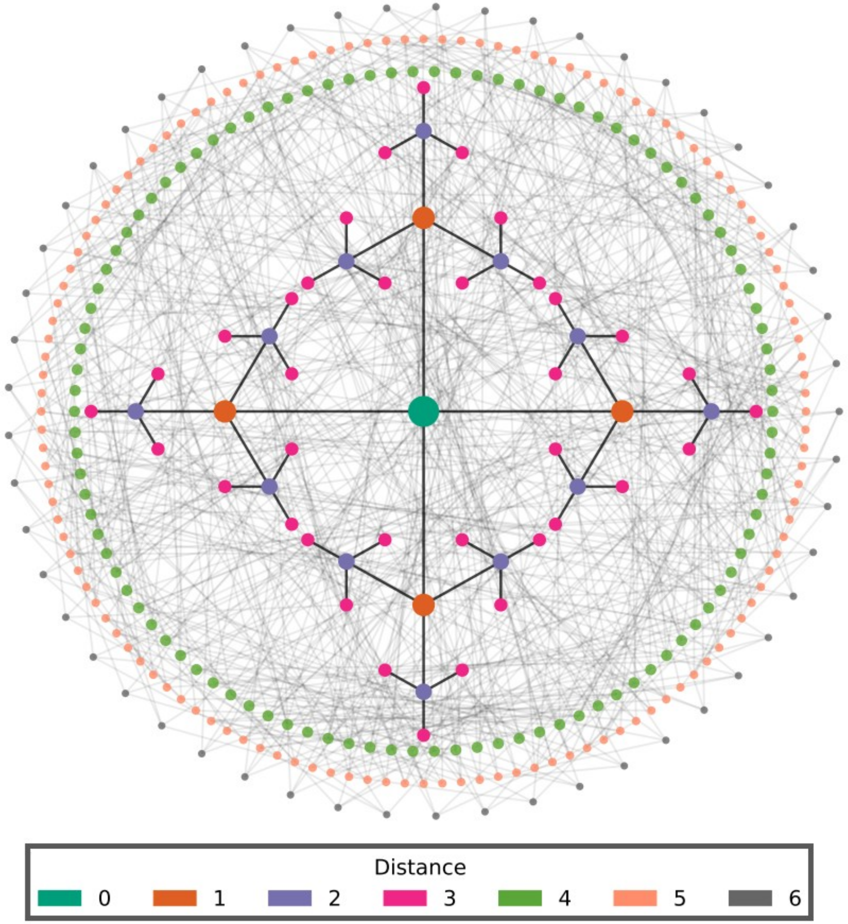
Spectralfly

Definition: A k-regular graph G is called Ramanujan if , where

$$\lambda(G) \leq 2 \times \sqrt{k-1},$$

denotes the largest magnitude adjacency eigenvalue of G not equal to ±k

If *w>2*sqrt(v*), then SpF is a (v+1)-regular Ramanujan graph



$\text{SpF}_{3,7}$

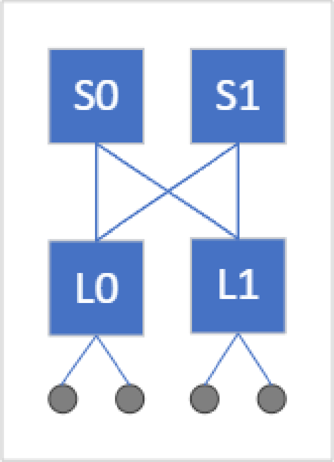
Source: SpectralFly: Ramanujan Graphs as Flexible and Efficient Interconnection Networks. S. Young, S. Aksoy, J. Firoz, R Gioiosa, T. Hagge, M. Kempton, J. Escobedo, M. Raugas

Megafly

Spine and leaf nodes $s = l = d/2$

Each spine router has s/g global links

Total of $s^2/g + 1$ groups

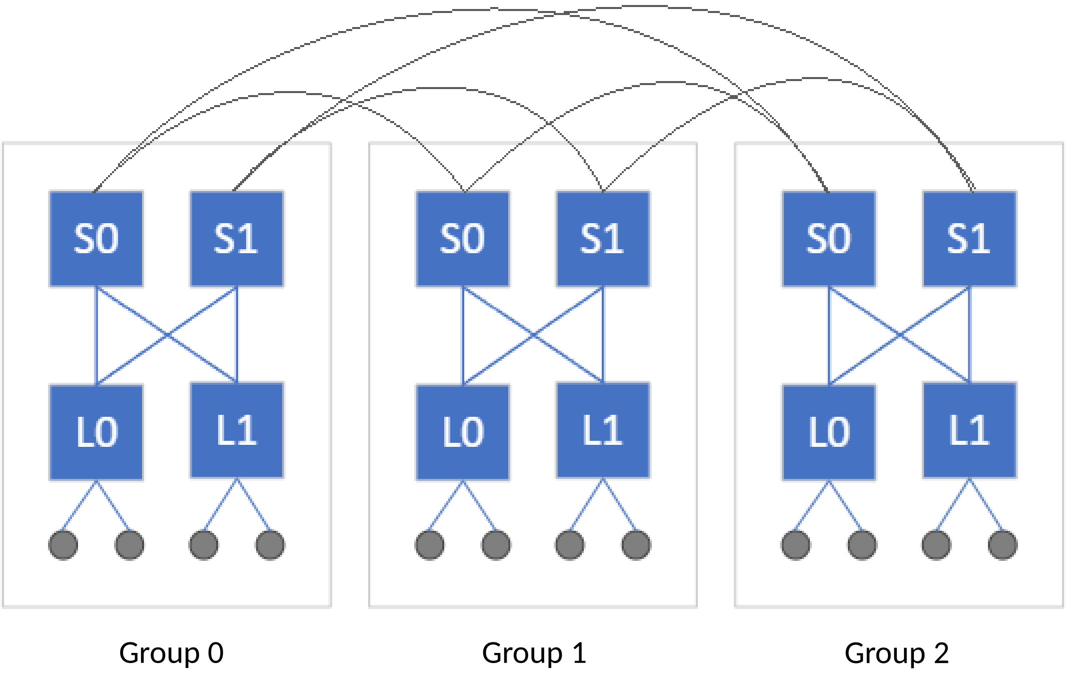


Megafly

Spine and leaf nodes *s* = *l* = *d/2*

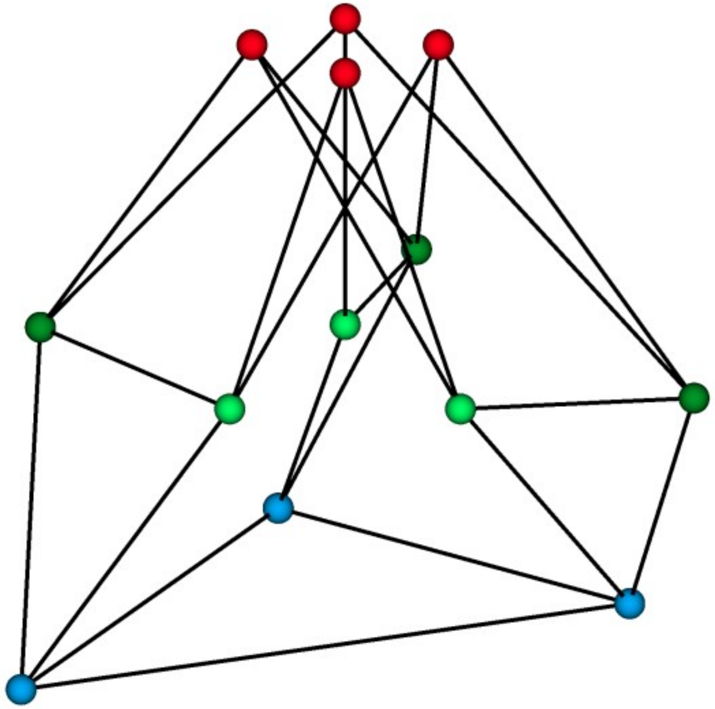
Each spine router has *s/g* global links

Total of $s^2/g + 1$ groups



$\mathrm{MF}_{2,4}$

Polarstar



Structure graph G: ER_3

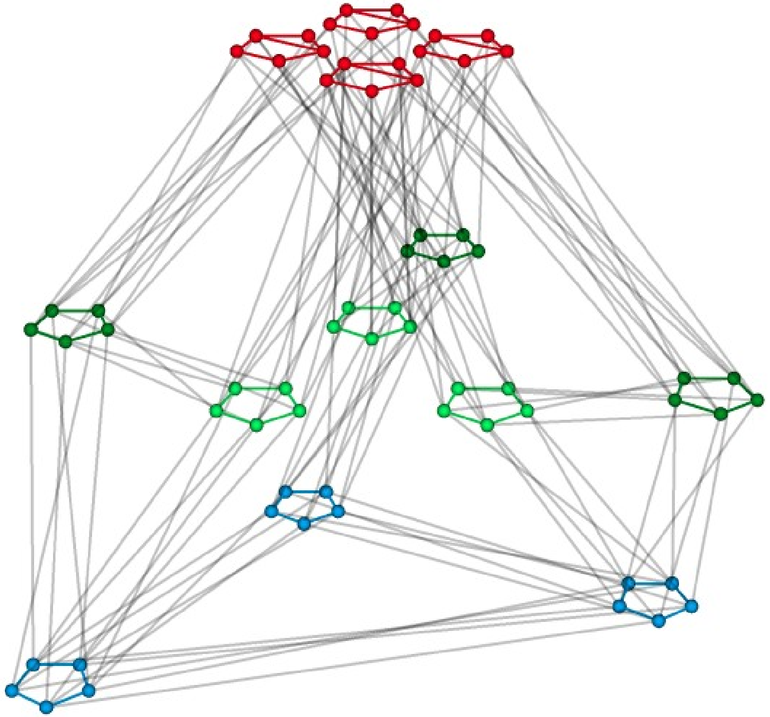
Source: PolarStar: Expanding the Scalability Horizon of Diameter-3 Networks. K. Lakhotia, L. Monroe, K. Isham, M. Besta, N. Blach, T. Hoefler, F. Petrini

Starproduct

Structure graph G is an ER graph

Subgraph either *BDF* or *Paley*

Polarstar



$G{*}G'\colon ER_3 * Paley(5)$

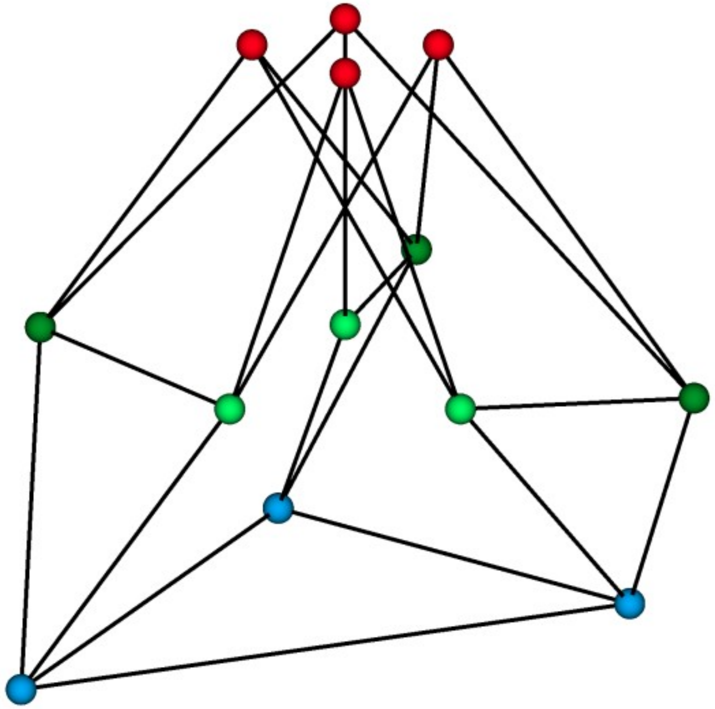
Source: PolarStar: Expanding the Scalability Horizon of Diameter-3 Networks. K. Lakhotia, L. Monroe, K. Isham, M. Besta, N. Blach, T. Hoefler, F. Petrini

Starproduct

Structure graph G is an ER graph

Subgraph either *BDF* or *Paley*

Polarstar



Structure graph G: ER_3

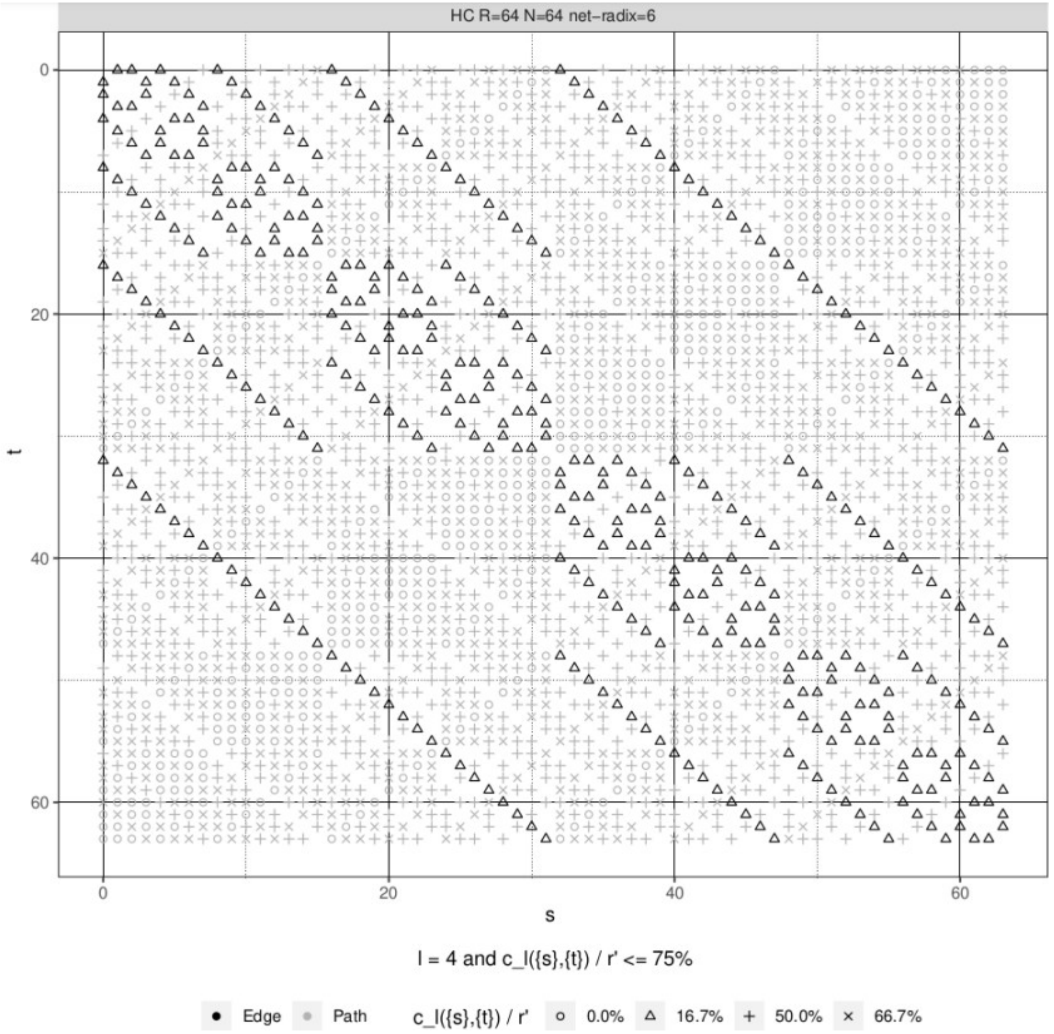
Source: PolarStar: Expanding the Scalability Horizon of Diameter-3 Networks. K. Lakhotia, L. Monroe, K. Isham, M. Besta, N. Blach, T. Hoefler, F. Petrini

Starproduct

Structure graph G is an ER graph

Subgraph either *BDF* or *Paley*

Previous Toolchain



Source: Facilitating design, analysis, and evaluation of network topologies.
Alessandro Maissen