Finding a Hamiltonian Cycle in a Hierarchical Dual-Net with Base Network of p-Ary q-Cube

Yamin Li, Department of Computer Science, Hosei University, yamin@hosei.ac.jp
Shietung Peng, Department of Computer Science, Hosei University, speng@hosei.ac.jp
Wanming Chu, Department of Computer Hardware, University of Aizu, w-chu@u-aizu.ac.jp

Contents

- Hierarchical Dual-Net (HDN)
  - Why
  - What
- Finding a Hamiltonian Cycle in an HDN
  - How

Interconnection Networks

- The modern high-performance supercomputers consist of hundreds of thousands of CPUs
- In the near future, the number of CPUs in supercomputers will reach several millions
- How to connect these extremely large number of CPUs is an important issue for achieving high performance of the supercomputers
- A "good" interconnection network should use a small number of links and meanwhile keep the diameter as shorter as possible
- The symmetric structure and efficient routing should also be considered

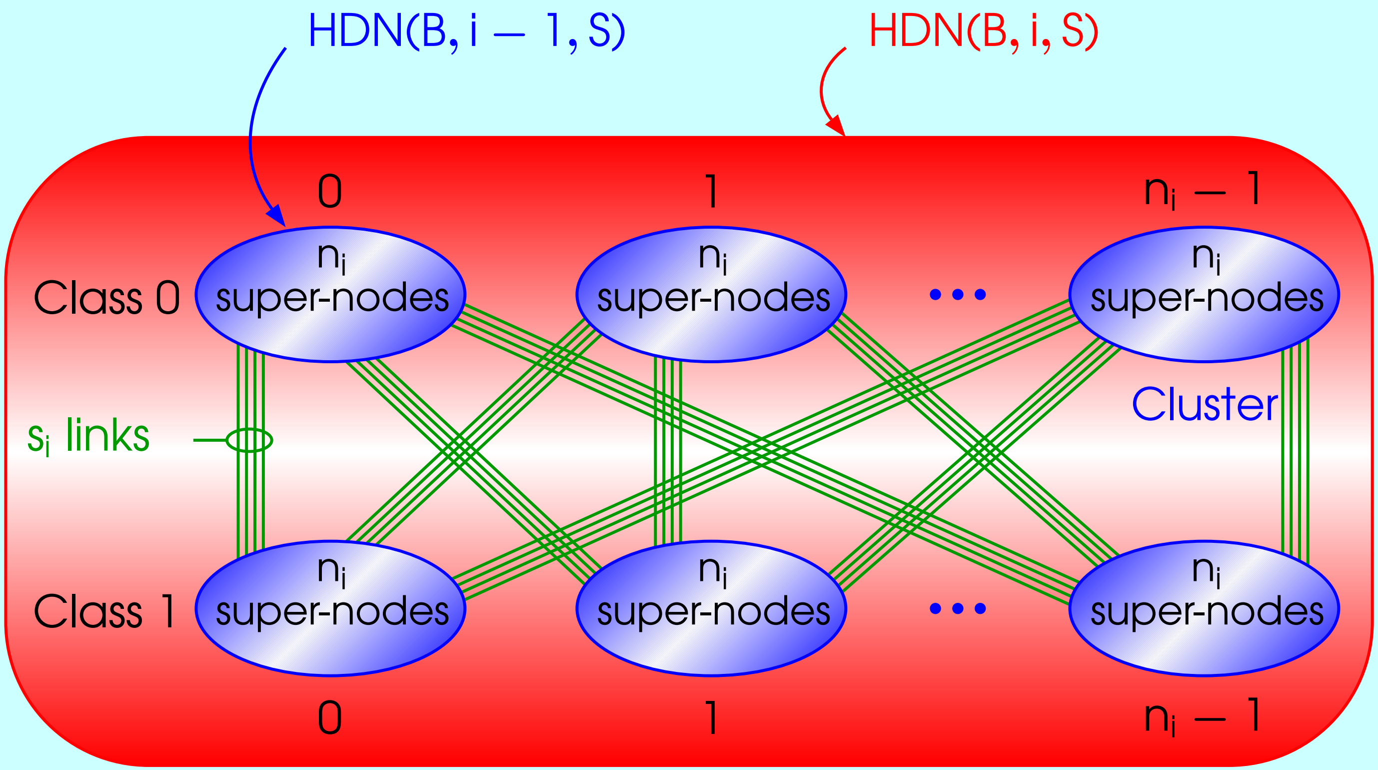
Interconnection Networks

- Ring
  - Too simple, long diameter
- Complete graph
  - Too complex, expensive
- 3D Torus
  - Long diameter in a large-scale system
- Hypercubes
  - The number of links increases logarithmically
- Tree / Fat Tree
  - Not symmetric

The Hierarchical Dual-Net (HDN)

- Use a small popurlar IN as *base network*
- Hierarchical structure
- Small and flexible node-degree
- Short diameter
- Efficient routing and broadcasting
- Flexible scale (small – very large systems)

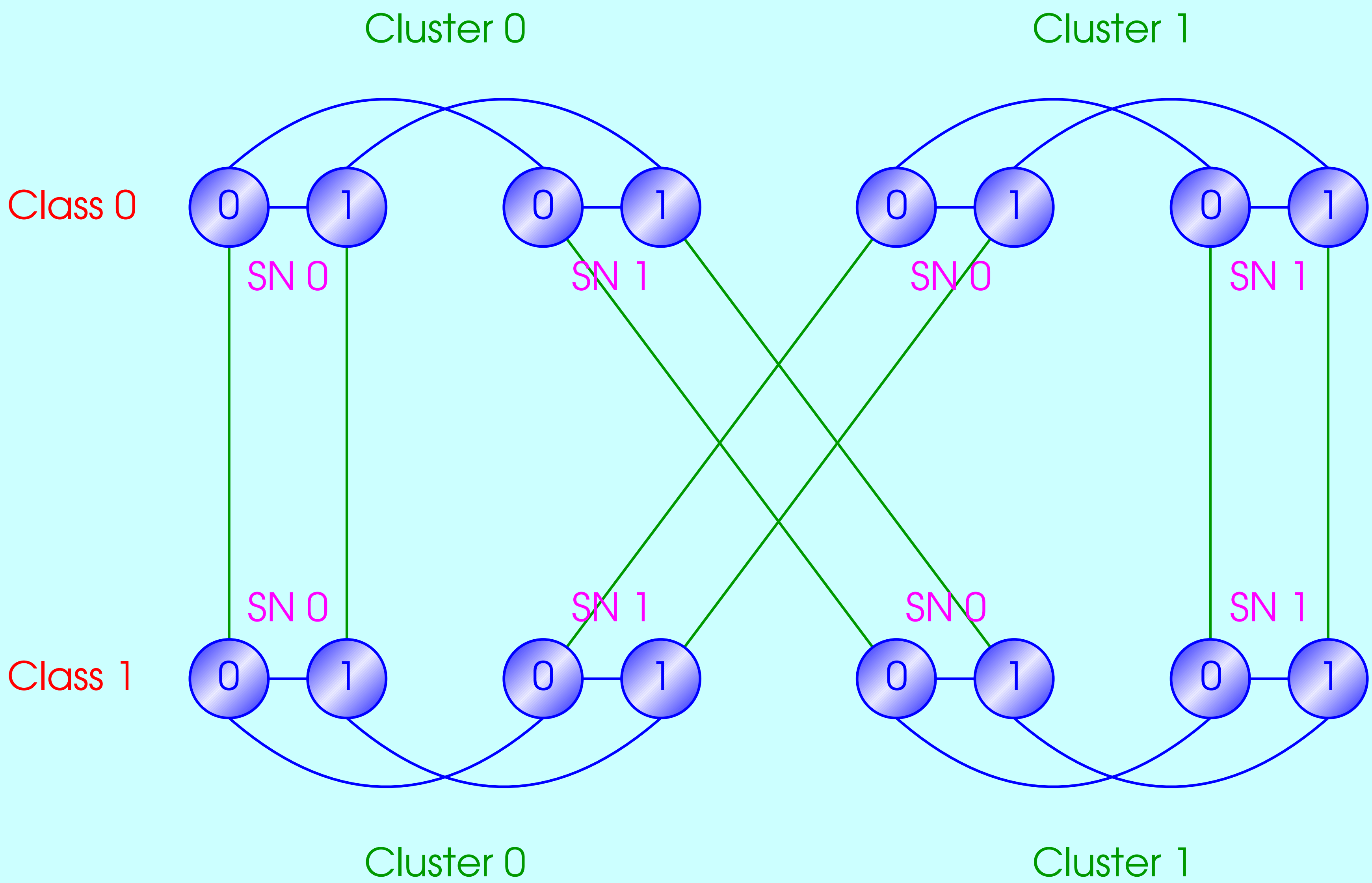
Building an HDN(B,i,S) from HDN(B,i−1,S)



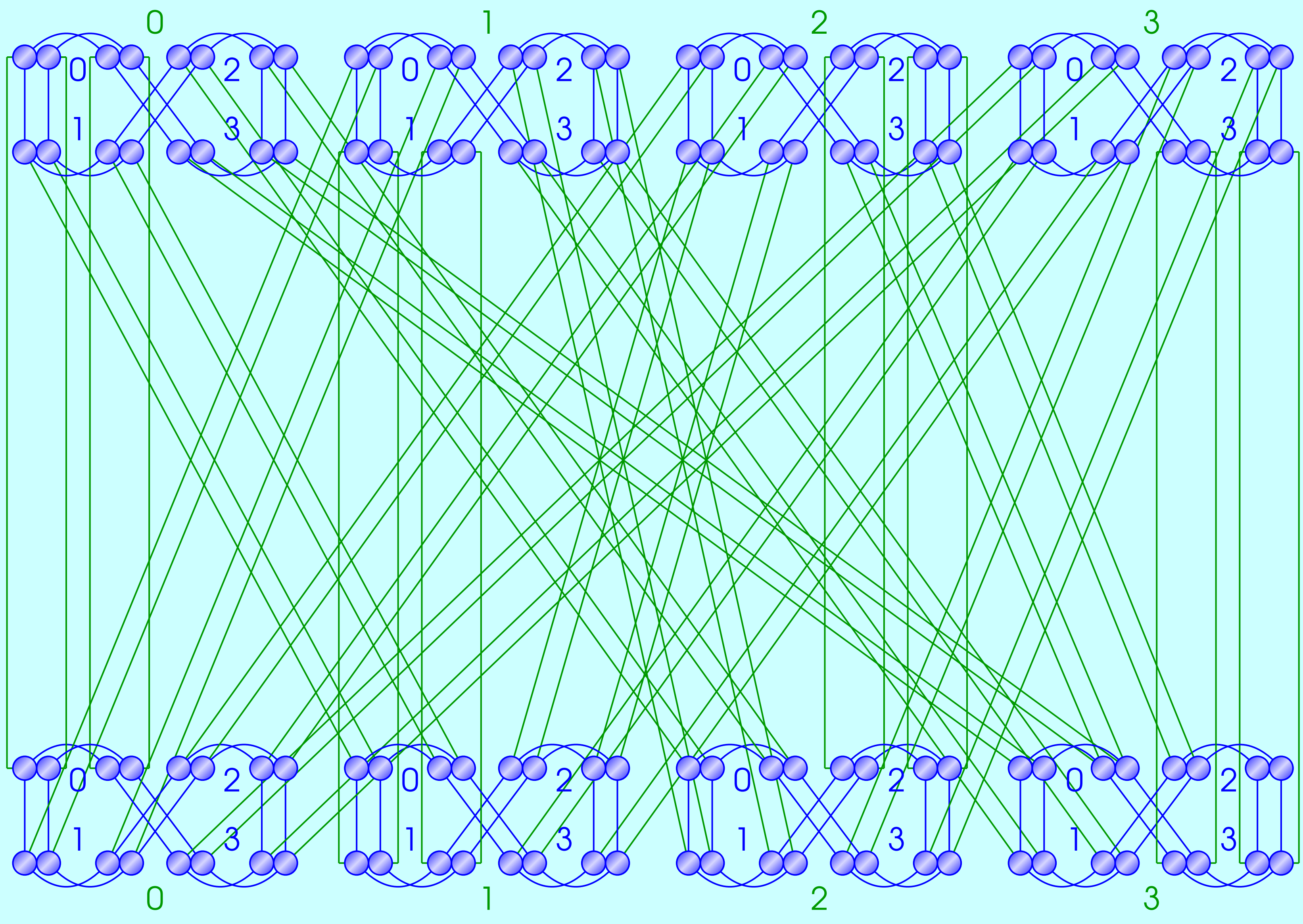
HDN(B, k, S):

B is a base network; k is the *level* of the HDN;
$S = \{s_1, s_2, \ldots, s_k\}$ where s_i is the number of nodes in a *super-node* at the level i.

An HDN(B,1,S) with s_1 = 2 (Base = 2-Cube)

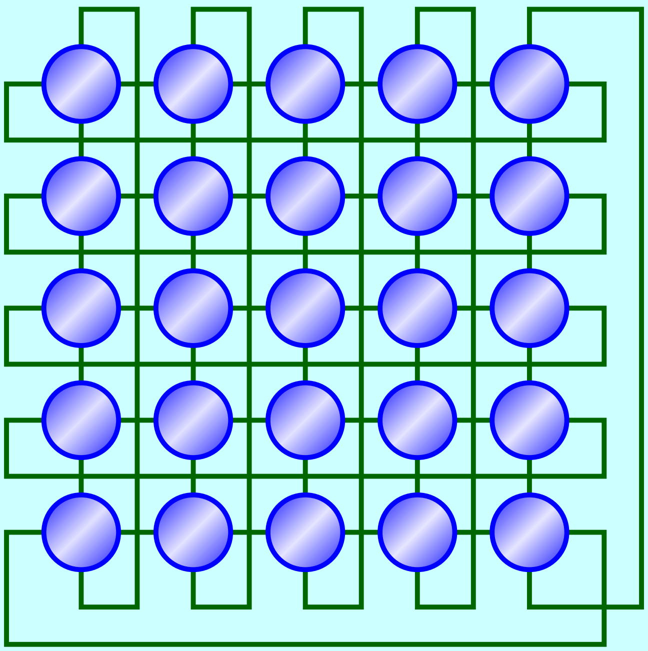


An HDN(B,2,S) with $s_1 = 2$ and $s_2 = 4$



Base Network

- Select a symmetric product graph as a base network
- For example,
  - hypercube,
  - torus, or
  - p-ary q-cube



a 5-ary 2-cube

The Number of Nodes in HDN(B,k,S)

Suppose that the base network is a $2 * 3 * 5$ torus

Degree = 7

$k = 1$	$s_1 = 1$	$s_1 = 2$	$s_1 = 3$	$s_1 = 5$	$s_1 = 6$	$s_1 = 10$	$s_1 = 15$	$s_1 = 30$
	1,800	900	600	360	300	180	120	60

Degree = 8

$k = 2$	$s_2 = 1$	$s_2 = 2$	$s_2 = 3$	$s_2 = 5$	$s_2 = 6$	$s_2 = 10$	$s_2 = 15$	$s_2 = 30$
$s_1 = 1$	6,480,000	3,240,000	2,160,000	1,296,000	1,080,000	648,000	432,000	216,000
$s_1 = 2$	1,620,000	810,000	540,000	324,000	270,000	162,000	108,000	54,000
$s_1 = 3$	720,000	360,000	240,000	144,000	120,000	72,000	48,000	24,000
$s_1 = 5$	259,200	129,600	86,400	51,840	43,200	25,920	17,280	8,640
$s_1 = 6$	180,000	90,000	60,000	36,000	30,000	18,000	12,000	6,000
$s_1 = 10$	64,800	32,400	21,600	12,960	10,800	6,480	4,320	2,160
$s_1 = 15$	28,800	14,400	9,600	5,760	4,800	2,880	1,920	960
$s_1 = 30$	7,200	3,600	2,400	1,440	1,200	720	480	240

Topological Properties

Network	The number of nodes (N)	Degree (d)
3D Torus	$x * y * z$	6
n-cube	2^n	n
CCC(n)	$n * 2^n$	3
Dual-Cube(n)	2^{2n-1}	n
RDN(m, k)	$(2m)^{2^k}/2$	$d_0 + k$
HDN(B, k, S)	$(2\|B\|)^{2^k}/(2\prod_{i=1}^{k} s_i)$	$d_0 + k$

Topological Properties

Network	Diameter (D)
3D Torus	$(x + y + z)/2$
n-cube	n
CCC(n)	$2n + \lfloor n/2 \rfloor - 2$
Dual-Cube(n)	$2n$
RDN(m, k)	$2^k * D_0 + 2^{k+1} - 2$
HDN(B, k, S)	$2^k(D(B) - \sum_{j=0}^{k-1} 2^j(D(SN^{k-j})) + 2^{k+1} - 2$

Define Weighted Cost Ratio :

$$CR_w(G) = \frac{w_1 \times d(G) + w_2 \times D(G)}{\log_2 |(G)|}$$

B: $2 * 3 * 5$ torus; $w_1 = w_2 = 50\%$

Network	N	d	D	CR_w
10-cube	1,024	10	10	1.00
3D-Tori(10)	1,000	6	15	1.05
HDN(B, 1, (1))	1,800	7	10	0.79
HDN(B, 1, (2))	900	7	9	0.82
HDN(B, 1, (3))	600	7	9	0.87

Weighted Cost Ratio

Define Weighted Cost Ratio :

$$CR_w(G) = \frac{w_1 \times d(G) + w_2 \times D(G)}{\log_2 |(G)|}$$

B: $2 * 3 * 5$ torus; $w_1 = w_2 = 50\%$

Network	N	d	D	CR_w
19-cube	524,288	19	19	1.00
3D-Tori(80)	512,000	6	120	3.32
HDN(B, 2, (2, 2))	810,000	8	19	0.69
HDN(B, 2, (2, 5))	324,000	8	18	0.71
HDN(B, 2, (5, 2))	129,600	8	17	0.74

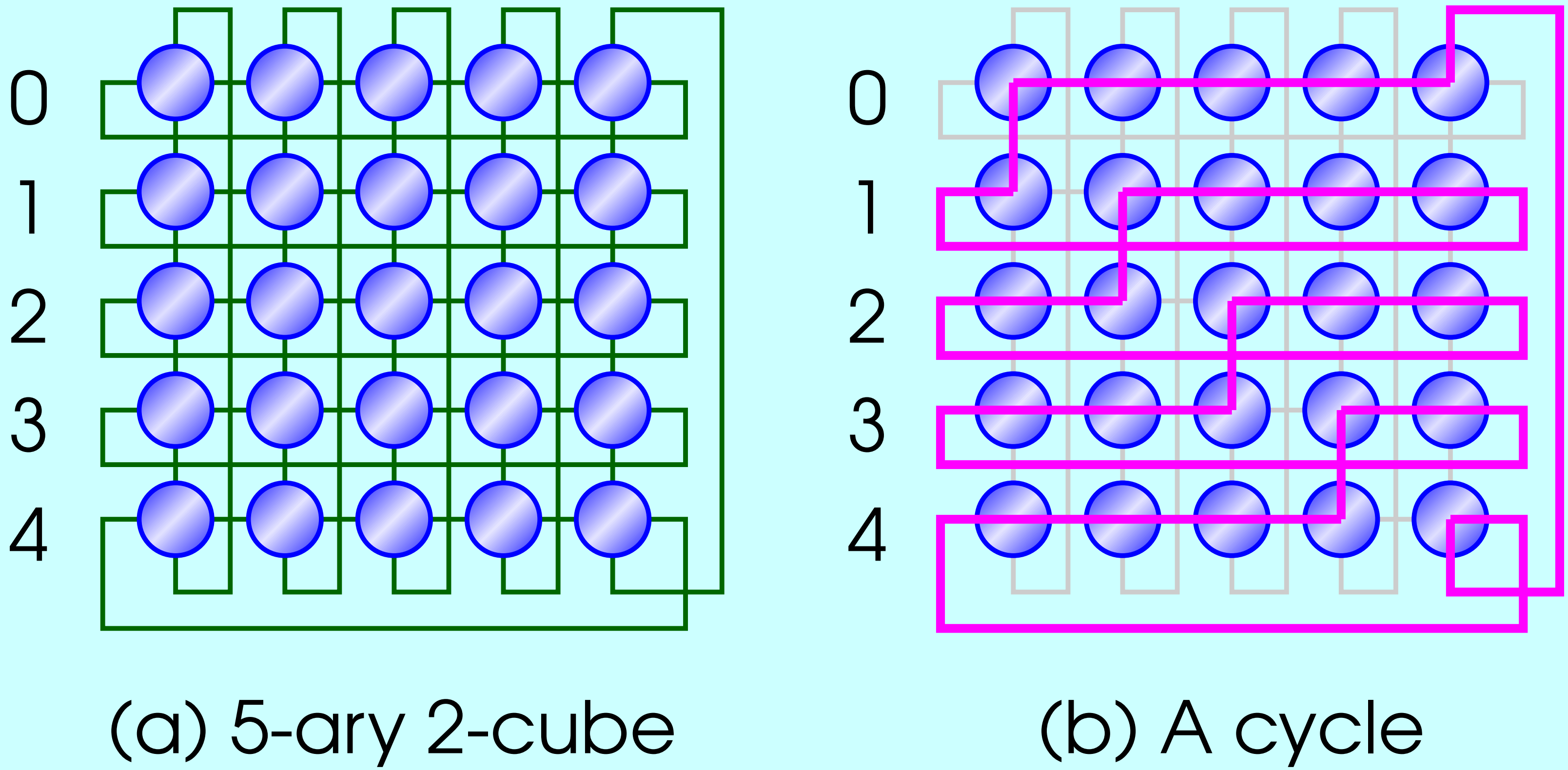
Hamiltonian Cycle Embedding

- Linear array and ring are two fundamental networks and many algorithms were designed based on linear array and ring
- Thus embedding linear array or ring in networks is important for emulating those algorithms
- A Hamiltonian cycle of an undirected graph G is a simple cycle that contains every node in G exactly once
- A Hamiltonian path in a graph is a simple path that visits every node exactly once
- A graph that contains a Hamiltonian cycle is said to be Hamiltonian

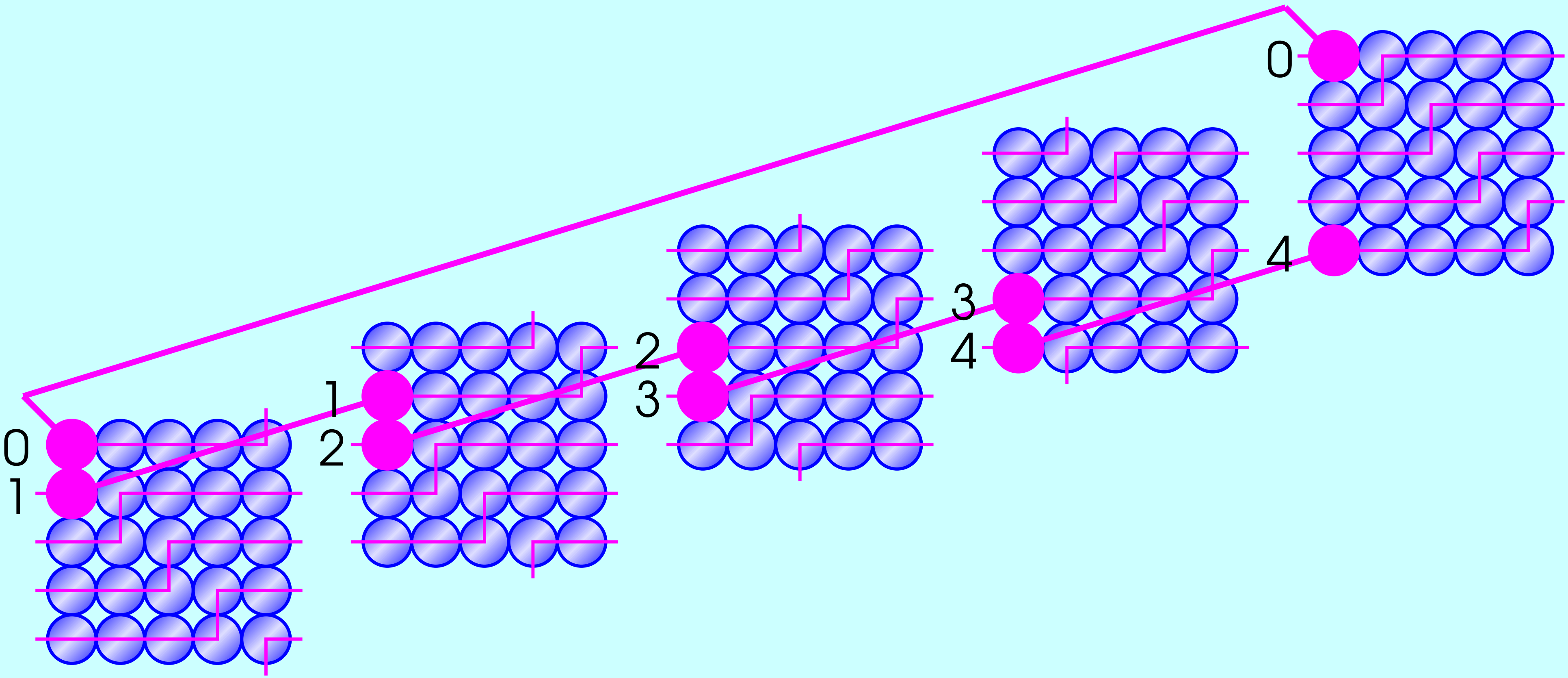
The P-Ary Q-Cube

- A p-ary q-cube connects p^q nodes
  - p: the number of nodes per dimension
  - q: the network dimension
- Each node can be identified by a q-digit radix-p address $(a_0, a_1, \ldots, a_i, \ldots, a_{q-1})$
- There is a link connecting node A with address $(a_0, a_1, \ldots, a_{q-1})$ and node B with address $(b_0, b_1, \ldots, b_{q-1})$ if and only if there exists i $(0 \leq i \leq q - 1)$ such that $a_i = (b_i + 1)\% p$ and $a_j = b_j$ for $0 \leq j \leq q - 1$ and $i \neq j$
- For examples, a 4-ary 2-cube is a 4×4 torus and a 2-ary 4-cube is a 4-cube

A Hamiltonian cycle in a 5-ary 2-cube



A Hamiltonian cycle in a 5-ary 3-cube

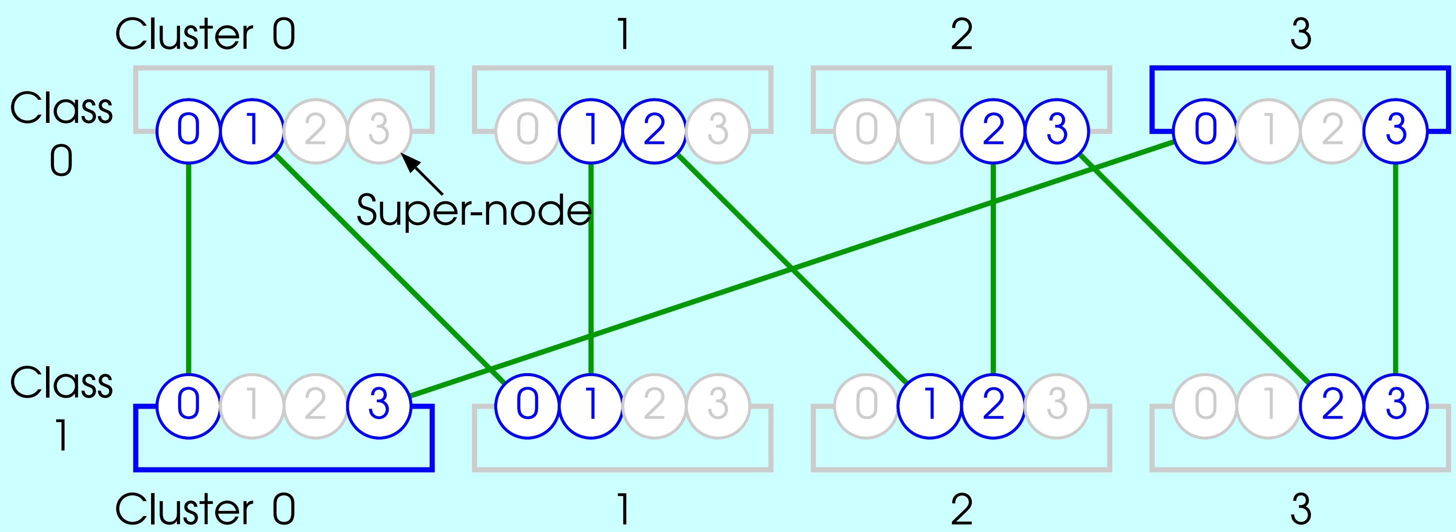


Lemma: There is a Hamiltonian cycle in a p-ary q-cube

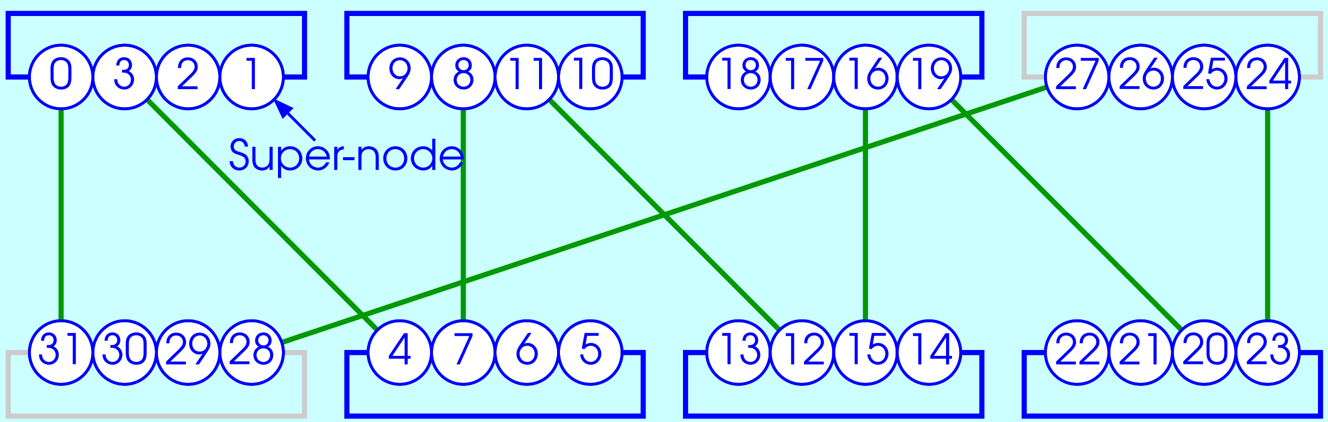
Building A Hamiltonian Cycle in Level 1

- There are $2n_1$ clusters and each cluster contains n_1 super-nodes
- Building a virtual Hamiltonian cycle (VHC)
  - An HVC connects all $2n_1$ clusters
  - Two super-nodes in a cluster are in the VHC
- Including all super-nodes
  - Replacing the two super-nodes with the Hamiltonian path
- Renaming super-nodes
- Including all nodes
  - Expanding from super-node level to node level

A Hamiltonian Cycle for Super-Nodes

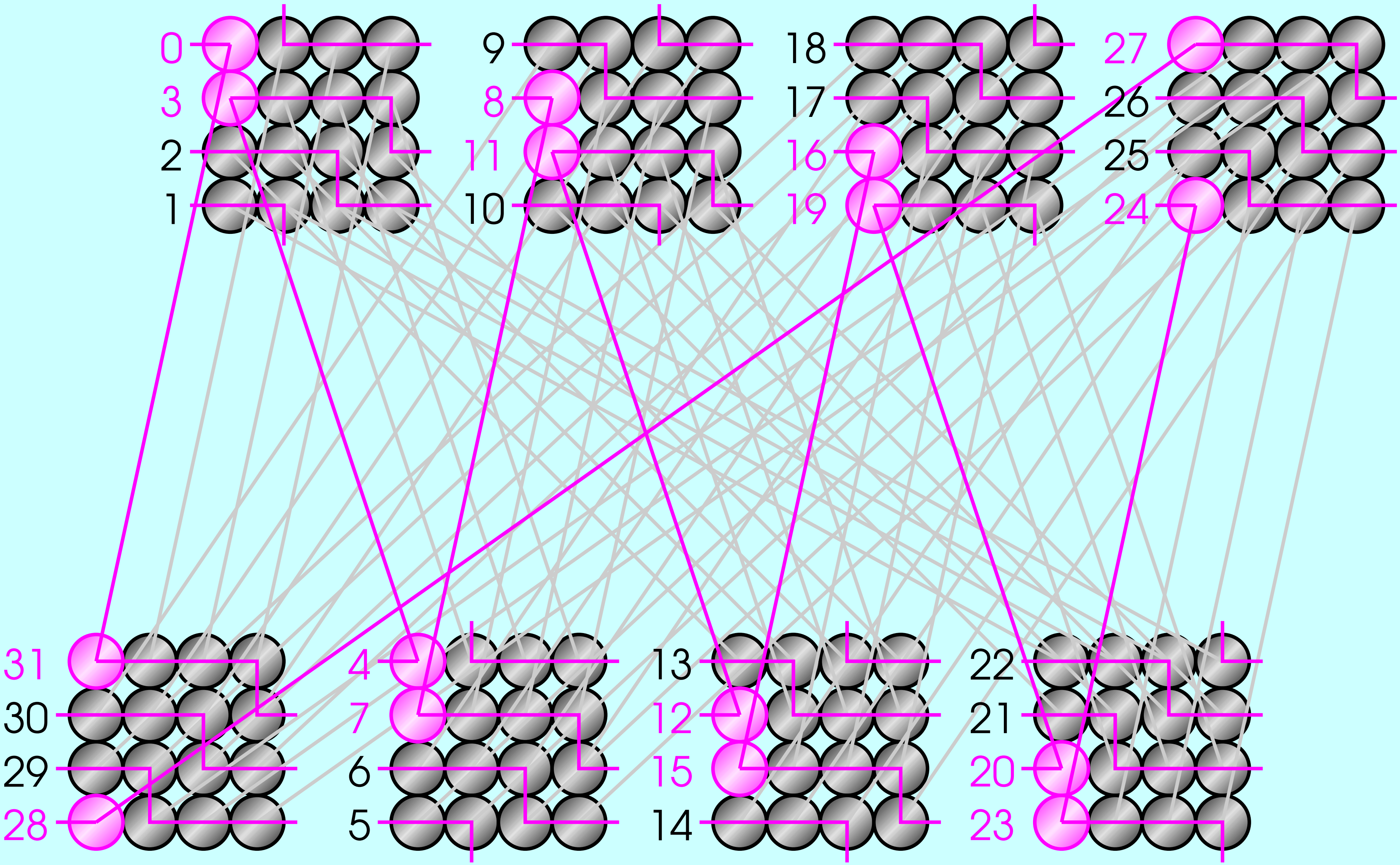


(a) A virtual Hamiltonian cycle connecting all clusters



(b) Including and renaming all super-nodes

A Hamiltonian Cycle for All Nodes in HDN(B,1,(4))

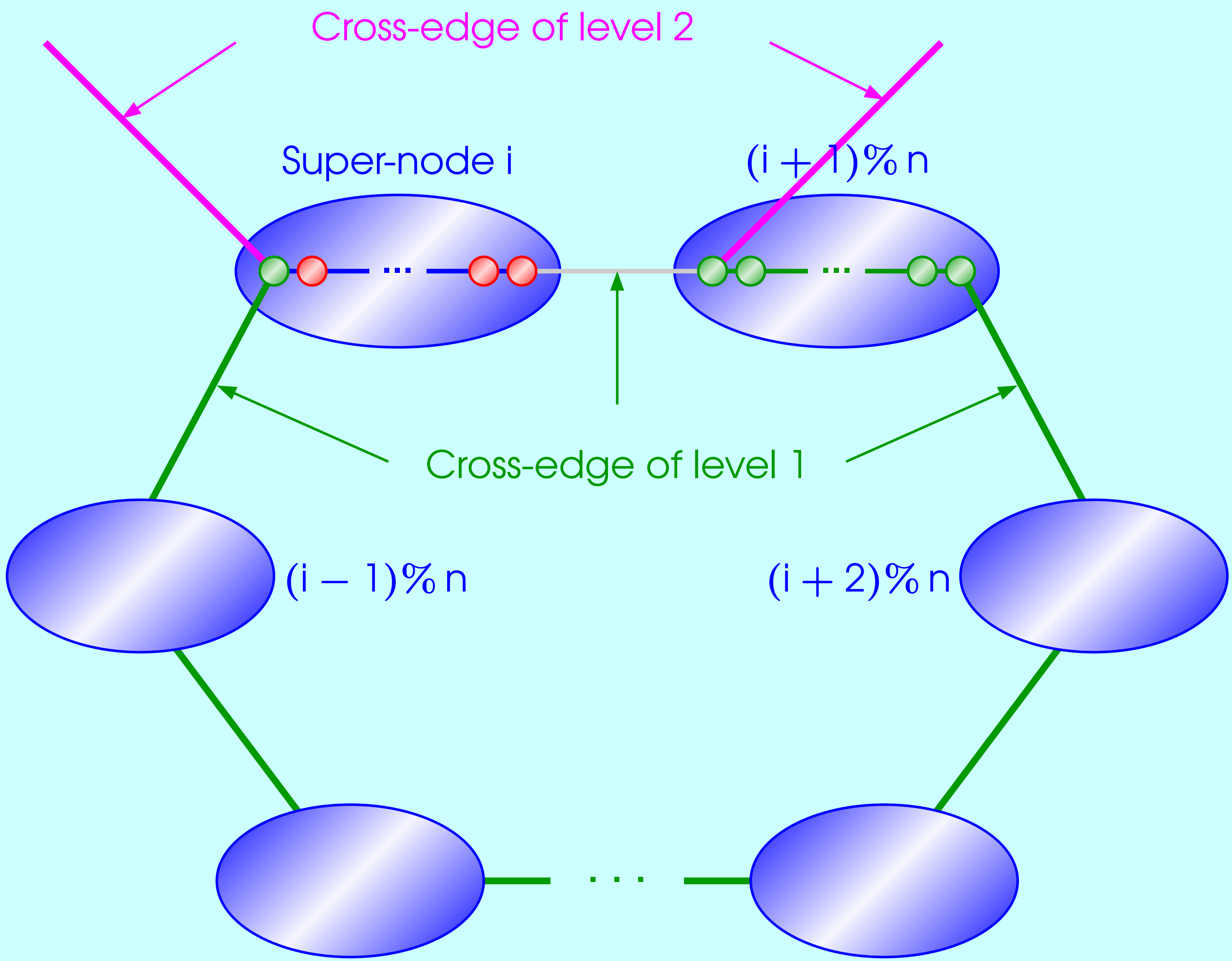


Building A Hamiltonian Cycle in Level 2

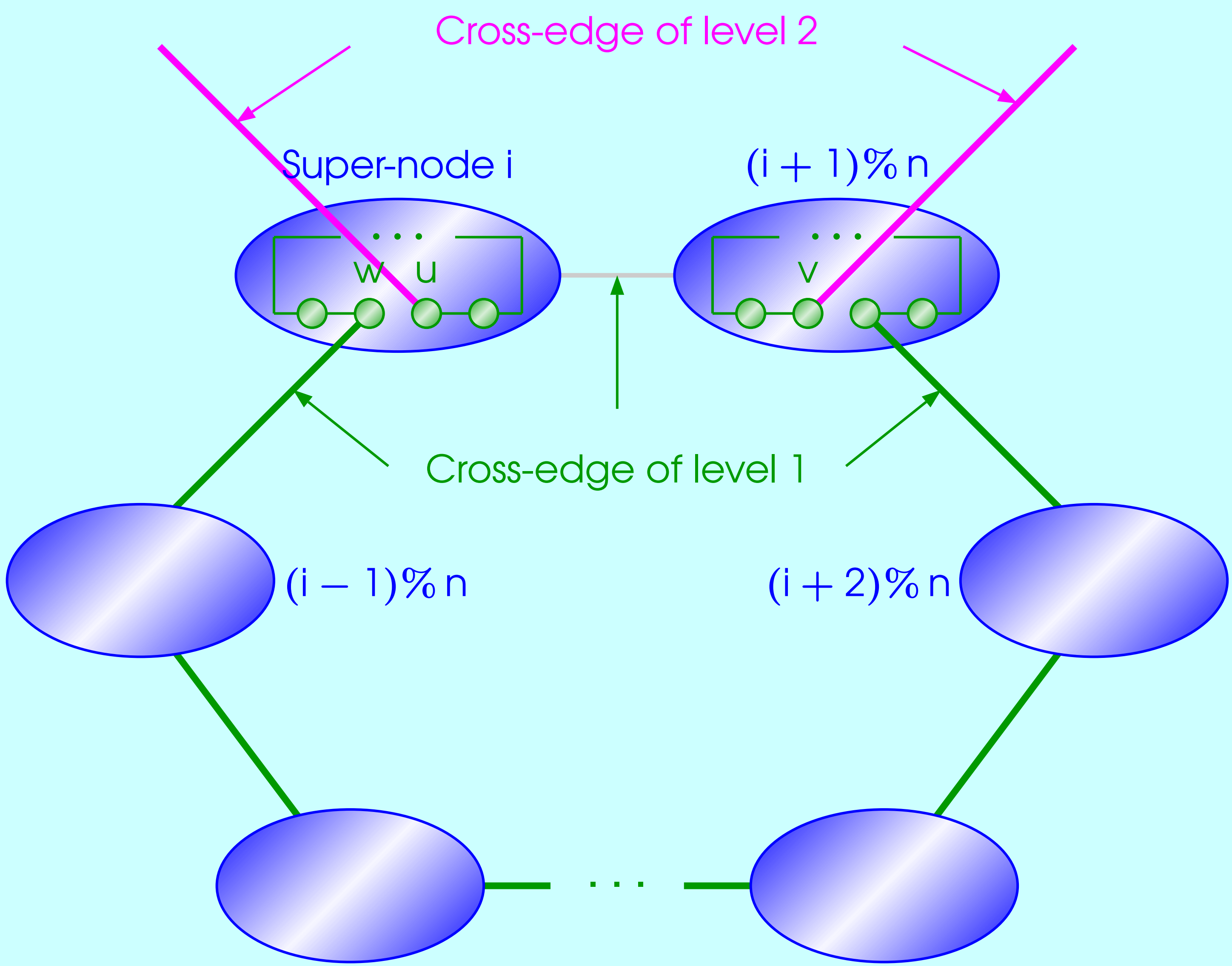
- Build an VHC connecting all clusters at level 2
- Insert all super-nodes to the cycle
- Expand super-nodes to nodes
  - There is a problem: some nodes cannot be included in super-node i
    - To solve this problem, we route in super-node i first

Lemma: For an HDN with the base network of p-ary q-cube, given any node $u \in$ renamed super-node i in a cluster, for $i = 0, 1, \ldots, n-1$, there is a path $u \longrightarrow v$ containing all of the nodes in the cluster with $v \in$ renamed super-node $(i+1)\%n$

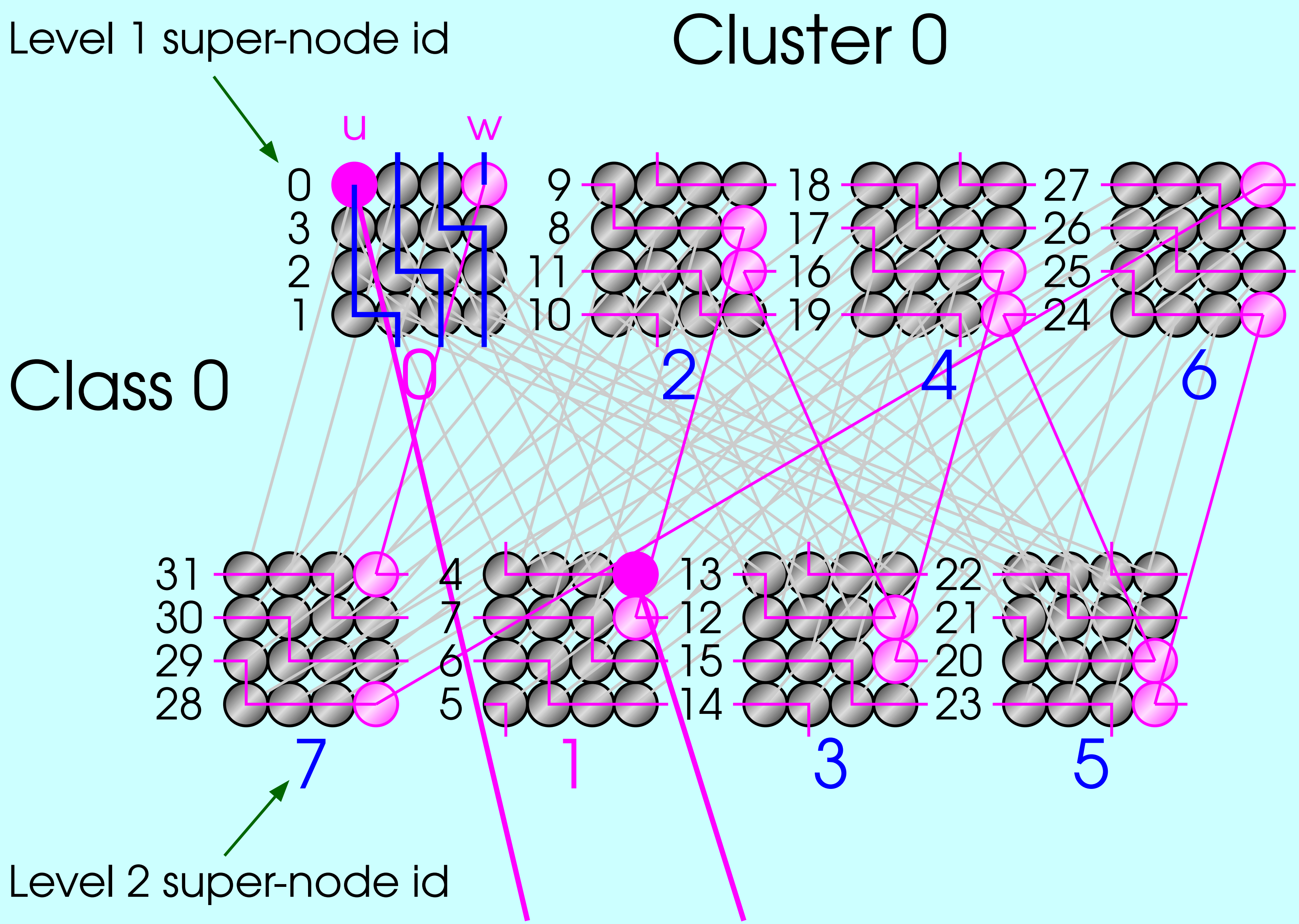
Some Nodes in Super-Node i Can Not Be Included



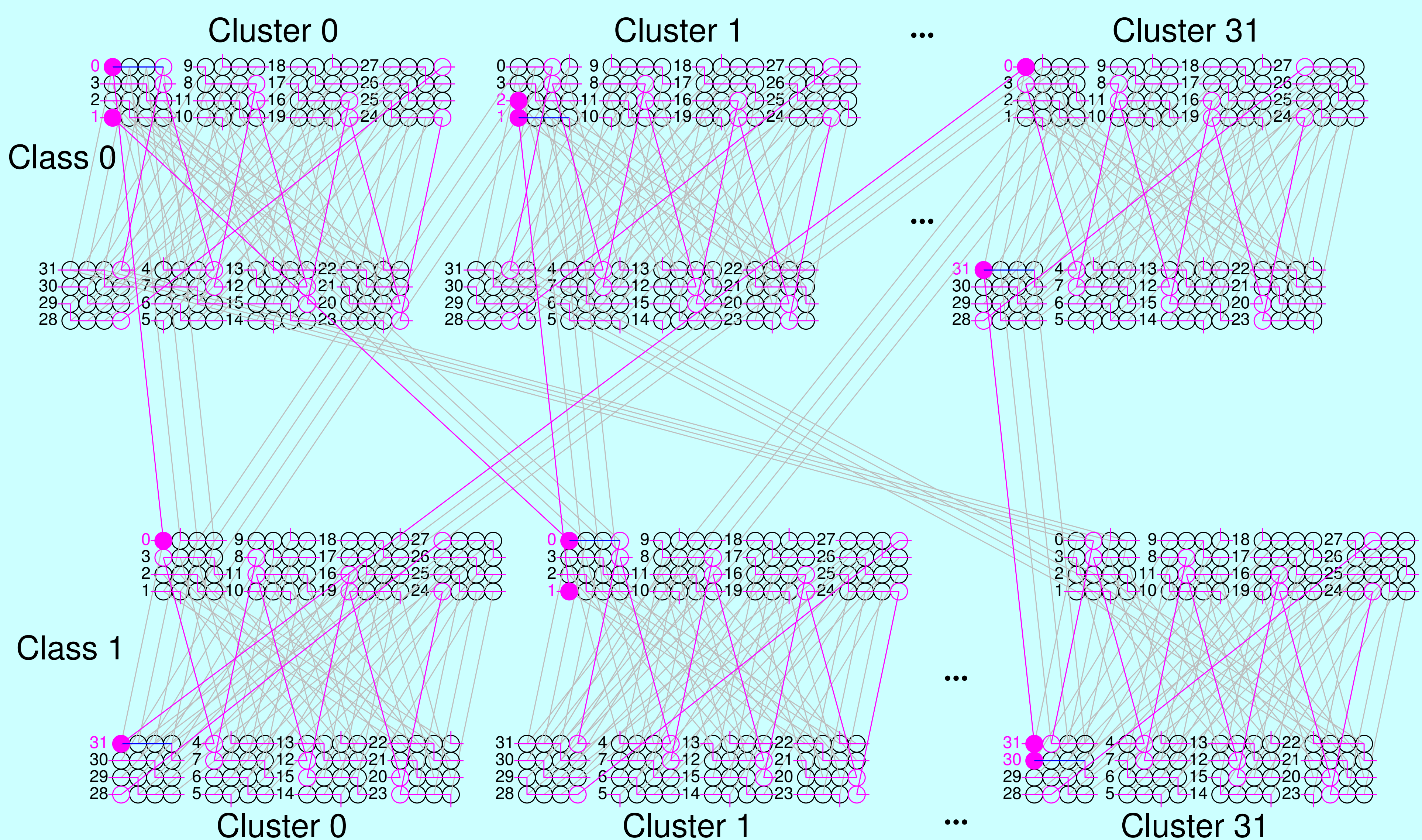
Routing in Super-Node i First



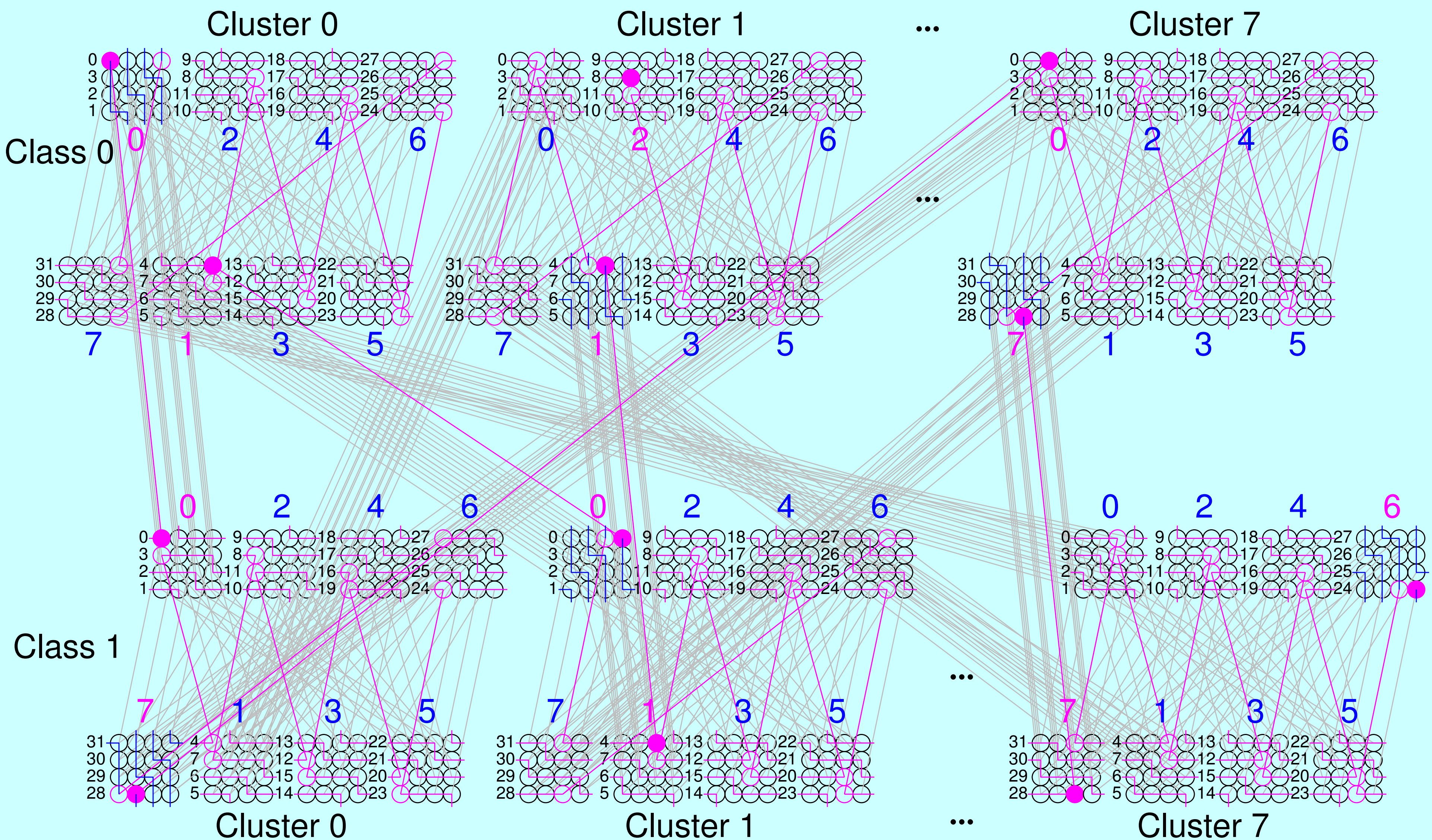
Part of Hamiltonian cycle at level 2



A Hamiltonian cycle in HDN(B,2,(4,4))



A Hamiltonian cycle in HDN(B,2,(4,16))



Algorithm for Finding a Hamiltonian Cycle in HDN

```
Algorithm 1: HDN_HC(HDN(B, k, S))
begin
    pc_0 = Hamiltonian cycle of base network;
    for j ← 1 to k do                                                          /* k levels */
        group nodes to super-nodes based on s_j;
        n_j = the number of super-nodes in HDN(B, k − 1, S);
        based on the cycle build at level j − 1,
        rename the super-node_id in HDN(B, j − 1, S) such
        that super-nodes i and (i + 1)%n_j are neighbors;
        u = 0; /* starting node_id in a super-node */
        for i ← 0 to n_j − 1 do                                   /* n_j clusters of each class */
            /* build a Hamiltonian path in cluster i of class 0 */
            /* based on j − 1 level Hamiltonian cycle pc_{j−1} */
            hp_i^0 = (0, i, i, u) → (0, i, (i + 1)%n_j, v);
            /* build a Hamiltonian path in cluster (i+1)%n_j of class 1 */
            /* based on j − 1 level Hamiltonian cycle pc_{j−1} */
            hp_i^1 = (1, (i + 1)%n_j, i, v) → (0, i, (i + 1)%n_j, w);
            u = w;                           /* end node_id → starting node_id of next cluster */
        endfor
        /* we get the j level Hamiltonian cycle pc_j */
        pc_j = ∅;
        for i ← 0 to n_j − 1 do                                   /* n_j clusters of each class */
            pc_j = pc_j ∪ hp_i^0 ∪ hp_i^1;
        endfor
    endfor
end
```

Conclusions

- The Hierarchical dual-net, HDN, has many attractive properties including
  - Small and flexible node-degree
  - Short diameter
  - Efficient routing and broadcasting
- We presented an algorithm for finding a Hamiltonian cycle in HDN with the base network of p-ary q-cube
- The HDN can be used as an efficient interconnection network of a supercomputer of the future generation

THX