

MIKAN

**A Mirrored K-Ary N-Tree for Reducing Hardware Cost
and Packet Latency of Fat-Tree and Clos Networks**

Yamin Li

Dept. of Computer Science

Hosei University

<http://yamin.cis.k.hosei.ac.jp/>

Motivations

- The interconnection network (IN) is the key part of supercomputers (www.top500.org)
- The supercomputer scales up year by year
 - The cost of INs becomes high
- Most popular INs and their disadvantages
 - **Fat-tree**: Not symmetric (root switches have fewer ports)
 - **Torus**: Long diameter and average distance
 - **Dragonfly**: Complex routing algorithm
- We focus on fat-tree and Clos networks
 - Fat-tree is a folded version of a Clos network

Motivations

- *A Clos or folded Clos network has a high cost and a high latency.*
- *The increased cost and latency both stem from the need to route packets first to an arbitrary middle stage switch and then to their ultimate destination.*
- *This doubles the number of long cables in the network, which approximately doubles cost, and doubles the number of inter-router channels traversed, which drives up latency.*

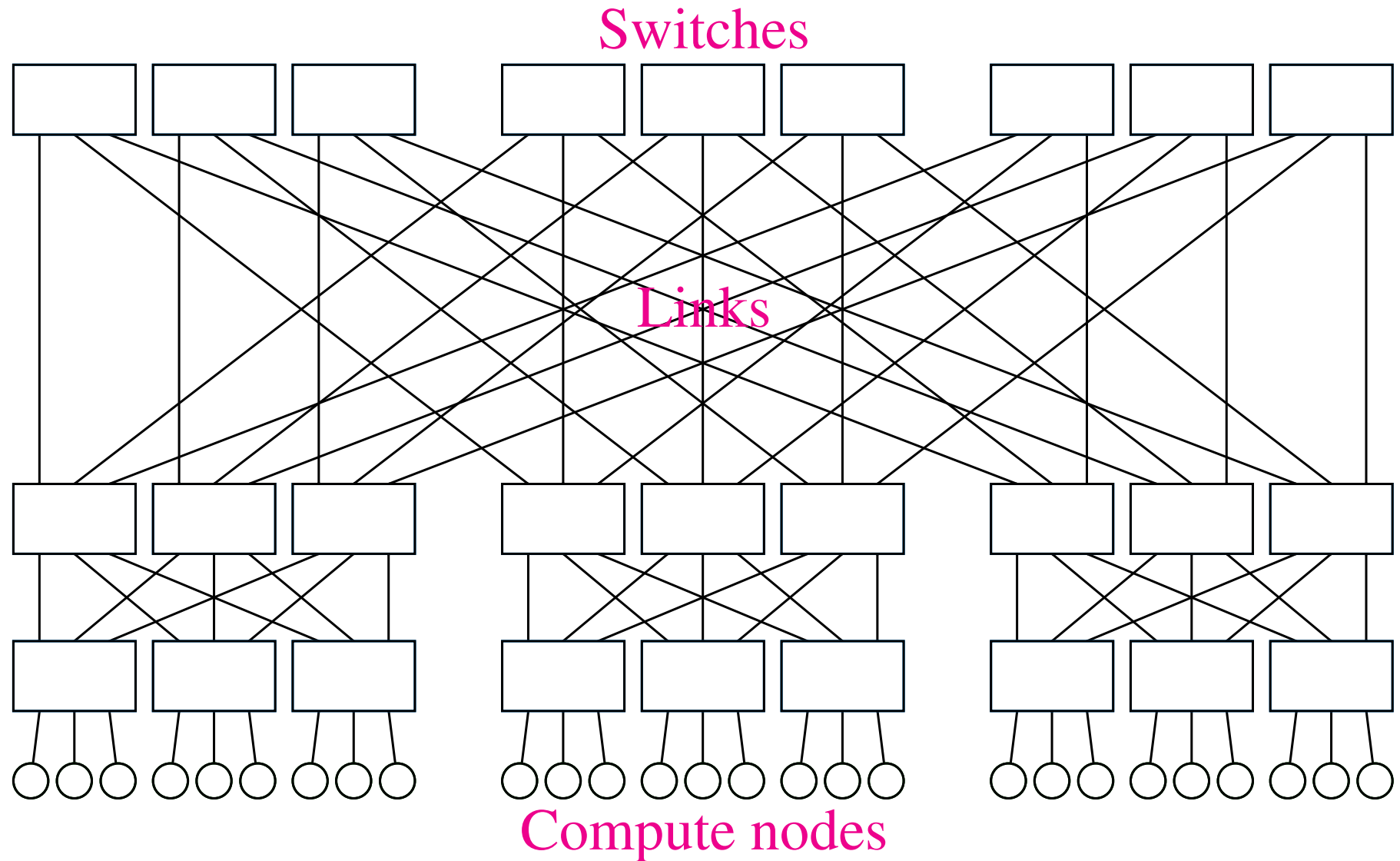
Ref.: D. Abts and J. Kim, *High Performance Datacenter Networks: Architectures, Algorithms, and Opportunities*. Morgan and Claypool Publishers, Mar. 2011.

<https://doi.org/10.2200/S00341ED1V01Y201103CAC014>

Motivations

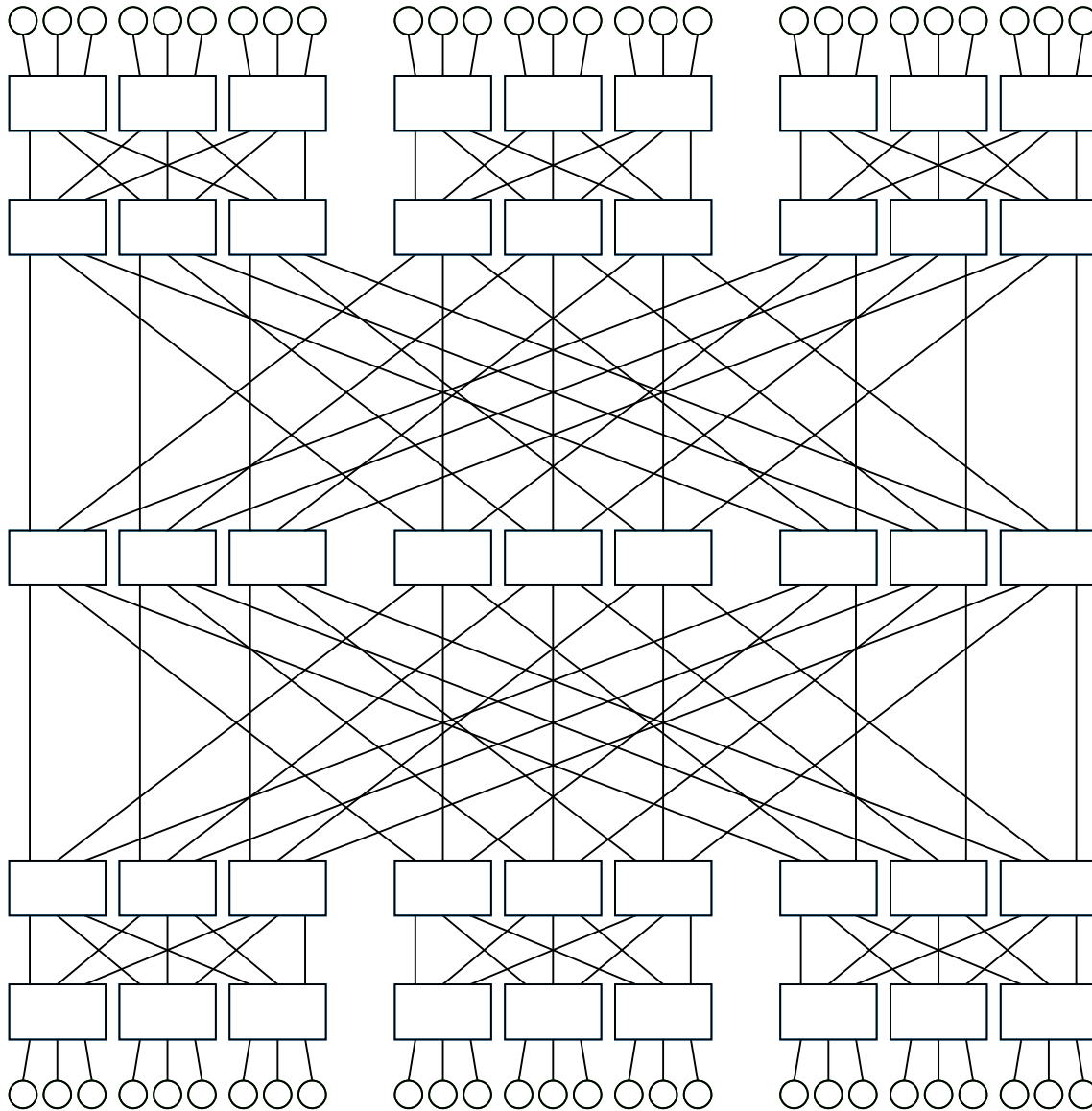
- We propose a **MiKANT** - **M**irrored **K**-**A**ry **N**-**T**ree
 - Symmetric (all switches have the same radix)
 - Low hardware (switch and link) cost
 - Short average distance
 - Short packet latency
 - Multiple paths from source to destination
 - for efficient fault tolerant routing
- We can consider the MiKANT as an alternative to the fat-tree or Clos network

Fat Tree (3-ary 3-tree)



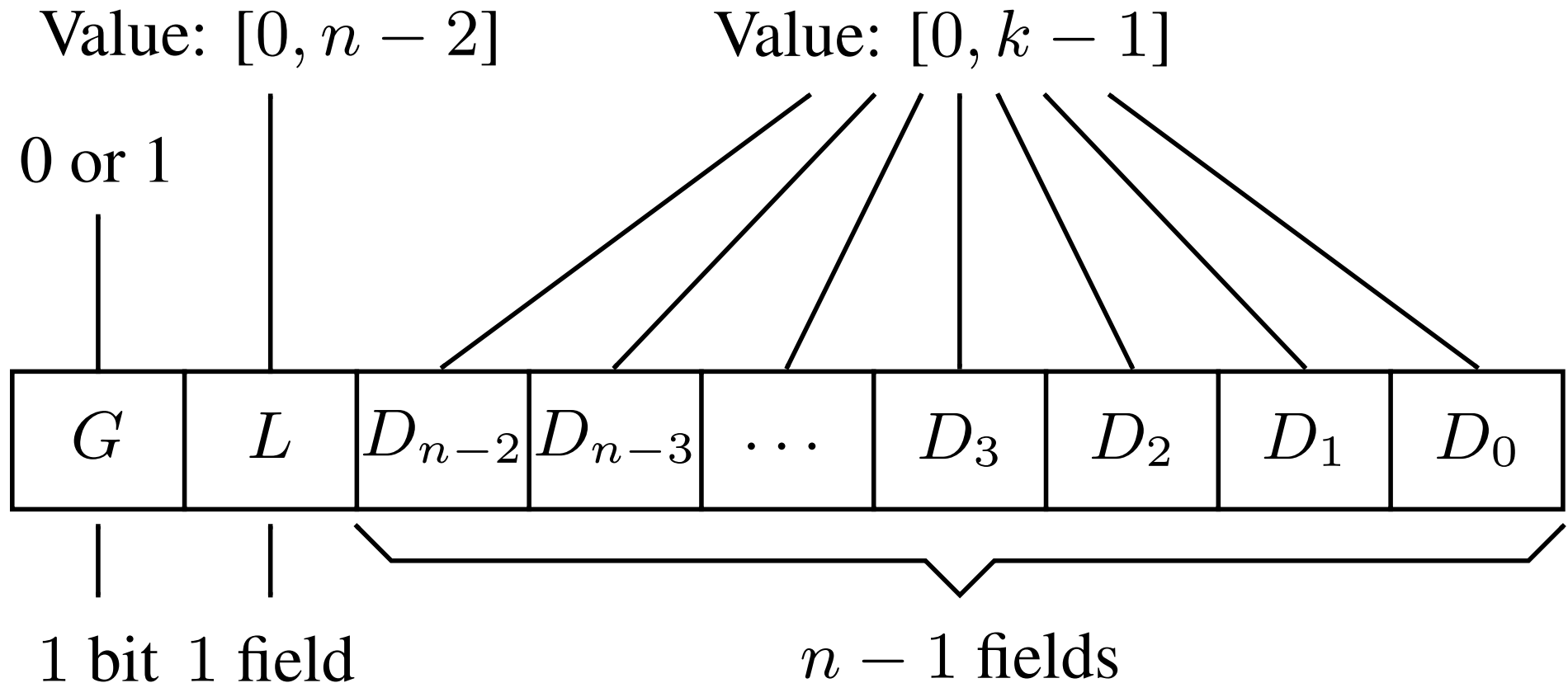
Root switches have fewer ports than others

Bidirectional Clos Network



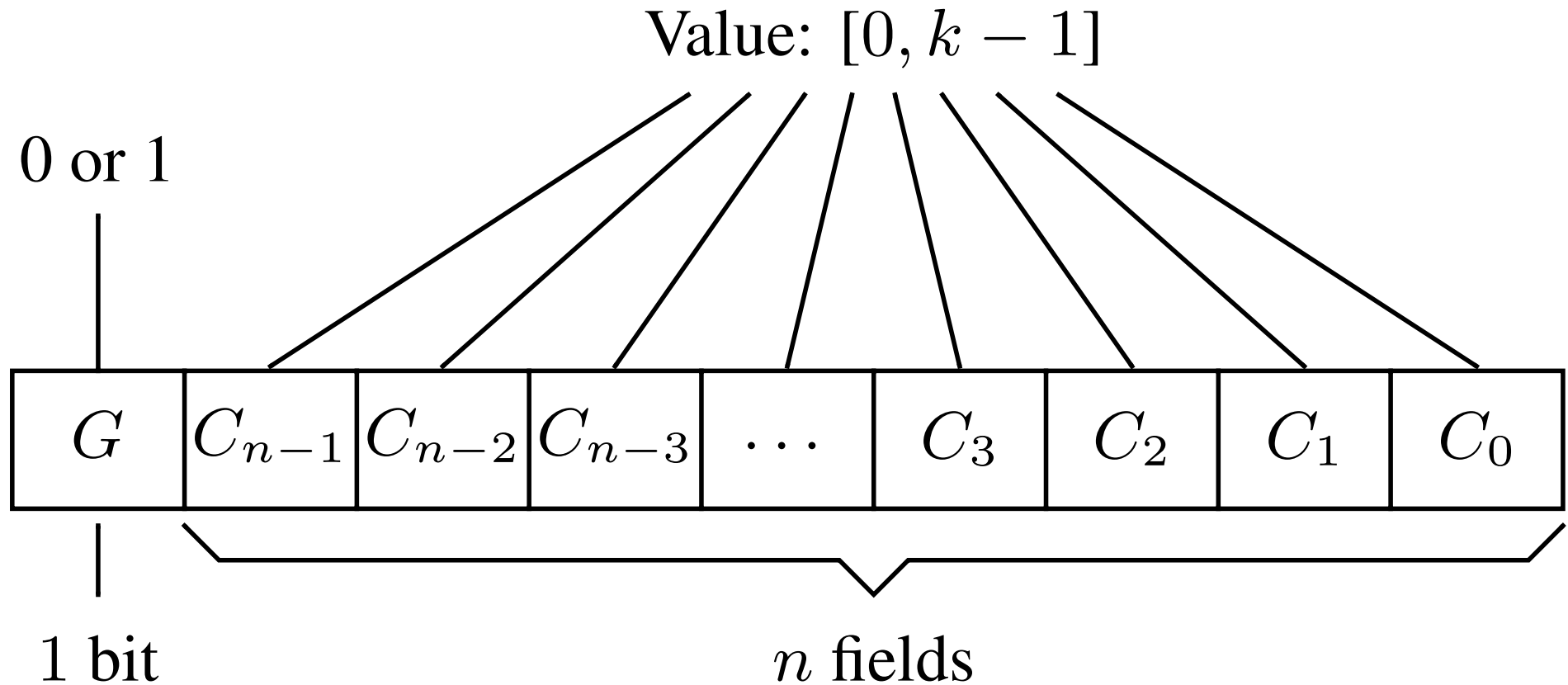
High switch cost
High link cost
Long distance

Switch Address Format of MiKANT(n, k)



G : Group ID; L : Level (Stage) ID; D : Switch ID

Node Address Format of MiKANT(n, k)



G : Group ID; C : Node ID; $C_{n-2} \dots C_0$: Switch ID

Switch-Switch Connections

A switch

$$\langle G, L, D_{n-2}, \dots, D_{L+1}, D_L, D_{L-1}, \dots, D_0 \rangle$$

will connect to switches

$$\langle G, L + 1, D_{n-2}, \dots, D_{L+1}, *, D_{L-1}, \dots, D_0 \rangle$$

if $0 \leq L \leq n - 3$; otherwise ($L = n - 2$) to switches

$$\langle \overline{G}, L, *, D_{n-3}, \dots, D_1, D_0 \rangle$$

Switch-Compute Node Connections

There is a link between a switch

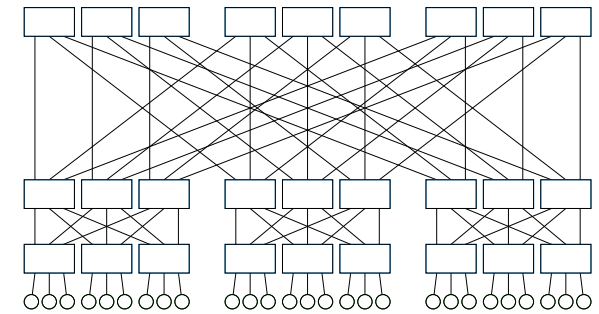
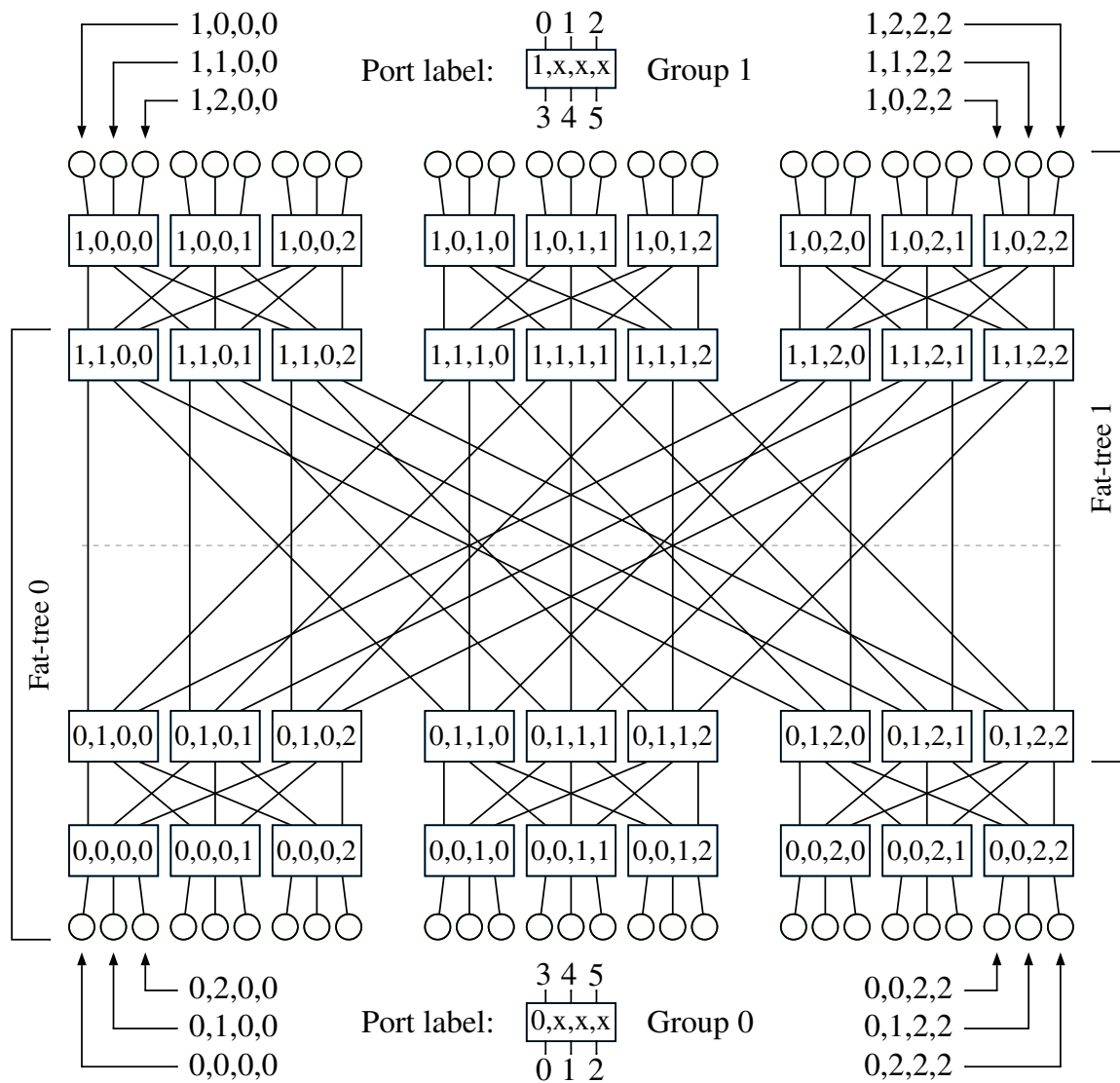
$$\langle G, 0, D_{n-2}, \dots, D_1, D_0 \rangle$$

and a compute node

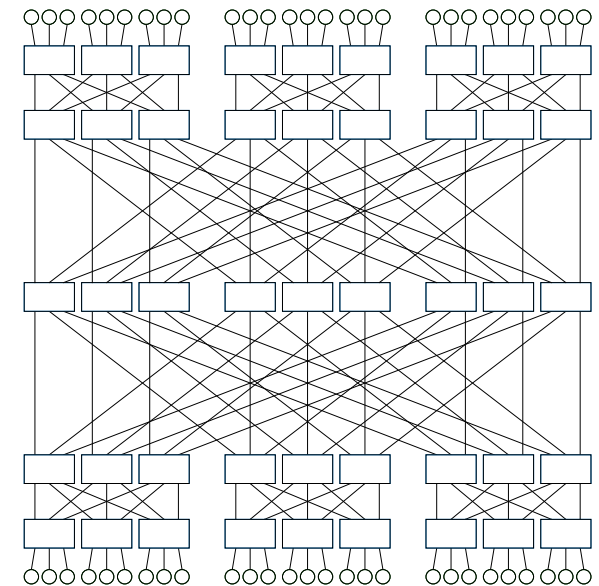
$$\langle G, C_{n-1}, C_{n-2}, \dots, C_1, C_0 \rangle$$

if $D_i = C_i$ for all $i \in \{n-2, \dots, 1, 0\}$.

MiKANT(3,3)

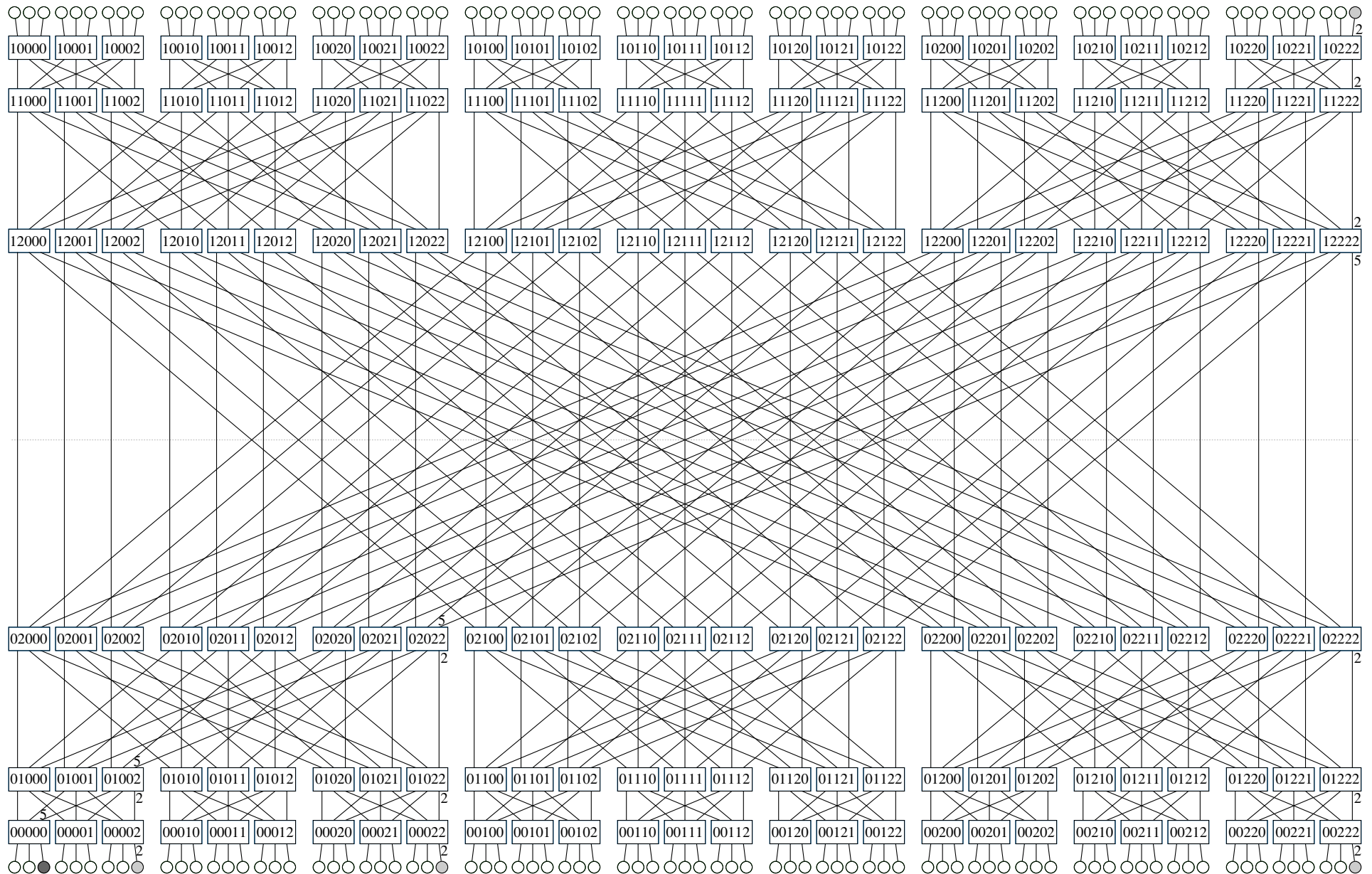


3-ary 3-tree



Clos 3-ary 3-tree

MiKANT(3,4)



Comparison of Topological Properties

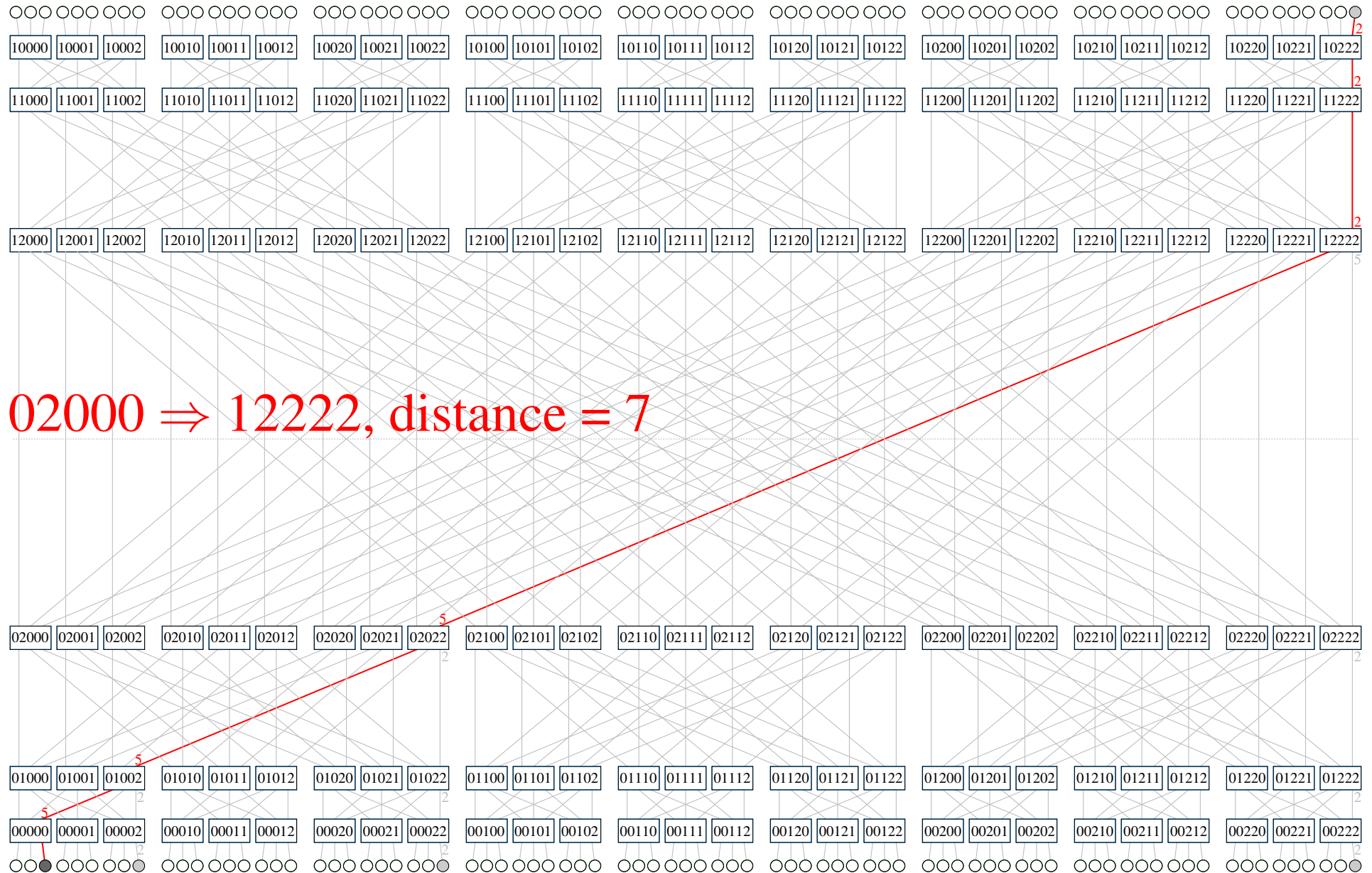
	Classical k -ary n -tree	Clos k -ary n -tree	Mirrored k -ary n -tree
#nodes	k^n	$2k^n$	$2k^n$
#switches	nk^{n-1}	$(2n-1)k^{n-1}$	$(2n-2)k^{n-1}$
#links	nk^n	$2nk^n$	$(2n-1)k^n$
Radix	$2k$	$2k$	$2k$
Diameter	$2n$	$2n$	$2n$
Bisection	$k^n/2$	$k^n/2$	$k^n/2$
Ave. dist	$2n - \frac{2}{k-1} + \frac{2}{(k-1)k^n}$	$2n - \frac{1}{k-1} + \frac{1}{(k-1)k^n}$	$2n - \frac{1}{k-1} + \frac{1}{(k-1)k^n} - \frac{1}{2}$

Minimal Per-Hop Deterministic Routing

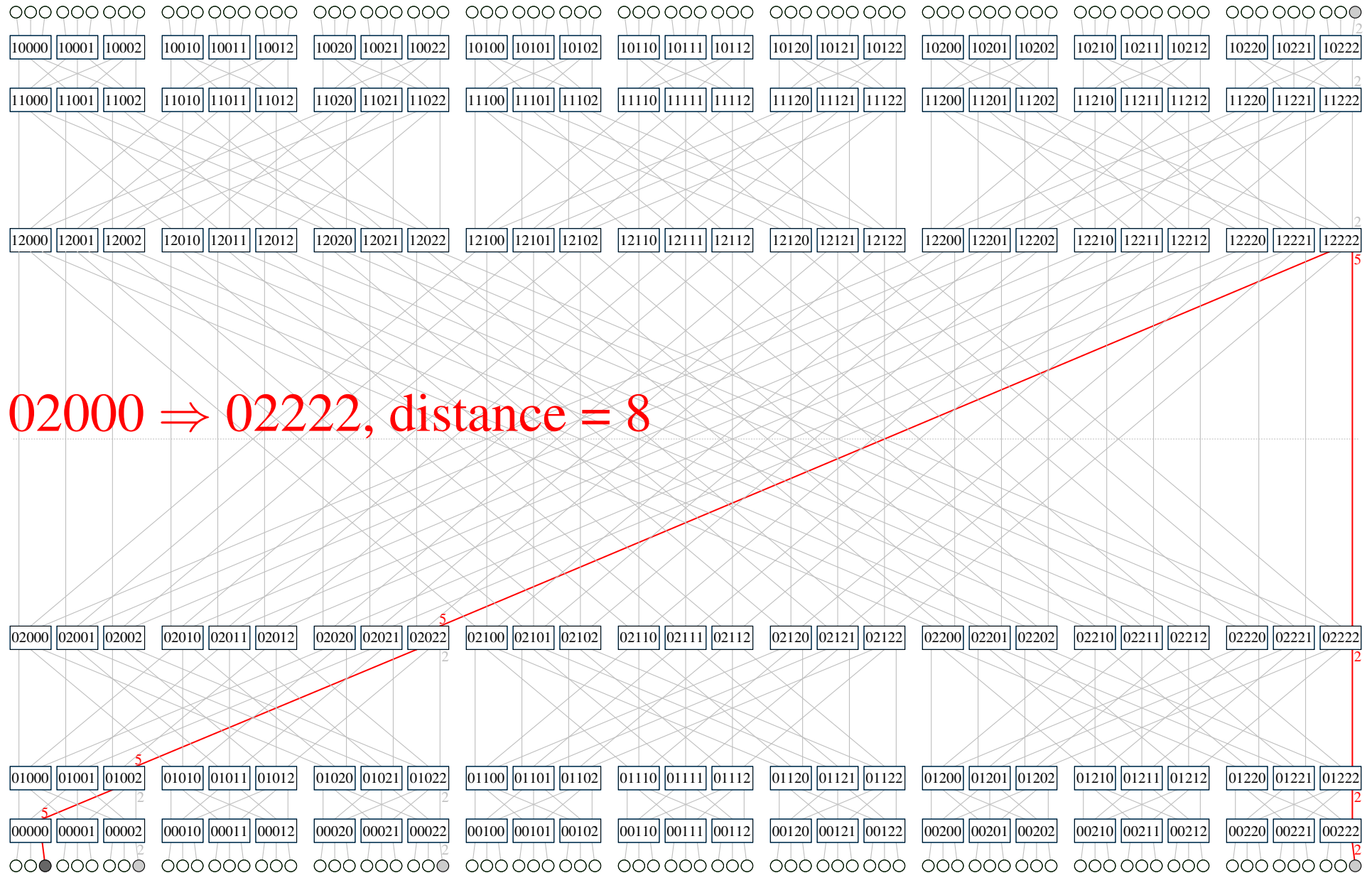
Algorithm 1 MiKANT_Routing (*packet*)

```
Input: packet =  $\langle T, data \rangle$ ;           /* received packet which will be sent to  $T$  */
 $T = \langle G_T, T_{n-1}, T_{n-2}, \dots, T_1, T_0 \rangle$ ;           /* destination node ID */
 $W = \langle G_W, L_W, W_{n-2}, \dots, W_1, W_0 \rangle$ ;           /* my switch ID */
if ( $G_W \neq G_T$ )                                           /*  $W, T$ : different groups */
    send packet to  $T_{L_W}^+$  port;                           /* increasing stage */
else                                                         /*  $W, T$ : same group */
    if ( $W_{n-2}, \dots, W_0 \neq T_{n-2}, \dots, T_0$ )         /* to nearest common ancestor (NCA) */
        send packet to  $T_{L_W}^+$  port;                       /* increasing stage */
    else                                                       /* to destination from NCA */
        if ( $L_W \neq 0$ )                                       /* not a stage 0 switch */
            send packet to  $T_{L_W}^-$  port;                   /* decreasing stage */
        else                                                   /* a stage 0 switch */
            send packet to  $T_{n-1}^-$  port;                   /* to destination node */
        endif
    endif
endif
endif
```

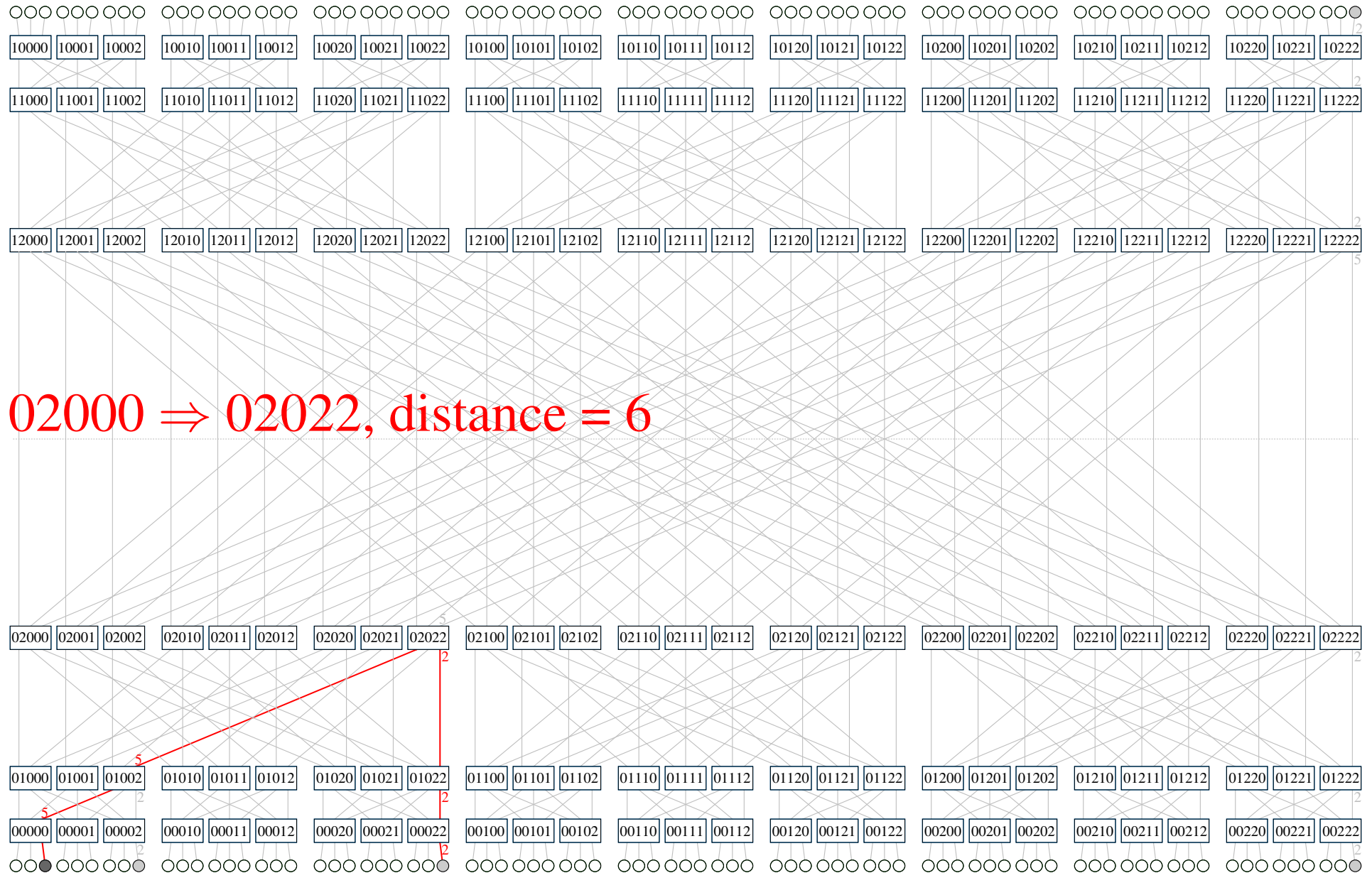
MiKANT(3,4) Routing Example 1



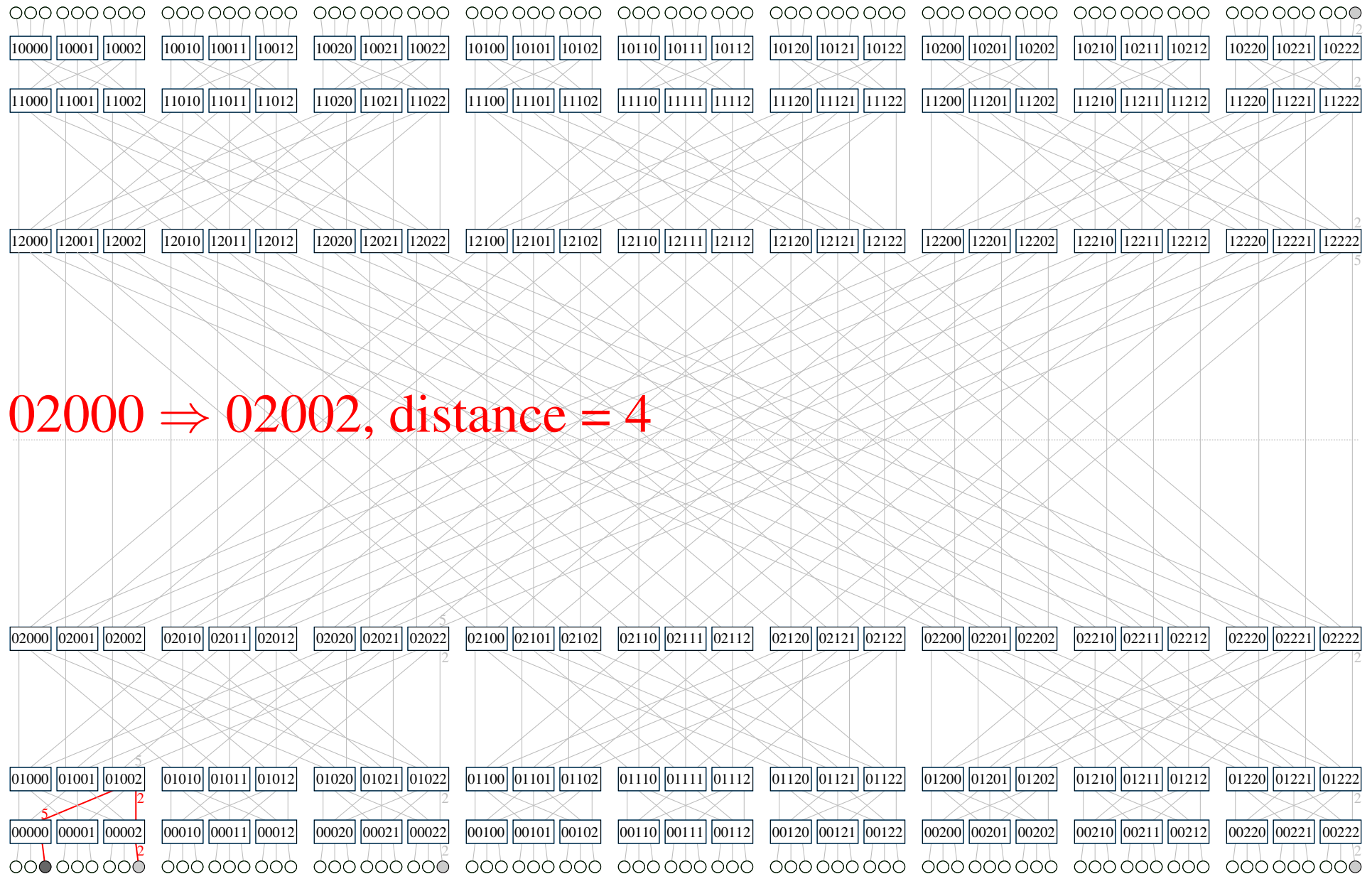
MiKANT(3,4) Routing Example 2



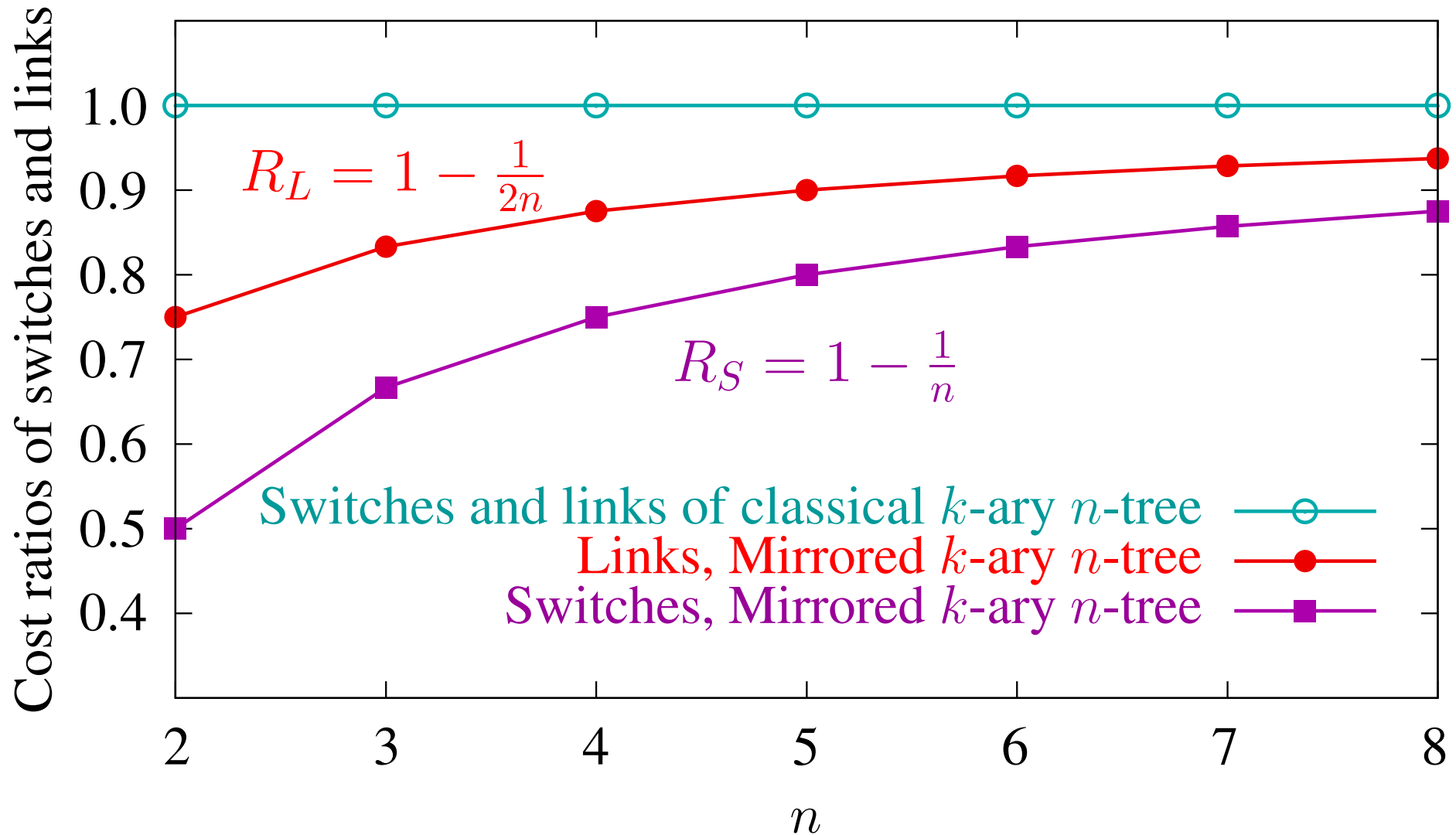
MiKANT(3,4) Routing Example 3



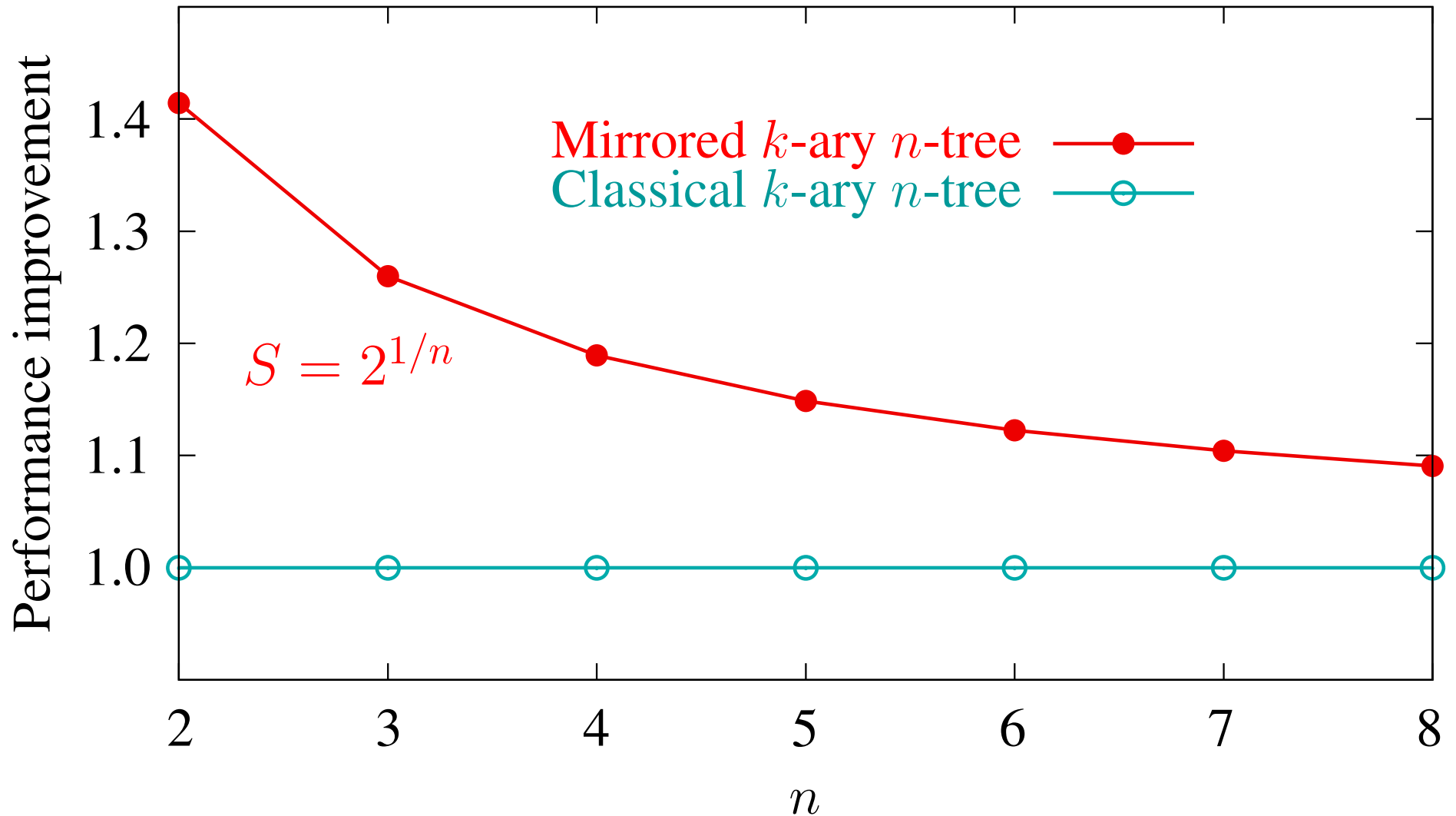
MiKANT(3,4) Routing Example 4



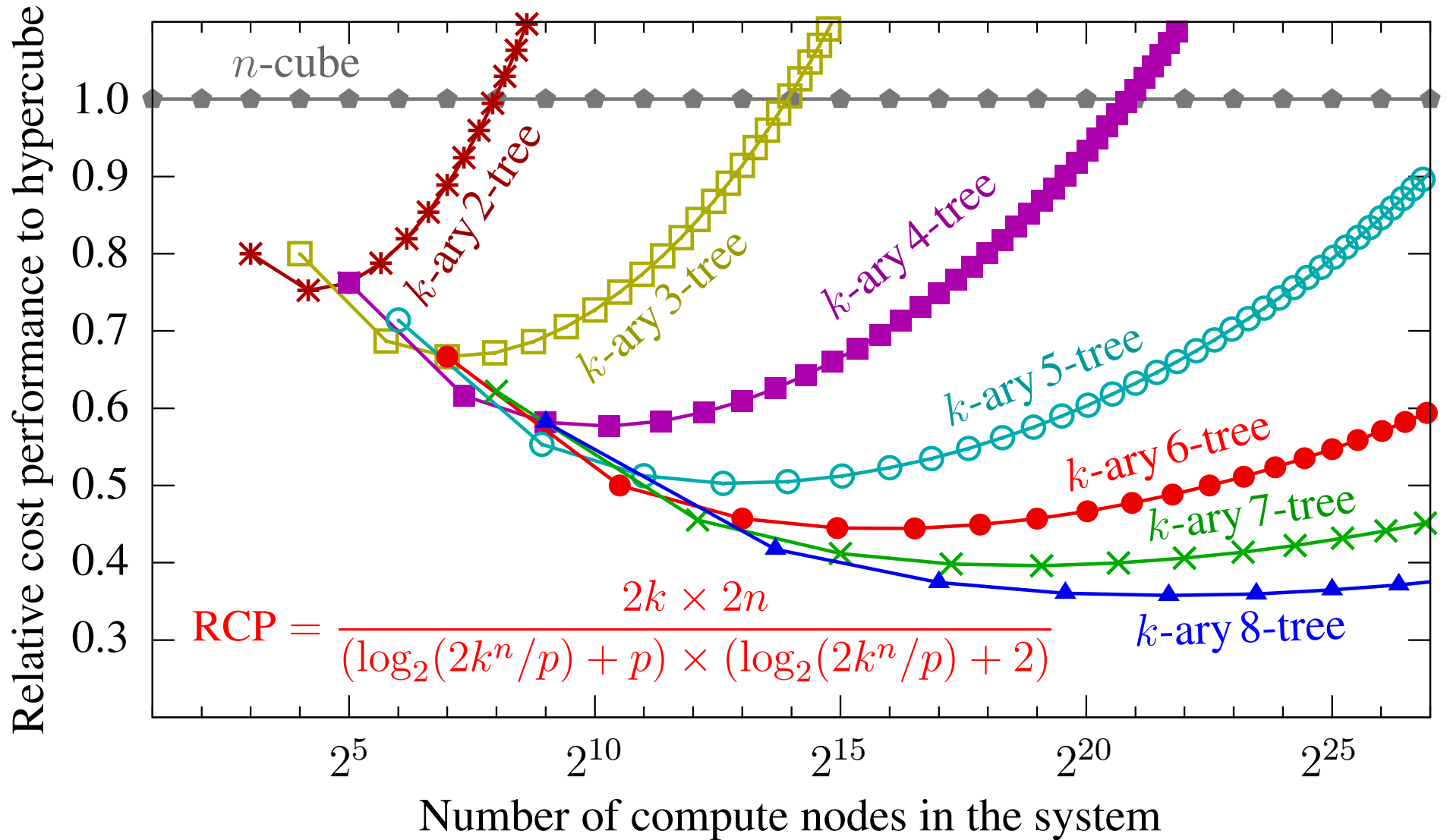
Cost Ratios of Links and Switches



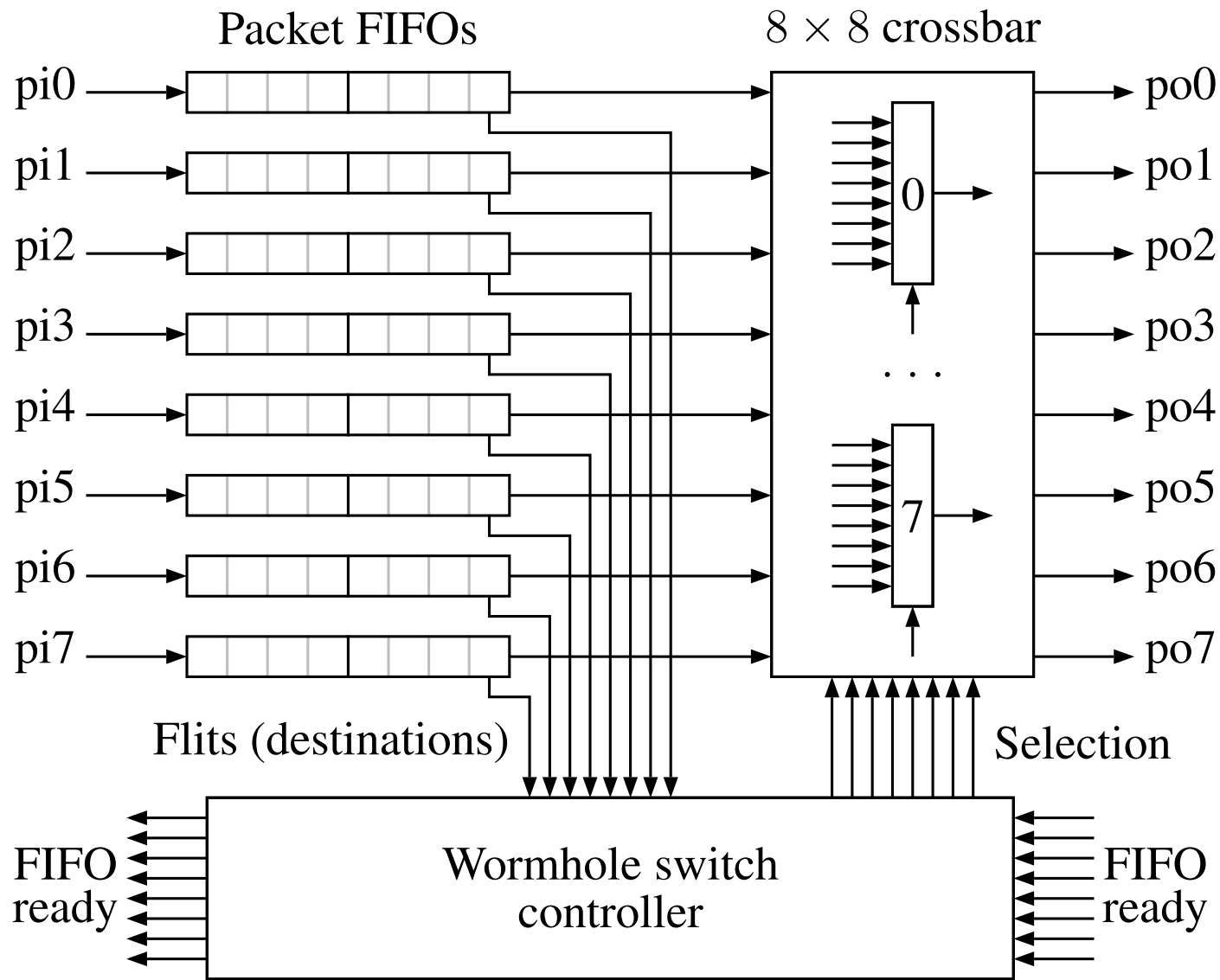
Performance Improvement of MiKANT



Relative Cost Performance to Hypercube



Block Diagram of Switch for Simulation

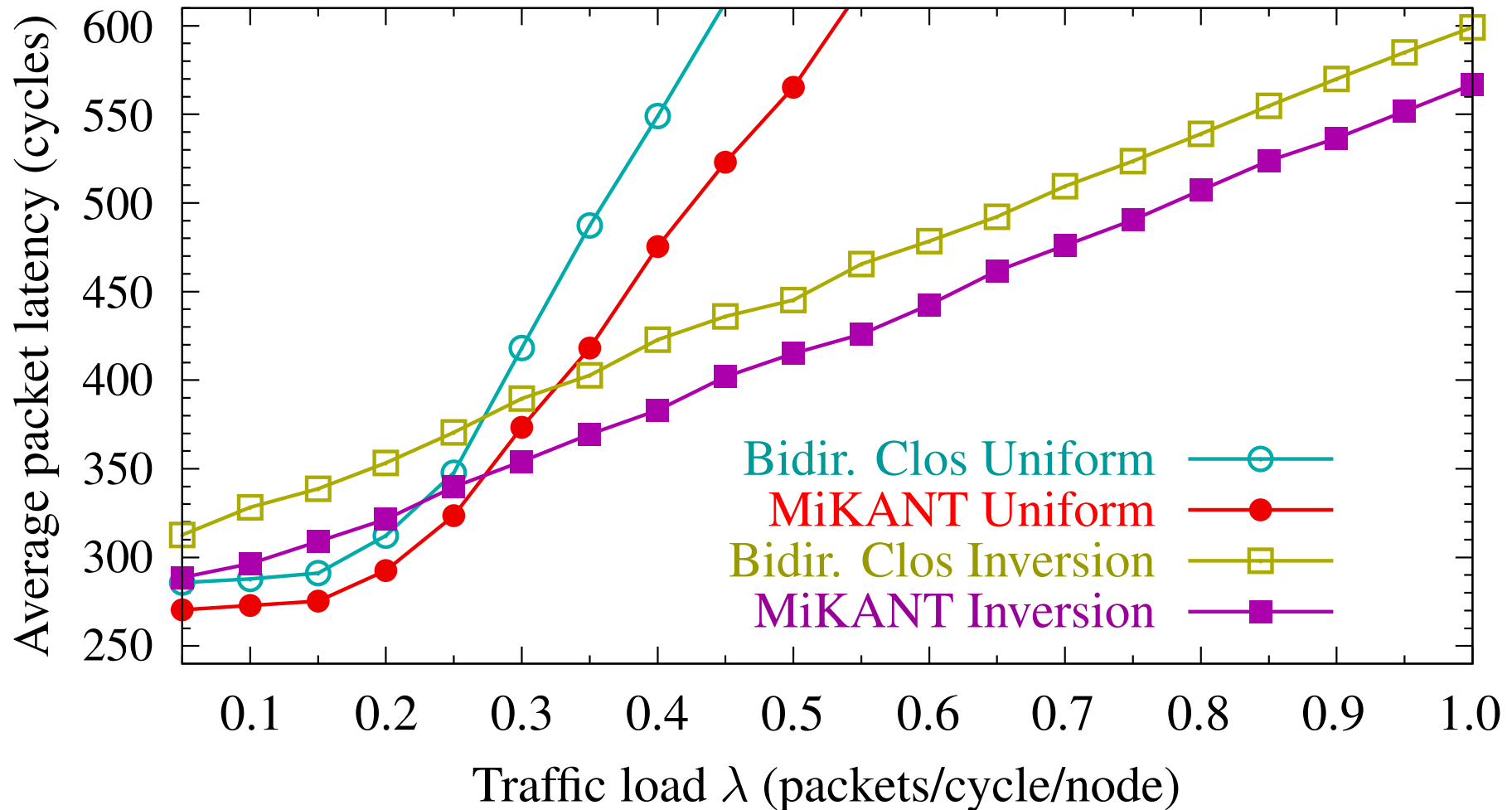


Synthetic Simulation

- Evaluate average packet latencies
- Network size: $k = 4$ and $n = 5$
- Use the minimal per-hop deterministic routing algorithm
- Simulate with two synthetic traffic patterns
 - **Uniform**: Destinations are randomly assigned
 - **Bit-inversion**: Destinations = bit-inversion of sources
- Traffic load λ in the range of 0.05 and 1.00
- Terminate when at least 200 packets of each source node reached their destinations
- Use the Wormhole switching routing technique (32-bit/flit)

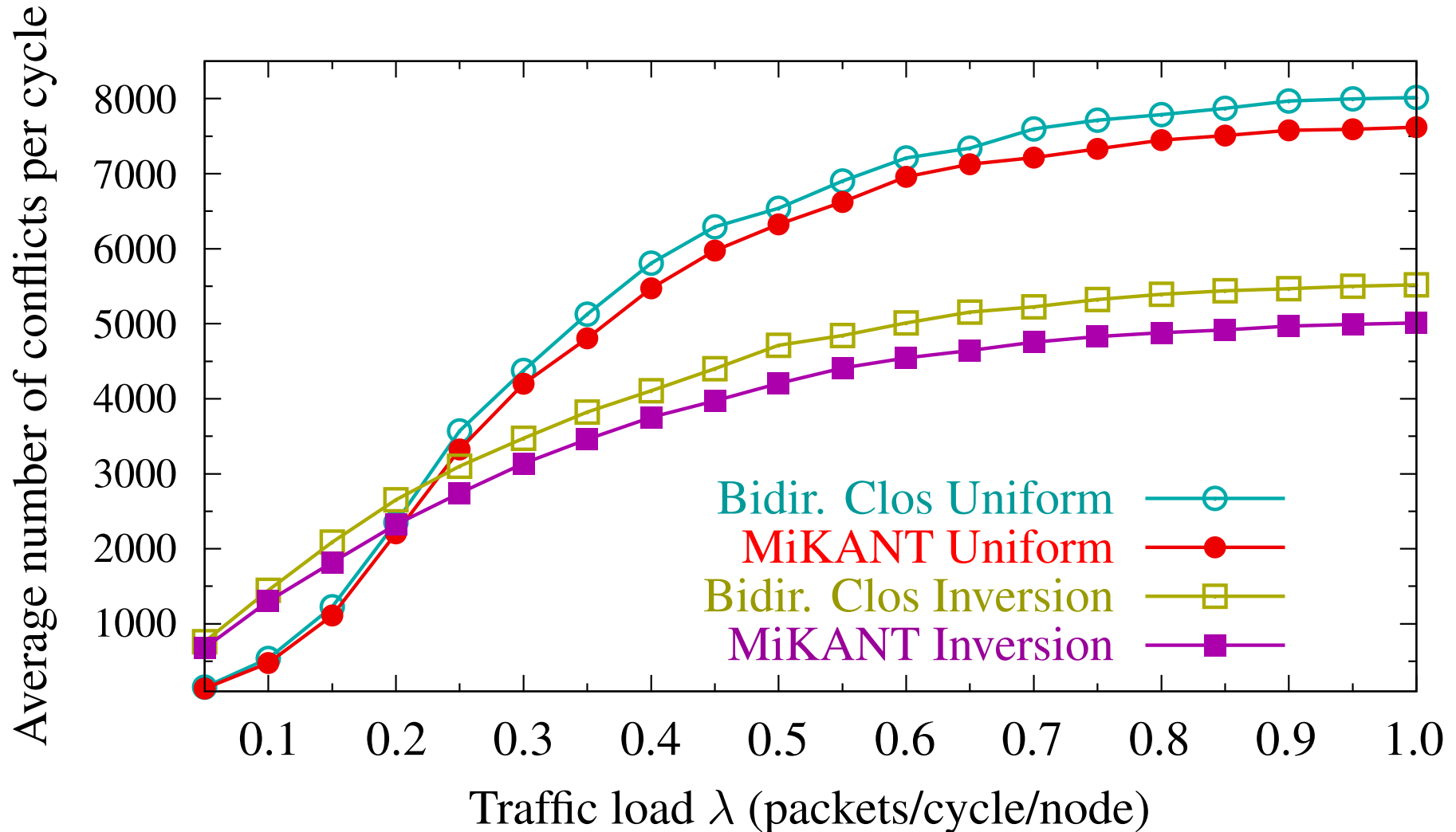
Average Packet Latencies (Clock Cycles)

MiKANT has shorter packet latency than Clos network



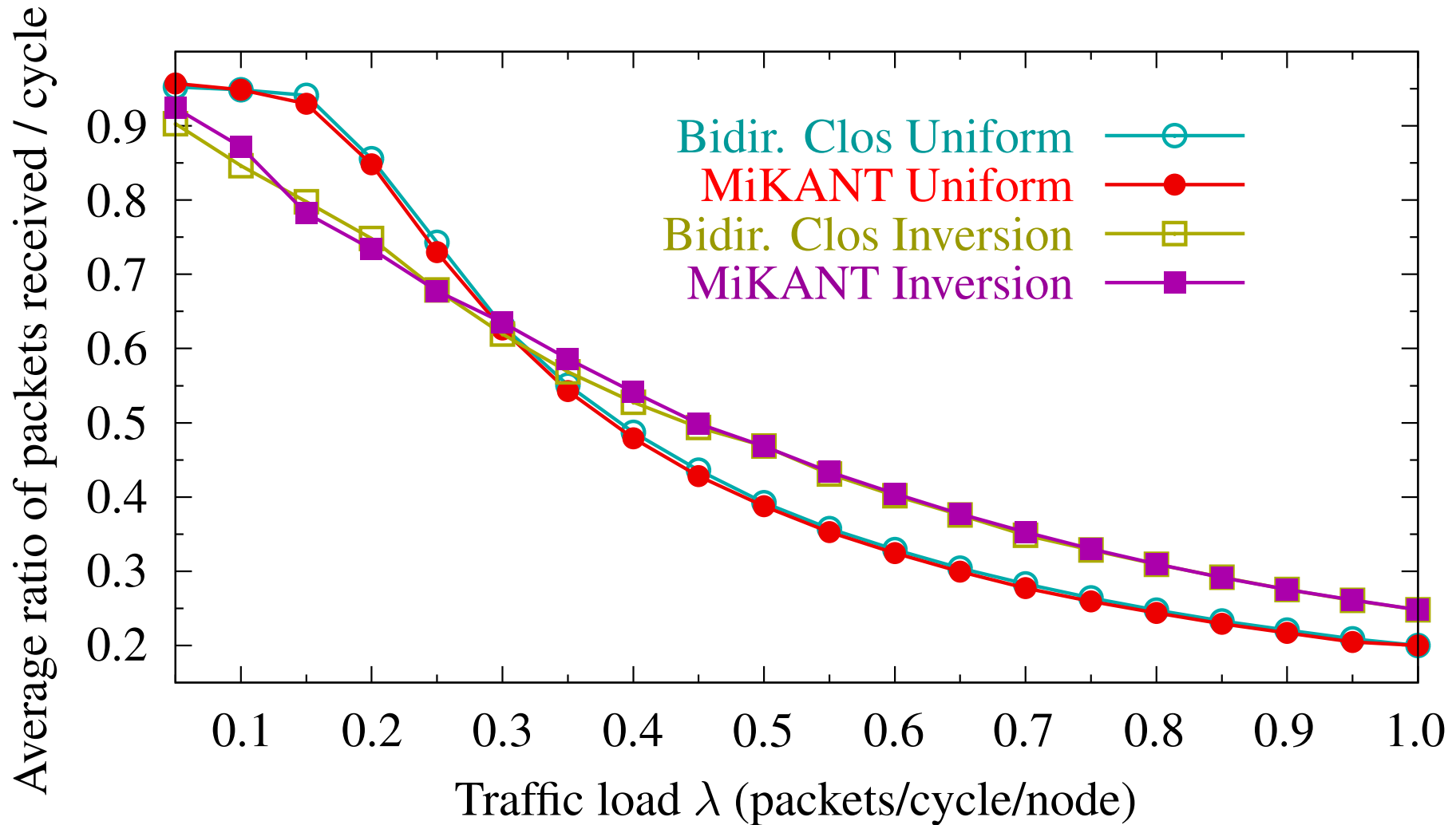
Average Output Port Conflicts Per Cycle

MiKANT has fewer output port conflicts than Clos network



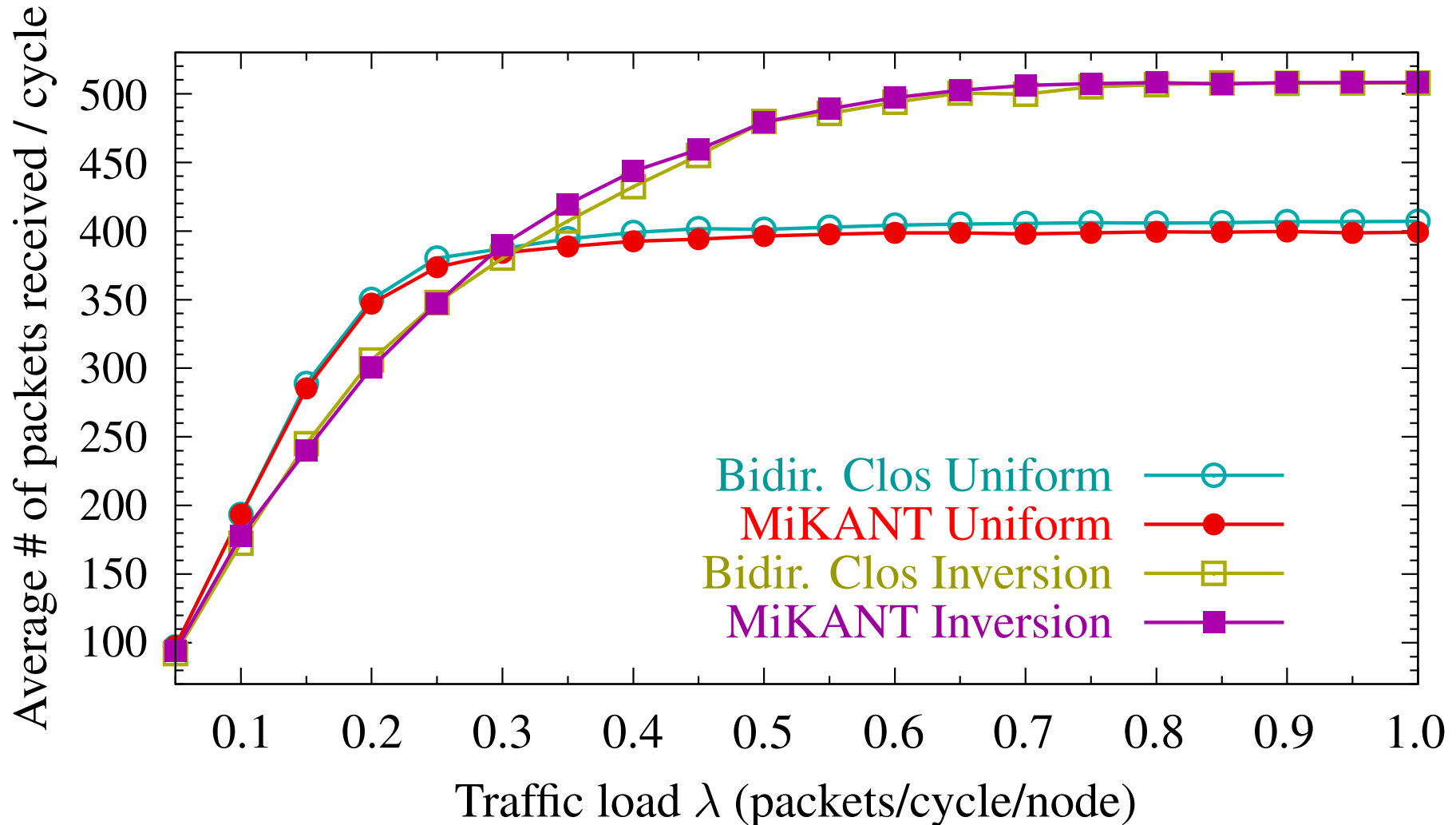
Ratios of Packets Received Per Cycle

MiKANT has almost the same capacity as Clos network



Average Packets Received Per Cycle

MiKANT has almost the same capacity as Clos network



Conclusions and Future Work

- Fat-tree & Clos network are widely used in supercomputers
- The proposed MiKANT has several advantages
 - Symmetric (all switches have the same radix)
 - Low hardware (switch and link) cost
 - Short average distance
 - Short packet latency
- Future research work
 - Adaptive routing algorithms for MiKANT
 - Fault tolerant routing in MiKANT
 - Implementation of MiKANT on NoC