1.) a.) Biswapped network B_n has node set $V(B_n) :=$
$\{(i,j,k) : i,j = 0,1,\dots,n-1; k = 0,1\}$
where i represents the node number
j represents the cluster number
k represents the bipartite part

Biswapped network B_n has edge set $E(B_n) :=$
$\{((i,j,k),(i+1,j,k)) : i,j = 0,1,\dots,n-1; k = 0,1\}$
$\cup \{((i,j,k),(j,i,k+1)) : i,j = 0,1,\dots,n-1; k = 0,1\}$
where addition is modulo n. The first set joins nodes within a cluster in a cycle and the second set joins nodes across the bipartite split.

b.) We prove that B_n has a hamiltonian cycle by construction

Abstractly, we form a path around every node in a cluster (in reverse order) before moving sequentially to the next cluster across the bipartite split and repeating.

Formally, $(0,0,0) \to (n-1,0,0) \to \dots \to (1,0,0)$
$\to (0,1,1) \to (n-1,1,1) \to \dots \to (1,1,1)$
$\to (1,1,0) \to (n-1,1,0) \to \dots \to (2,1,0)$
$\to (1,2,1) \to (0,2,1) \to \dots \to (2,2,1)$
$\to \dots$
$\to (n-1,n-1,0) \to (n-2,n-1,0) \to \dots \to (0,n-1,0)$
$\to (n-1,0,1) \to (n-2,0,1) \to \dots \to (0,0,1)$
$\to (0,0,0)$

Clearly, this cycle visits each node exactly once so is Hamiltonian

2.) a.) Star graph S_k has node set $V(S_k) :=$
$\{(x_0, x_1, x_2, x_3) : x_i \in \{0,1,2,3\};\ x_i \neq x_j\ \forall i \neq j\}$
i.e. the set of all permutations on k elements

Star graph S_k has edge set $E(S_k) :=$
$\{(x, x(01)), (x, x(02)), (x, x(0,3)) : x \in V(S_k)\}$
i.e. an edge is formed by swapping the first element of a node with another

b.) Star graph S_n has node set $V(S_n) :=$
$\{(x_0, x_1, \ldots, x_n) : x_i \in \{0, 1, \ldots, n-1\};\ x_i \neq x_j\ \forall i \neq j\}$

Star graph S_n has edge set $E(S_n) :=$
$\{(x, x(0i)) : x \in V(S_n);\ i \in \{1, 2, \ldots, n-1\}\}$

c.) The number of nodes (processors) in S_n is given by the size of the node set $|V(S_n)| = n!$ which grows faster than exponentially as n increases

Therefore, there is an increasingly larger gap in the number of processors (nodes) required between an S_n and an S_{n+1} network, resulting in poor scalability

The same goes for the number of edges (wiring)
$|E(S_n)| = (n-1)|V(S_n)| = (n-1)\,n!$

d.) $S_{n,k}$ gives more flexibility over the number of nodes ($|V(S_{n,k})| = n!/(n-k)!$) compared to S_n allowing the choice of an appropriate number of processors for the task. In fact, S_n is a special case of $S_{n,k}$ for $k = n$.

e.) There are two types of edges on $S_{n,k}$: $(u \in V(S_{n,k}))$

(i-edge) applying the swap $(0\ i)$ on u

(0-edge) swapping u_0 with some $x \in \{0,1,\dots,n-1\} \setminus u$

Using these two types of edges we can devise a routing algorithm from any node $(x_0 x_1 \cdots x_{n-1}) \in V(S_{n,k})$ to the node $(0,1,\dots,k-1) \in V(S_{n,k})$

```
For i in range(k):
    if i ∈ u: (internal cycle)
        (i-edge)
        (j-edge) with j = u.index(i)
        (i-edge)
    if i ∉ u: (external cycle)
        (i-edge)
        (0-edge)
        (i-edge)
```

All i-edge and 0-edge operations occur at u_0 so we must pivot / store $u_0 = 0$ at u_i using i-edge operations so it isn't removed

Some cases (e.g. $u_i = i$ already) cause the edge operations to return the same node, so in practise we only return distinct nodes at each step (*)

Example in $S_{7,4}$: $(7,5,3,1) \rightarrow$

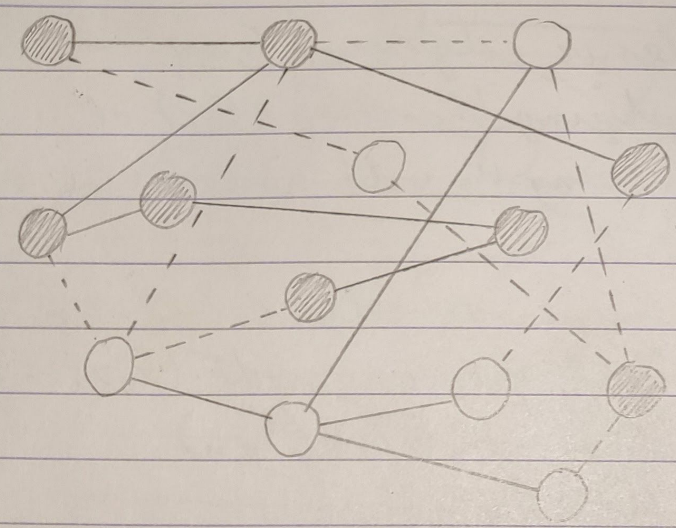
$(7,5,3,1) \rightarrow (0,5,3,1) \overset{(*)}{\rightarrow} (0,5,3,1) \rightarrow$ $i=0$ external

$(5,0,3,1) \rightarrow (1,0,3,5) \rightarrow (0,1,3,5) \rightarrow$ $i=1$ internal

$(3,1,0,5) \rightarrow (2,1,0,5) \rightarrow (0,1,2,5) \rightarrow$ $i=2$ external

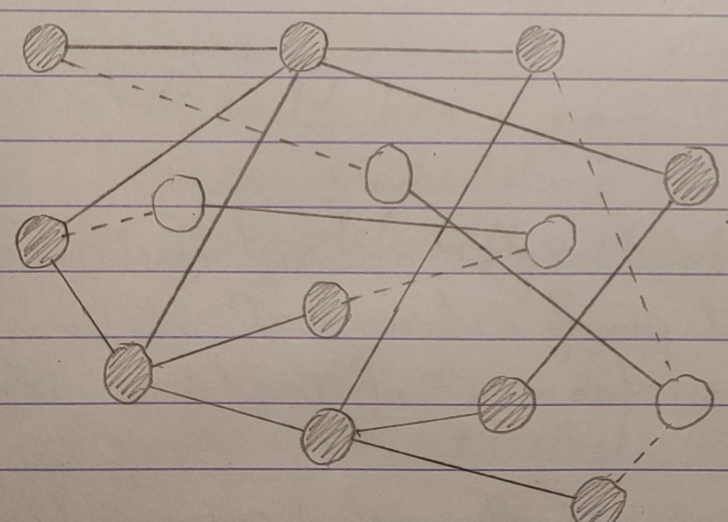
$(5,1,2,0) \rightarrow (3,1,2,0) \rightarrow (0,1,2,3)$ $i=3$ internal

3.) a.)



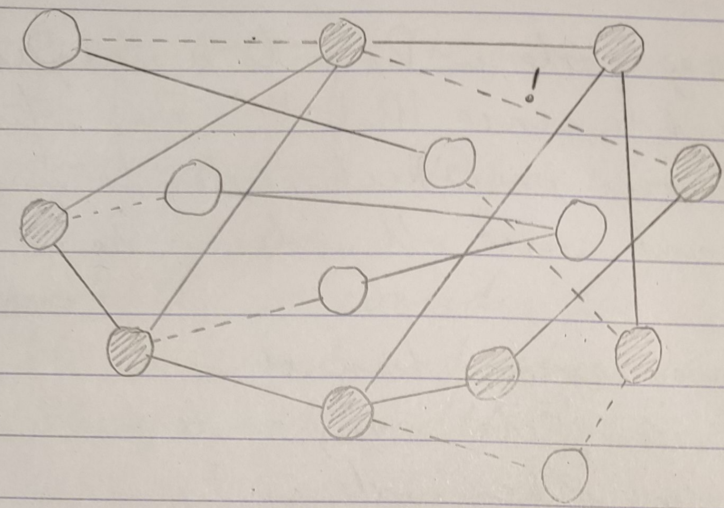
The above is a cut since it partitions the nodes into two sets (N_1, N_2) given by $(\bullet, \circ)$ s.t. $N^* = N_1 \cup N_2$, $N_1 \cap N_2 = \emptyset$, $N_1 \neq \emptyset \neq N_2$, and each channel in the cut joins a node in N_1 to a node in N_2.

b.)



The above is a cut since it partitions the nodes into two sets (N_1, N_2) given by $(\bullet, \circ)$ s.t. $N^* = N_1 \cup N_2$, $N_1 \cap N_2 = \emptyset$, $N_1 \neq \emptyset \neq N_2$, and each channel in the cut joins a node in N_1 to a node in N_2

c.)



The above is not a cut since the channel given by ! connects two nodes in the same set

k.) Let $C(N_1, N_2)$ be a bisection of N of width $\beta = |C(N_1, N_2)|$

Since $|N|$ is even, we have exactly $|N_1| = |N_2| = n/2$

Each node in N_1 has a path to each node in N_2 since φ is an embedding of D_n onto N of load 1 and D_n is complete (all distinct vertices connected by edge)

These paths must pass through $C(N_1, N_2)$ at least once so provide $|N_1| \cdot |N_2| = n^2/4$ channels of $C(N_1, N_2)$

Since D_n is a digraph, we get another set of paths from N_2 to N_1 providing $|N_1| \cdot |N_2|$ aswell

Hence, we have a total contribution of $n^2/4 + n^2/4 = n^2/2$

However, since φ has a congestion of at most γ, $C(N_1, N_2)$ has a total congestion of at most $\beta\gamma$

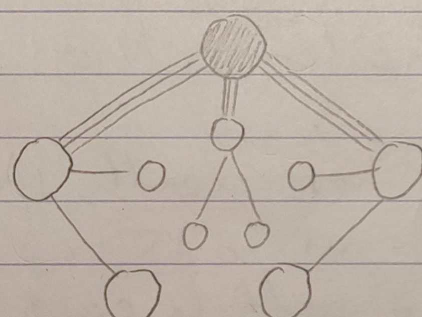
Therefore, $\beta\gamma \geq n^2/2$ to ensure this is satisfied

5.) Let φ map every node in D_{10} onto a unique node in P. This is possible since $|D_{10}| = |P|$. This satisfies load 1. Note, the choice of which nodes to map together is irrelevant since all nodes in D_{10} are equivalent.

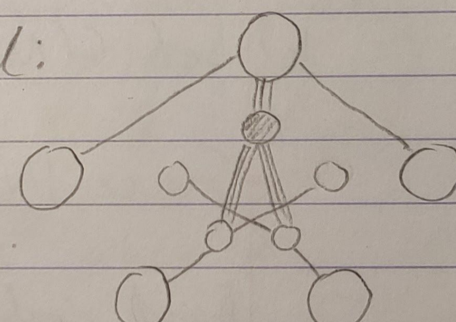
We are left to map every channel in D_{10} to a path in P. Since D_{10} is complete, every node in P must have a path to every other node in P.

Using the symmetry of P, we give two sets of paths for the external and the internal nodes

External:



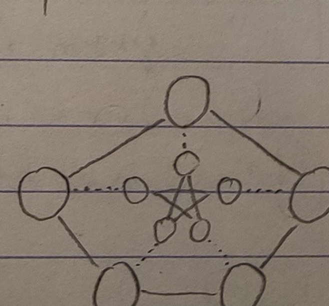
Internal:



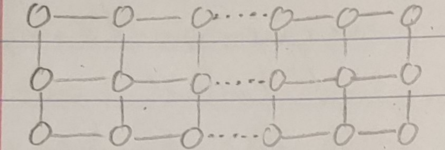
By rotating these two graphs 5 times and adding all the lines we get exactly 10 for each edge. Since P has two oppositely oriented channels for each edge, congestion $= 10/2 = 5$

From k.) with $n = 10$ and $\gamma = 5$ we get $\beta \geq 10$

A cut that separates the external nodes from the internal nodes gives a bisection with width 10 so $\beta \leq 10$



Therefore, $\beta = 10$ as required

6.)  The given bisection has width $B_{in} = 6$

We now show that this is minimal

Equivalently, we consider the undirected graph of the 3×6 mesh and show that the bisection above with width $B_G = B_{in}/2 = 3$ is minimal

There are no cuts of size 1 that split the nodes into 2 sets so there are no bisections of width $B_G = 1$

The only cuts of size 2 isolate a corner node from the rest of the graph but then $|N_1| \neq |N_2|$ so there are no bisections of width $B_G = 2$

7.) a.) Q_n^k: $\Theta_{ideal} \leq 2b\,B_C/|N| = 2b(k\,k^{n-1})/k^n = 8b/k$ ☆

$\Theta_{ideal} \leq |C|b/H_{ave}|N| = 2nk^n b/(n\lfloor k^2/4 \rfloor/k)\,k^n = 2bk/\lfloor k^2/4 \rfloor$

CCC_n: $\Theta_{ideal} \leq 2b\,B_C/|N| = 2b(2^{n-1})/n2^n = b/n$ ☆

$\Theta_{ideal} \leq |C|b/H_{ave}|N| = 3n2^n b/(7n/4)n2^n = 12b/7n$

In both cases, the first upper bound is lower

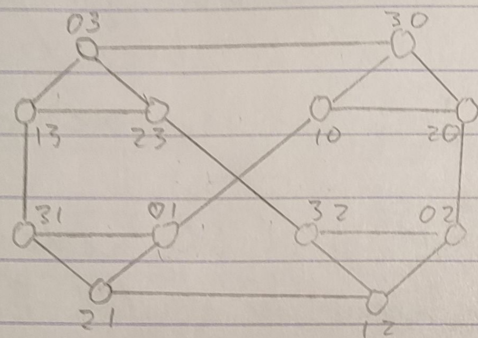
To use as many of our 900 processor (nodes) as possible we get $k=9, n=3$ for Q_n^k and $n=7$ for CCC_n

Then we get $\Theta_{ideal} \leq 8/9\, b$ for Q_n^k and $\Theta_{ideal} \leq 1/7\, b$ for CCC_n

Therefore, Q_n^k is the preferable network as it has the higher possible ideal throughput over CCC_n

b.) We assume the random traffic pattern, no contention, and perfect load balancing when calculating Θ_{ideal} which are not necessarily true in a non-idealised world

8.)



We show that given $a, b \in V(S_{4,2})$ with $a \neq b$, there exists an automorphism F s.t. $F(a) = b$

Let $V(S_{4,2}) \ni p = p_1 p_2$ where $p_1, p_2 \in \{0,1,2,3\}$ s.t. $p_1 \neq p_2$

so we get $a = a_1 a_2$ and $b = b_1 b_2$ as above

Let $F(p) = f(p_1) f(p_2)$ where f is the bijective map given by

$$\begin{cases} f(x) = b_i & \text{if } x = a_i \in a \\ f(x) = y & \text{if } x \in b \setminus a \text{ for } y \in a \setminus b \text{ (one-to-one mapping)} \\ f(x) = x & \text{if } x \notin a \cup b. \end{cases}$$

Clearly $F(a) = b$. We now show that F is an automorphism

Let $p = p_1 p_2$ and $q = q_1 q_2$ for $p, q \in V(S_{4,2})$ with $p \neq q$.
Then $p_1 \neq q_1$ so $f(p_1) \neq f(q_1)$ since f is bijective.
Therefore, $F(p) \neq F(q)$ as required

Let $V(S_{4,2}) \ni p = p_1 p_2$ be connected to $q \in V(S_{4,2})$.
Then either $q = p_2 p_1$ (1-edge) or $q = r p_2$ (0-edge) for $r \in \{0,1,2,3\} \setminus \{p_1, p_2\}$.
Then $F(p) = f(p_1) f(p_2)$ and either $F(q) = f(p_2) f(p_1)$ or $F(q) = f(r) f(p_2)$.
Therefore, $F(p)$ and $F(q)$ are still connected by either a (1-edge) or a (0-edge) as required

9.) a.) Given starting node 0, we find $H_{min}(0,x)$ for all nodes $x \in V(C_{n,3})$ i.e lengths of minimal paths

For $x \le n/2$, $H_{min}(0,x) = \lfloor x/3 \rfloor + x \% 3$
$x \ge n/2$, $H_{min}(0,x) = H_{min}(0, n-x)$

Here, $\lfloor x/3 \rfloor$ represents the number of 'skip' edges and $x \% 3$ represents the number of 'cycle' edges

Since $n = 6m+k$ is even, the node $n/2 = 3m+2$ occurs in both halves of $C_{n,3}$ so needs to be accounted for in the following sum

$$\sum_{x \in C_{n,3}} H_{min}(0,x) = \left(2 \cdot \sum_{x=0}^{n/2} H_{min}(0,x)\right) - H_{min}(0, n/2)$$

$$= \left(2 \cdot \sum_{x=0}^{3m+2} H_{min}(0,x)\right) - H_{min}(0, 3m+2)$$

$$= 2(0+1+2+1+2+3+\dots+(m-1)+(m)+(m+1)+(m)+(m+1)+(m+2)) - (m+2)$$

$$= 2(0+1+1+2+2+2+\dots+(m)+(m)+(m)+(m+1)+(m+1)+(m+2)) - (m+2)$$

$$= 2\left(3m+6+\sum_{y=2}^{m} 3y\right) - (m+2) = 2\left(3m+6+\tfrac{3}{2}m(m+1)-3\right) - (m+2)$$

$$= 3m^2 + 8m + k = (3m+2)(m+2)$$

By symmetry this is the same for all source nodes so $\sum_{x,y \in C_{n,3}} H_{min}(x,y) = n \sum_{x \in C_{n,3}} H_{min}(0,x)$

Therefore, $H_{ave} = \dfrac{\sum_{x,y \in C_{n,3}} H_{min}(x,y)}{n^2} = \dfrac{\sum_{x \in C_{n,3}} H_{min}(0,x)}{n}$

$$= \frac{(3m+2)(m+2)}{6m+k} = \frac{m+2}{2}$$

b.) $T_0 = (H_{ave} + 1)t_r + L/b$ (as we ignore time of flight T_w)

$C_{124,3}$ has $n = 124$ so $m = 20$ and $H_{ave} = 11$
Then $T_0 = 12 \cdot 20 + 1024/5 = 444.8$

$C_{124} = Q_1^{124}$ so $H_{ave} = \lfloor 124^2/4 \rfloor / 124 = 31$
Then $T_0 = 32 \cdot 20 + 1024/5 = 844.8$

Therefore, $C_{124,3}$ has better latency than C_{124}

10.) a.) Partition $S_{n,k}$ into subgraphs $S^k_{n-1,k-1}$ by fixing the last dimension k (the value of z)

This produces n subgraphs (since $z \in \{0, 1, \ldots, k-1\}$) of size
$$|S^k_{n-1,k-1}| = \frac{|S_{n,k}|}{n} = \frac{n!}{(n-k)!\,n} = \frac{(n-1)!}{(n-k)!} = \frac{(n-1)!}{((n-1)-(k-1))!} = |S_{n-1,k-1}|$$

Let $f: S^n_{n-1,k-1} \longrightarrow S_{n-1,k-1}$ by $(x_1, \ldots, x_{k-1}, z) \rightarrow (x_1, \ldots, x_{k-1})$
where $S_{n-1,k-1}$ is defined on $\{1, \ldots, n\} \setminus \{z\}$

Clearly f is a bijection so we have left to show that f is an isomorphism to claim $S^k_{n-1,k-1} \cong S_{n-1,k-1}$

Let $\underline{x} = (x_1, x_2, \ldots, x_{k-1}, z)$ and $\underline{y} = (y_1, y_2, \ldots, y_{k-1}, z)$ s.t. $\underline{x} \neq \underline{y}$
Then $\exists i \in \{1, 2, \ldots, k-1\}$ s.t. $x_i \neq y_i$ so $f(\underline{x}) \neq f(\underline{y})$

Let $\underline{x} = (x_1, x_2, \ldots, x_{k-1}, z)$ be connected to $\underline{y}$
Then either $\underline{y} = (x_i, \ldots, x_1, \ldots, x_{k-1}, z)$ (i-edge) or
$\underline{y} = (y_1, x_2, \ldots, x_{k-1}, z)$ (0-edge) for $y_1 \notin \{x_i\} \cup \{z\}$.
Then $f(\underline{x}) = (x_1, x_2, \ldots, x_{k-1})$ and either
$f(\underline{y}) = (x_i, \ldots, x_1, \ldots, x_{k-1})$ or $f(\underline{y}) = (y_1, x_2, \ldots, x_{k-1})$.
Then $f(\underline{x})$ and $f(\underline{y})$ are still connected by an (i-edge) or a (0-edge)

OV

b.) Infact, we can partition $S_{n,k}$ over any dimension into n subgraphs isomorphic to $S_{n-1,k-1}$

Let $x, y \in S_{n,k}$ s.t. $x \neq y$ and $x_i = y_i$ for some $i \in \{1, \dots, k\}$.
Then we partition on dimension i s.t. $x, y \in S^i_{n-1,k-1}$

Since $S^i_{n-1,k-1} \cong S_{n-1,k-1}$, there are $n-2$ node disjoint paths connecting x and y in $S^i_{n-1,k-1}$ (from the question)

x and y have degree $n-1$ in $S_{n,k}$ and $n-2$ in $S^i_{n-1,k-1}$.
Hence, there is one edge connecting x (resp. y) to some node not in $S^i_{n-1,k-1}$. This is given by the i-edge from x to $(x_i, \dots, x_1, \dots, x_{k-1}) \notin S^i_{n-1,k-1}$ (resp. y)

We construct a path from x to y without using nodes in $S^i_{n-1,k-1}$ so that the $n-2$ node disjoint paths from the question are still valid giving $n-1$ total node disjoint paths from x to y in $S_{n,k}$

Path: Leave $S^i_{n-1,k-1}$ from x using the i-edge identified above
Follow a routing algorithm similar to 2.) e.) to $(y_i, \dots, y_1, \dots, y_{k-1})$ ensuring that dimension i never contains x_i or y_i otherwise we are back in $S^i_{n-1,k-1}$
Enter $S^i_{n-1,k-1}$ to y using the i-edge identified above

Let $x, y \in S_{n,k}$ s.t. $x \neq y$ but $x_i \neq y_i$ for any $i \in \{1, \dots, k\}$
so we cannot simply partition on dimension i