Deterministic Multiplexing of NoC on Grid CMPs

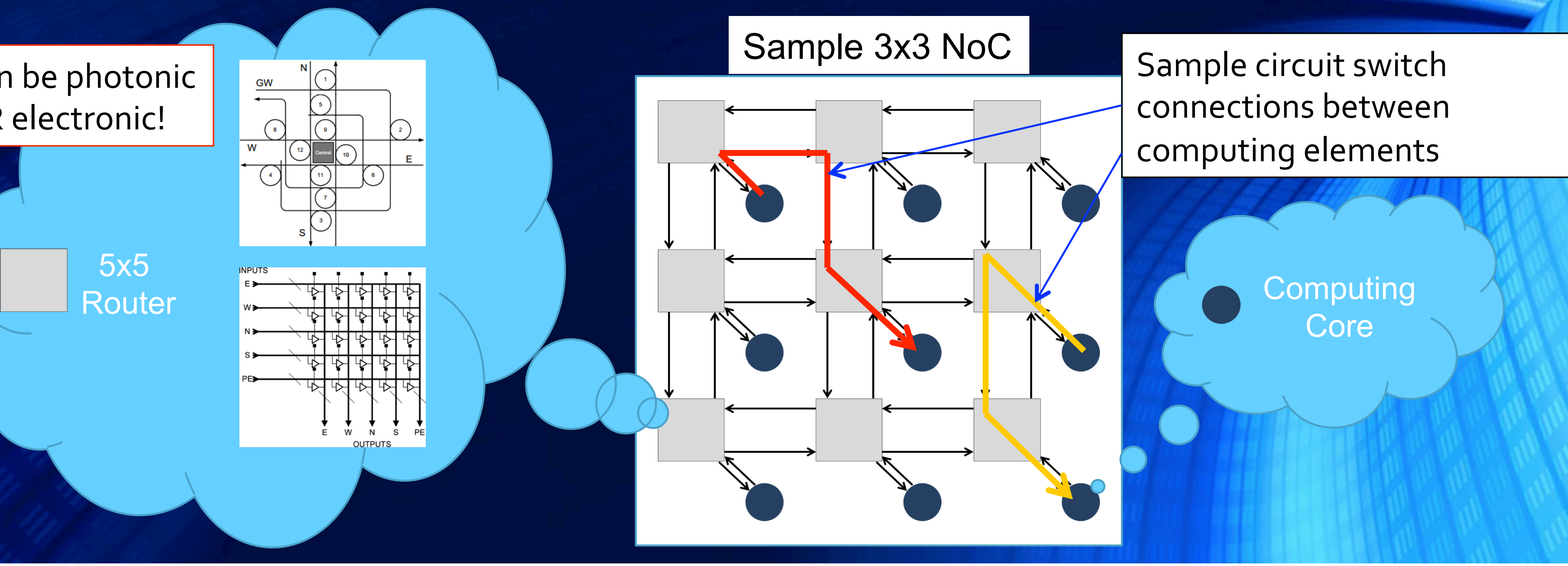
JOHN CARPENTER
RAMI MELHEM
AUGUST 21, 2013

Overview

- Definition of problem
- Multiplexing strategies
- Previous works
- Motivation
- Solution to Problem
- Results

Definition of Problem

- The Network
  - An NoC constructed as a grid of routers (switches) where each router is connected by unidirectional links to its four neighbors and to a local computing core



Definition of Problem: Network Contention

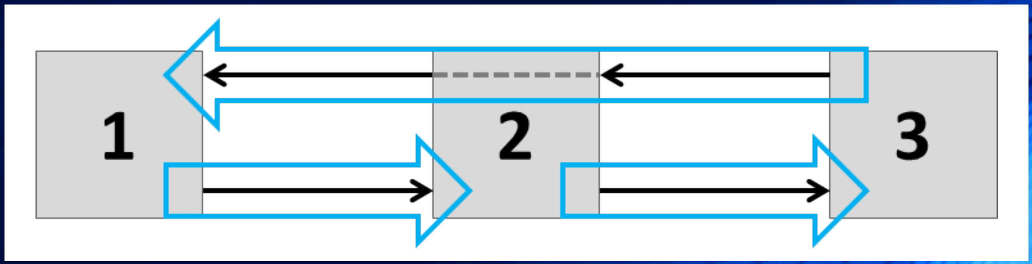
- In order to facilitate all-to-all connectivity, multiplexing is necessary due to the contentions inherent in the network
  - Types of contention on the network
    - Link Contention: only one connection can use a waveguide per multiplexing slot
    - Sender Contention: a node on the network can only send one message per multiplexing slot
    - Receiving Contention: a node on the network can only receive one message per multiplexing sl
- Using Multiplexing, we can achieve all-to-all connectivity by creating "slots of valid network configurations on the network

  - A valid network configuration is a set of connections that can be realized in one multiplexing slot on the network without contention

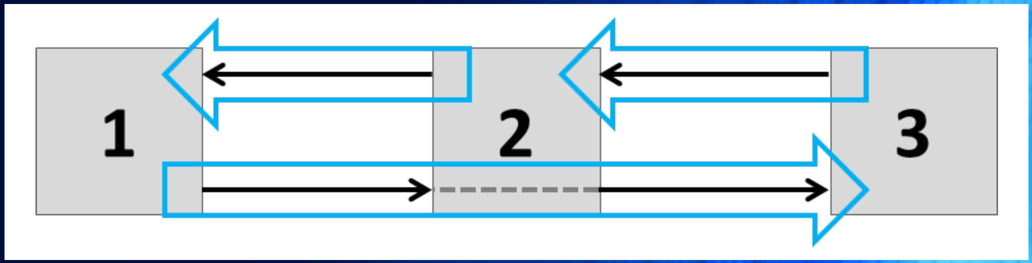
Multiplexing Strategies

- Strategy 1: Time Division Multiplexing (TDM)
  - Divide the communication paths needed for all-to-all connectivity amongst multiple <u>discrete time slots</u>
  - Example (1x3 network):

<u>Time slot 1 of 2</u>
1 sends to 2
2 sends to 3
3 sends to 1



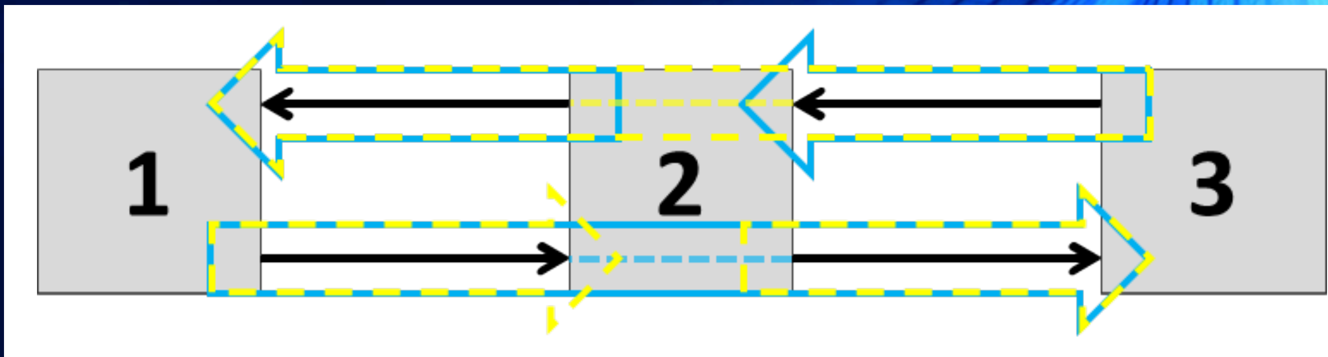
<u>Time slot 2 of 2</u>
1 sends to 3
2 sends to 1
3 sends to 2



Multiplexing Strategies (in case of optics)

- Strategy 2: Wavelength Division Multiplexing (WDM)
  - Divide the communication paths needed for all-to-all connectivity amongst multiple wavelengths (simultaneously)
  - Example (1x3 network):

Wavelength 1 of 2
1 sends to 2
2 sends to 3
3 sends to 1



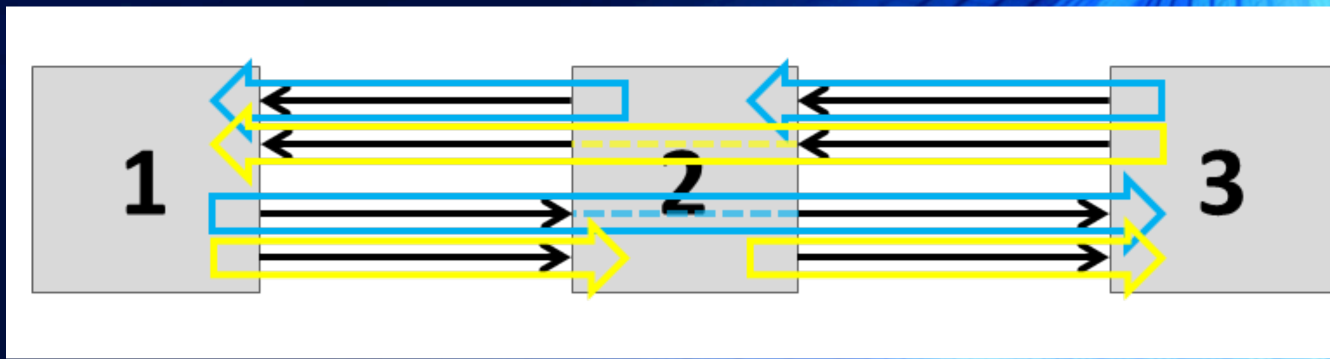
Wavelength 2 of 2
1 sends to 3
2 sends to 1
3 sends to 2

Multiplexing Strategies

- Strategy 3: Space Division Multiplexing
  - Divide the communication paths needed for all-to-all connectivity amongst multiple physical planes/links (in case of optics, waveguides)
  - Example (1x3 network):

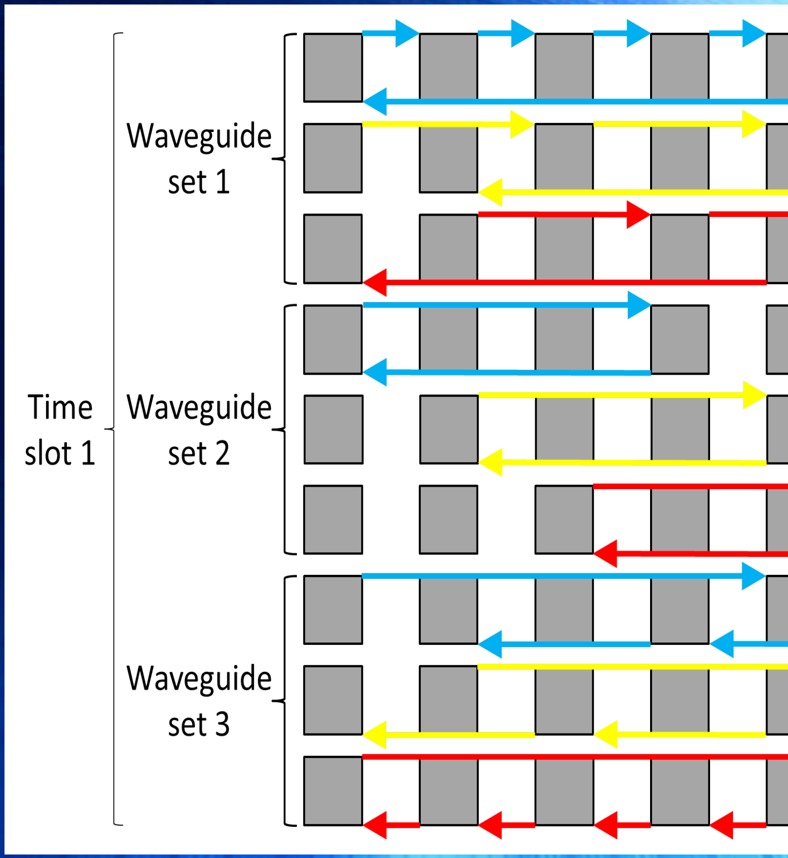
Plane/link 1 of 2
1 sends to 2
2 sends to 3
3 sends to 1

Plane/link 2 of 2
1 sends to 3
2 sends to 1
3 sends to 2



Combining Multiplexing Strategies

- Consider a 1x6 network with the following schedule
  - 9 multiplexing slots
  - If TDM is the sole multiplexing method used, 9 time slots are needed
  - If 3 wavelengths are available to use, only 3 time slots are needed
  - If 3 wavelengths and 3 waveguides are available to use, only 1 time slot is needed
- Combining multiple multiplexing strategies with TDM reduces number of time slots and communication latency

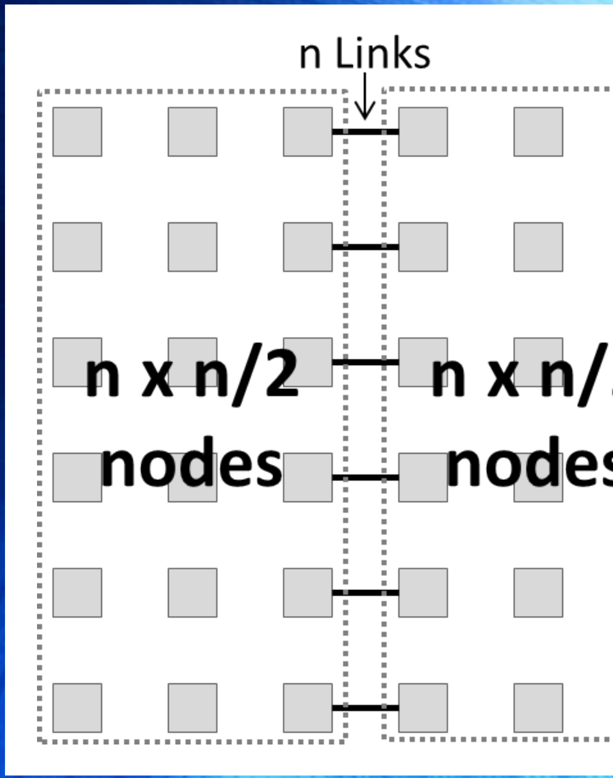


Problem Description

- In this work, we propose creating a regular static schedule for an *nxn* network by systematically dividing the connections into slots of valid network configurations
- Regularity in scheduling is useful for 3 reasons
  - Design automation
  - Proof of scalability
  - Low run time of scheduling algorithm
- Regularity of scheduling will be necessary for our future work of efficiently combining slots into TDM, WDM and SDM.

Baseline: Theoretical Minimal Scheduling

- Using the bisectional width of the network, we can determine that for an nxn network, the minimal multiplexing degree is given as follows
  - For n is even: $n*\frac{n}{2}$ senders must communicate with $n*\frac{n}{2}$ receivers via n links
    - $\frac{\left(n*\frac{n}{2}\right)*\left(n*\frac{n}{2}\right)}{n}=\frac{n^3}{4}$ minimal multiplexing degree
    - Similar analysis can be done for odd-sized networks
- The theoretical min will be used as a baseline against our systematic multiplexing algorithm



Previous work

- In previous work, [1]Hendry et al. created the proposed optical network and developed a nondeterministic algorithm to create a static scheduling of time slots to achieve all-to-all connectivity
  - Ran experiments with real benchmarks to determine that their circuit switched network was a viable alternative to a packet switched network
- This work used a Genetic Algorithm, which uses heuristics iteratively to densely pack the network with non-conflicting connections
  - No regularity in how connections are chosen
  - No guarantee on scalability of solution
  - Slow run time to determine their algorithm's best schedule
  - Hard to combine multiplexing strategies

1. G. Hendry, J. Chan, S. Kamil, L. Oliker, J. Shalf, L. Carloni, and K. Bergman, "Silicon Nanophotonic Network-on-Chip using TDM Arbitration," Proceedings of IEEE Symposium on High-Performance Interconnects, 2010.

Motivation: Theoretical Minimal Scheduling vs. Results of Prior Work

- The following table shows the theoretical minimum scheduling of a 4x4, 6x6, and 8x8 network next to the multiplexing degree proposed in previous work[1]

Size of Network	4x4	6x6	8x8	9x9	10x10
Theoretical Min.	16	54	128	180	250
Genetic Algorithm	18	61	142	*	*

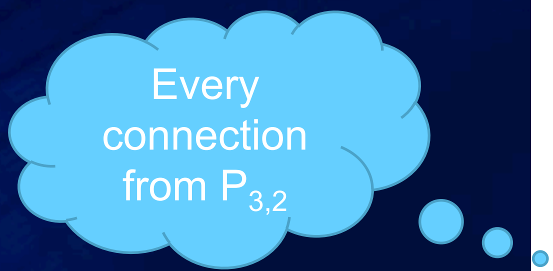
  - There is a gap in terms of multiplexing slots between the theoretical minimum scheduling a genetic algorithm
  - * The genetic algorithm, due to its run time was only run on these 3 network sizes

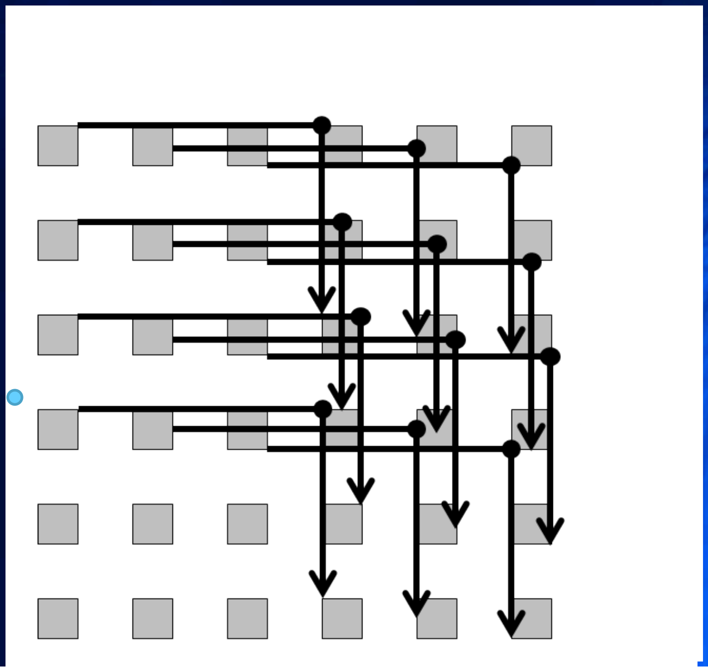
- Our goal is to lessen the gap, and provide a <u>regular</u> way of scaling our solution to l networks

1. G. Hendry, J. Chan, S. Kamil, L. Oliker, J. Shalf, L. Carloni, and K. Bergman, "Silicon Nanophotonic Network-on-Chip using TDM Arbitration," Proceedings of IEEE Symposium on High-Performance Interconnects, 2010.

Scheduling method: Connection Patterns

- Divide connections into connection patterns, designated $P_{a,b}$
  - Connection pattern $P_{a,b}$ are those connections which have identical offsets, <a,b>
- Idea: Use regularity the of connection patterns to systematically create a schedule for all-to-all communication



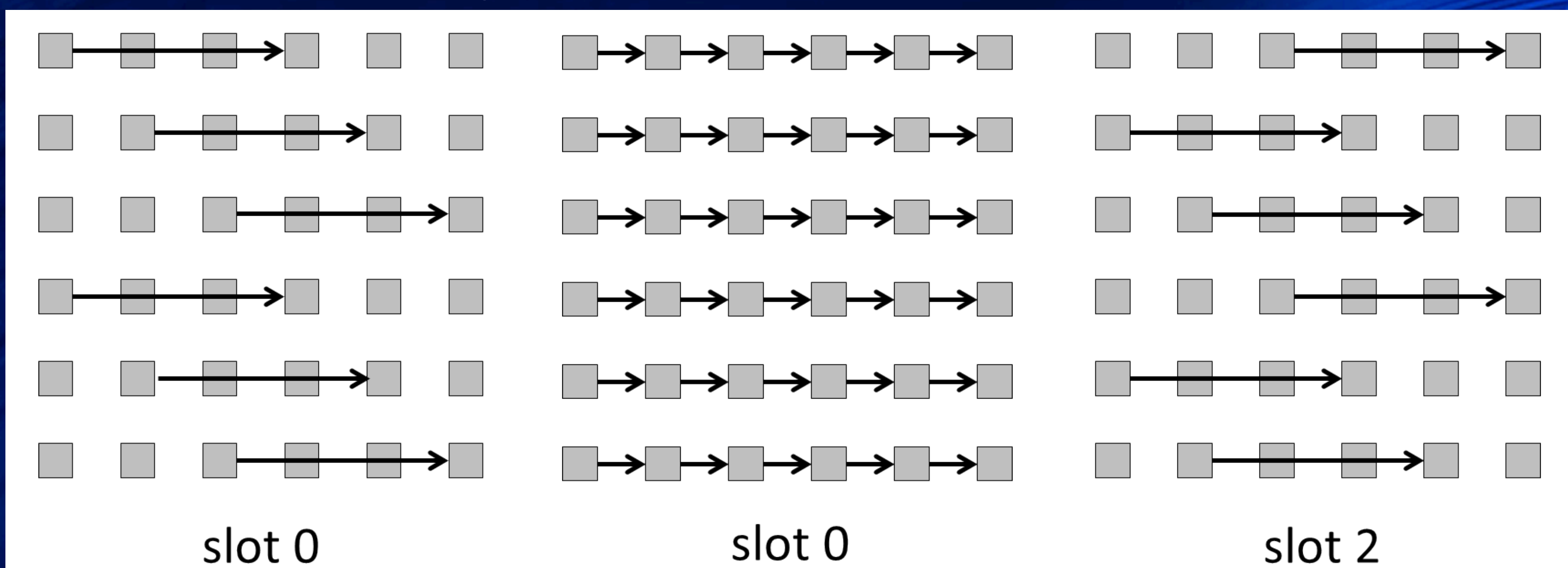


1D Connection Patterns (a=0 or b=0)

- Theorem: All 1D connections in connection pattern $P_{a,0}$ can be scheduled in min(a,n-a) slots
  - Note: “slots” may refer to TDM, WDM, or SDM multiplexing slots
  - Similar idea for 1D patterns in $P_{0,b}$

$P_{3,0}$ scheduling example in 3 slots

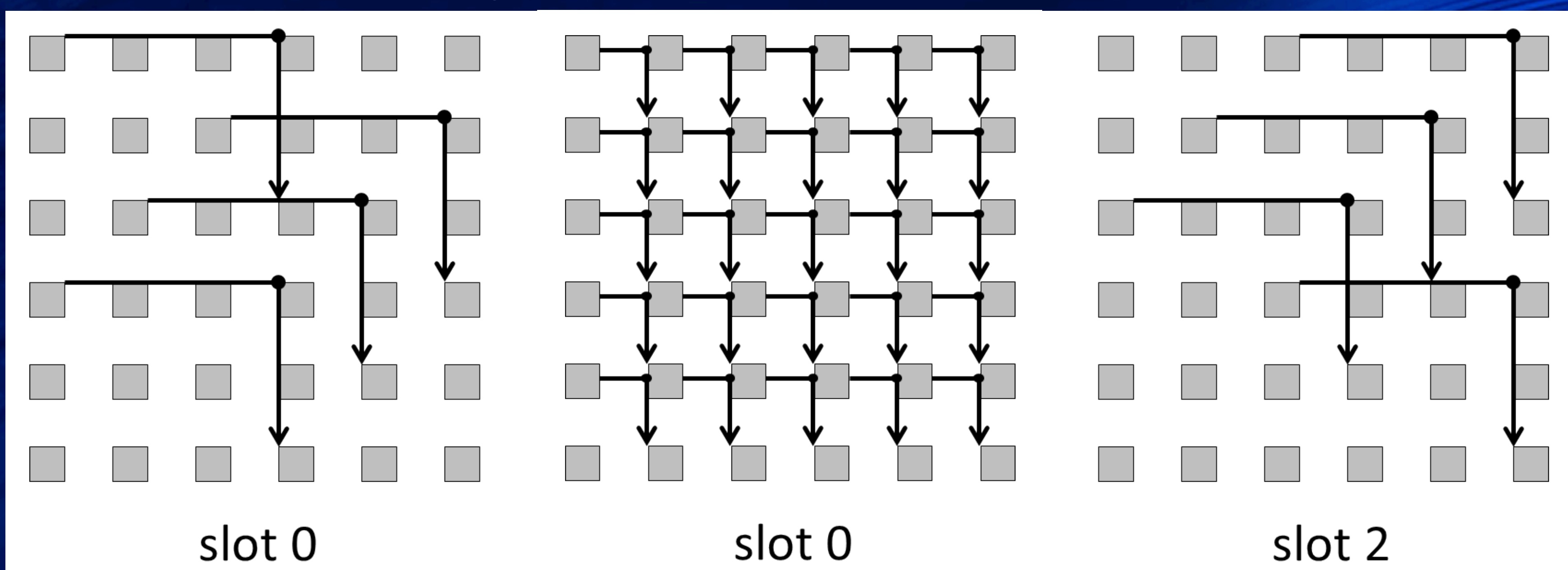
$P_{1,0}$ scheduling example in 1 slot



2D Connection Patterns (a>0 and b>0)

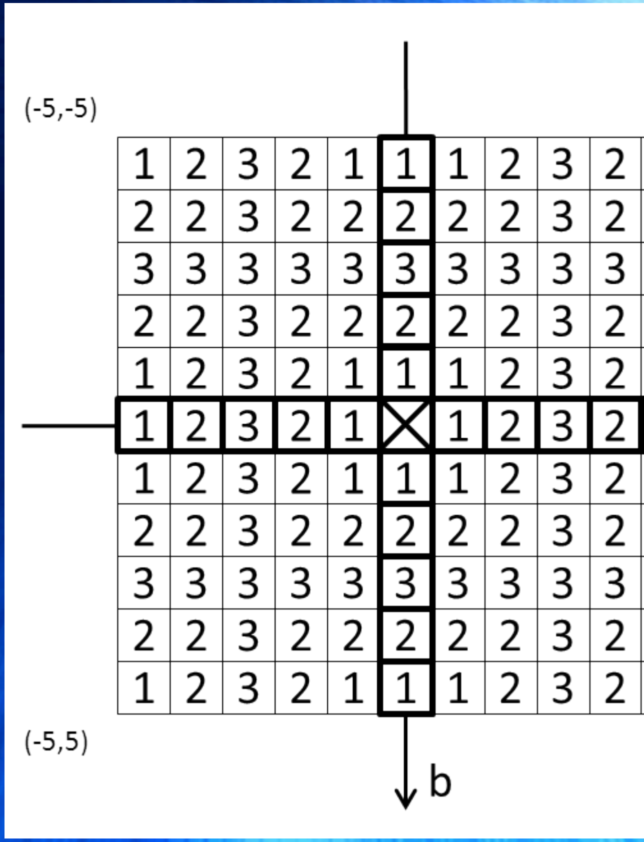
- Theorem: Generally, all connections in connection pattern $P_{a,b}$ can be scheduled in max(min(a,n-a),min(b,n-b)) slots
  - Note: "slots" may refer to TDM, WDM, or SDM multiplexing slots
  - 1D connections also satisfy this equation

$P_{3,2}$ scheduling example in 3 slots



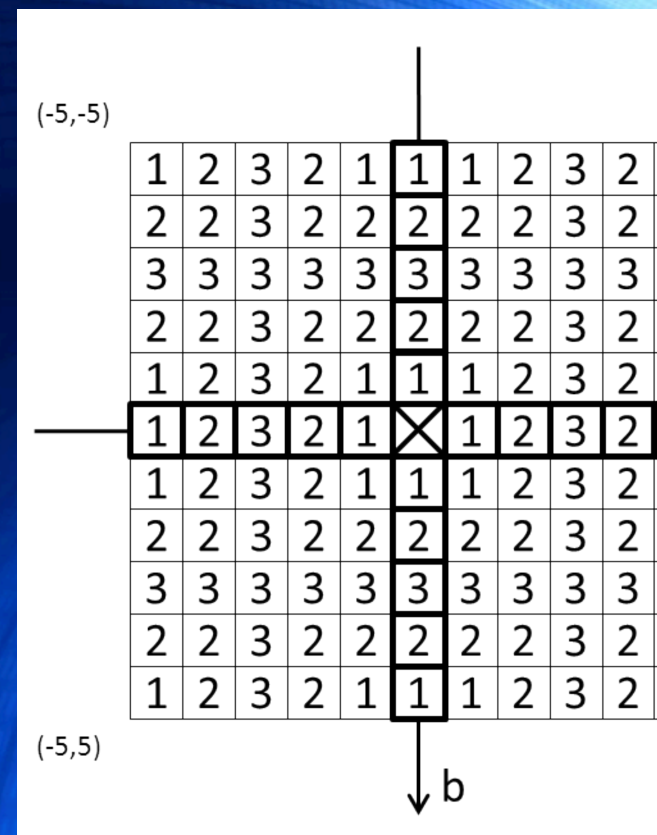
Connection Profile

- We define *U* to be the union of all connection patterns, and name it the <u>connection profile</u>
- Pictorially, we represent the connection profile by grid on the right for a 6x6 grid network
- To represent the multiplexing degree of each connection pattern, we fill in the profile with the corresponding values from our theorem
  - Multiplexing degree of $P_{a,b}$ = max(min(a,n-a),min(b,n-b))
- Using this information, we can determine the multiplexing degree for the trivial solution of separately multiplexing each connection pattern



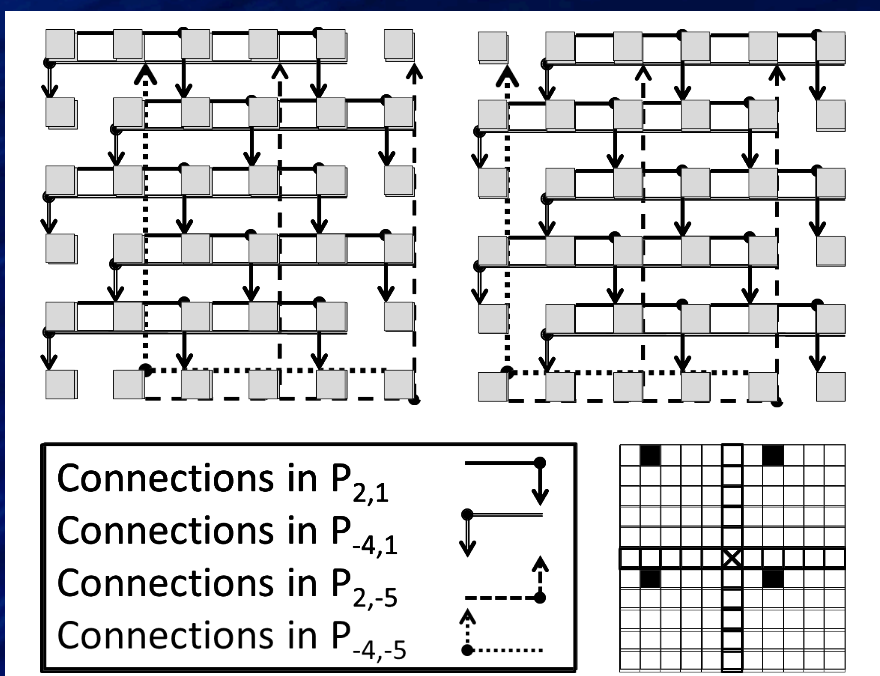
Trivial Scheduling Solution

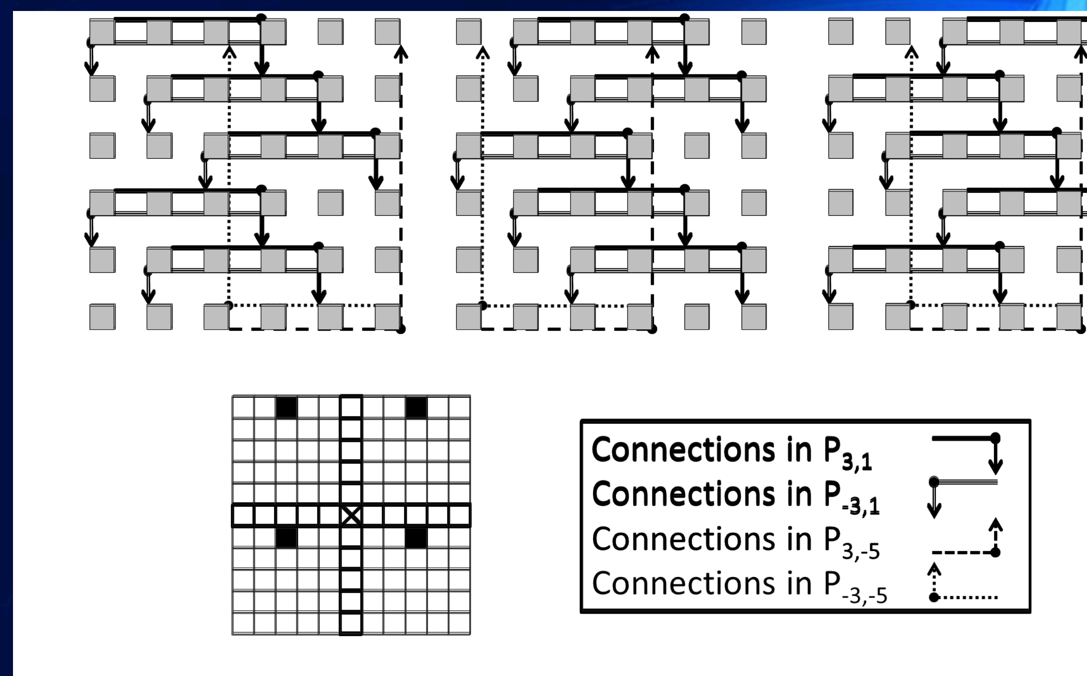
- By separately multiplexing each of the patterns, we can achieve a multiplexing degree of 256 for a 6x6 mesh
  - This solution is found by summing the multiplexing degrees of each of the connection patterns in *U*
- The theoretical minimal scheduling of a 6x6 is 54 multiplexing slots, so we can do better
- Next step: combine connection patterns together to lower the multiplexing degree of the system



Combining Connection Patterns

- To lessen the multiplexing degree of the system, we define a Connection Group: $G_{a,b}$ to be the union of the following 4 patterns
  - $G_{a,b} = P_{a,b} \cup P_{n-a,b} \cup P_{a,n-b} \cup P_{n-a,n-b}$: a > 0, b > 0
    - We will ignore patterns where a = 0, or b = 0 for the time being
  - Examples: $G_{2,1}$ and $G_{3,1}$ for a 6x6 network



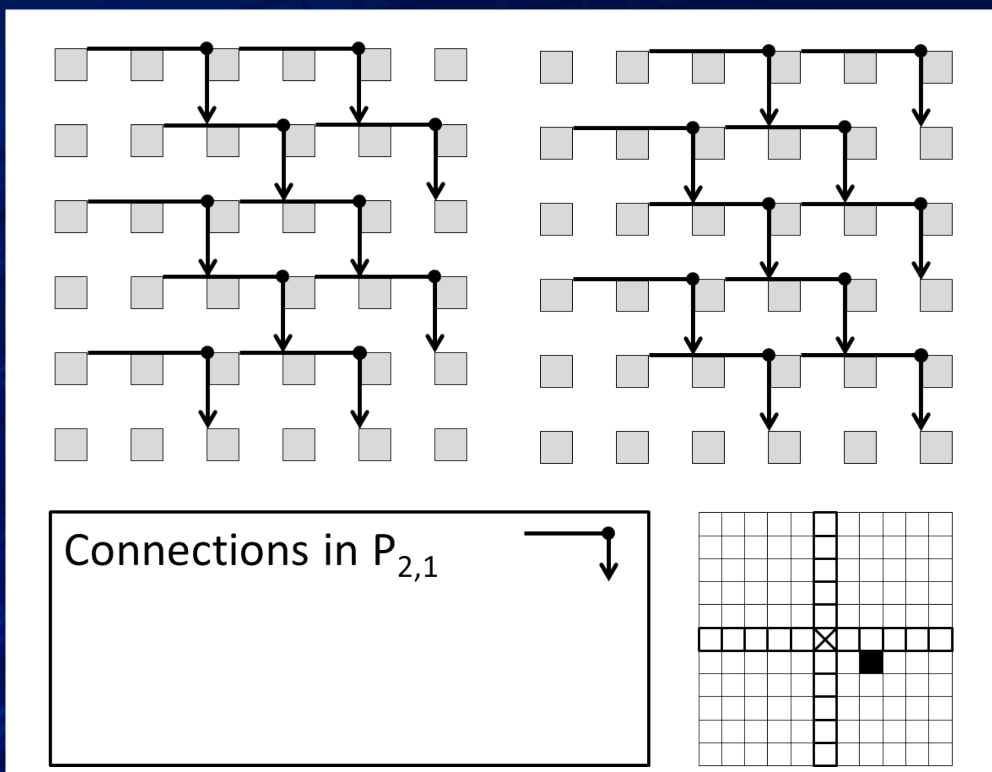


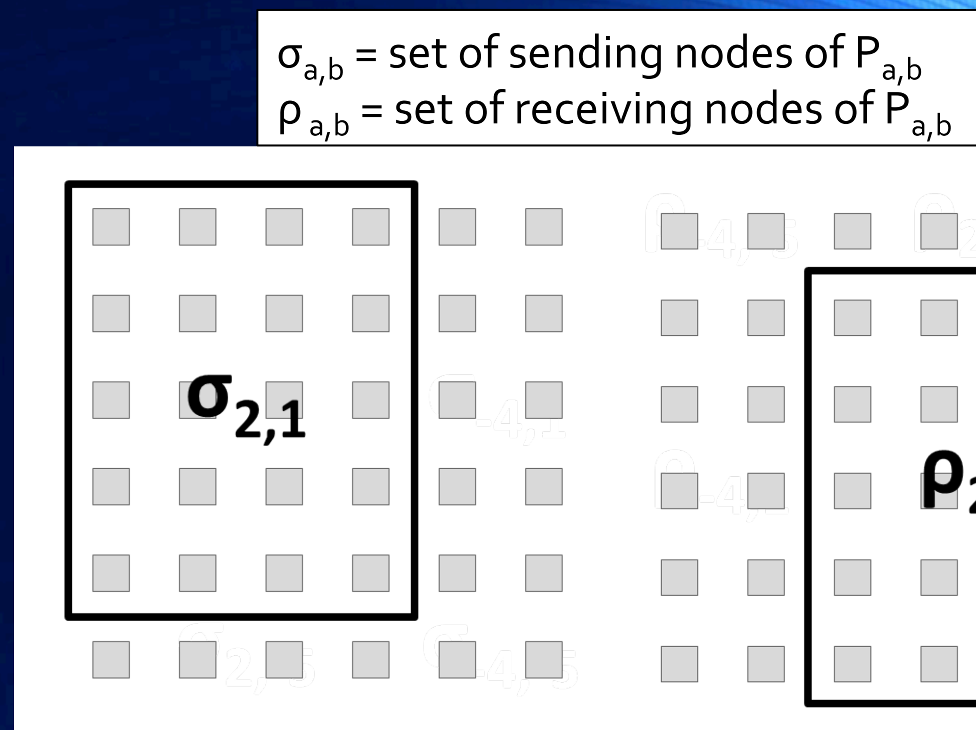
Connection Groups

- Now we must show that patterns within a connection group are disjoint
  - From our definition of "valid network configuration", we have the following criteria for non-conflicting connections, and by extension connection patterns
    - No Link Conflicts: only one connection can use a waveguide at a time
    - No Sender Conflicts: a node on the network can only send one message at a time
    - No Receiving Conflicts: a node on the network can only receive one message at a time
- Since we know connections within connection patterns do not conflict, we only need to show that the connection patterns of a group do not conflict with each other

Connection Groups

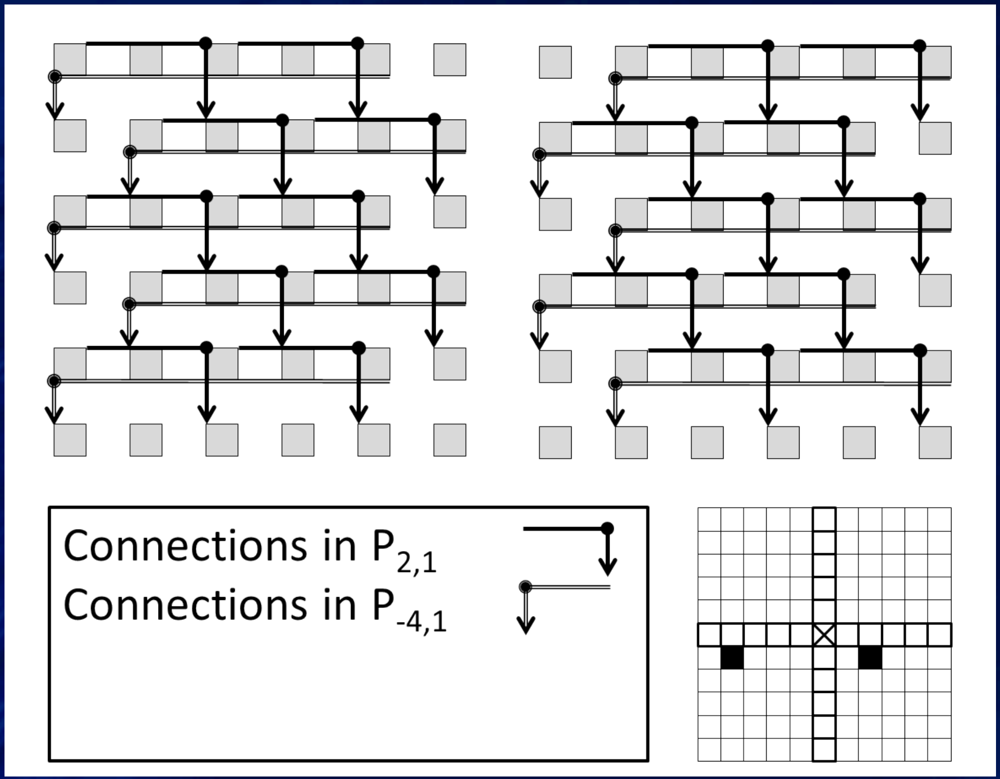
- No sending or receiving contention
  - Example: $G_{2,1}$



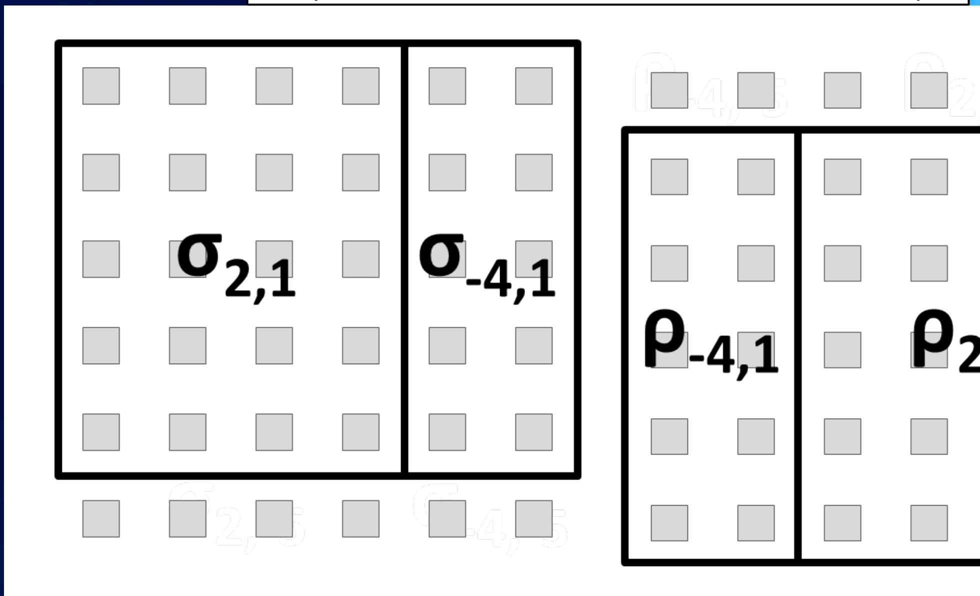


Connection Groups

- No sending or receiving contention
  - Example: $G_{2,1}$

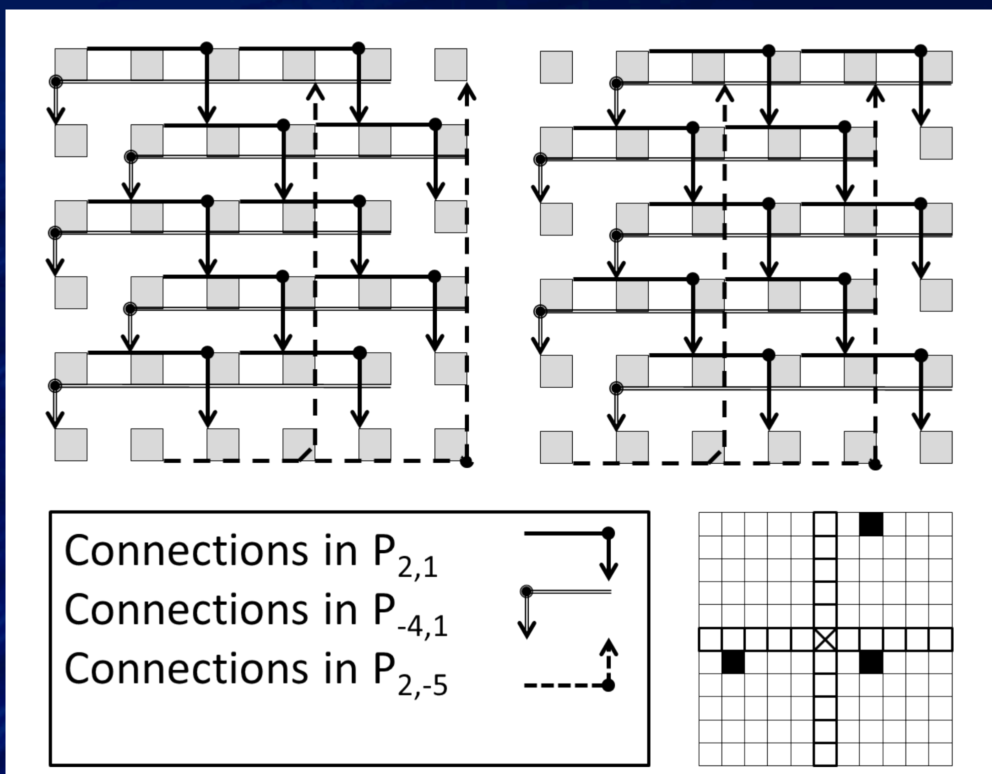


$\sigma_{a,b}$ = set of sending nodes of $P_{a,b}$
$\rho_{a,b}$ = set of receiving nodes of $P_{a,b}$

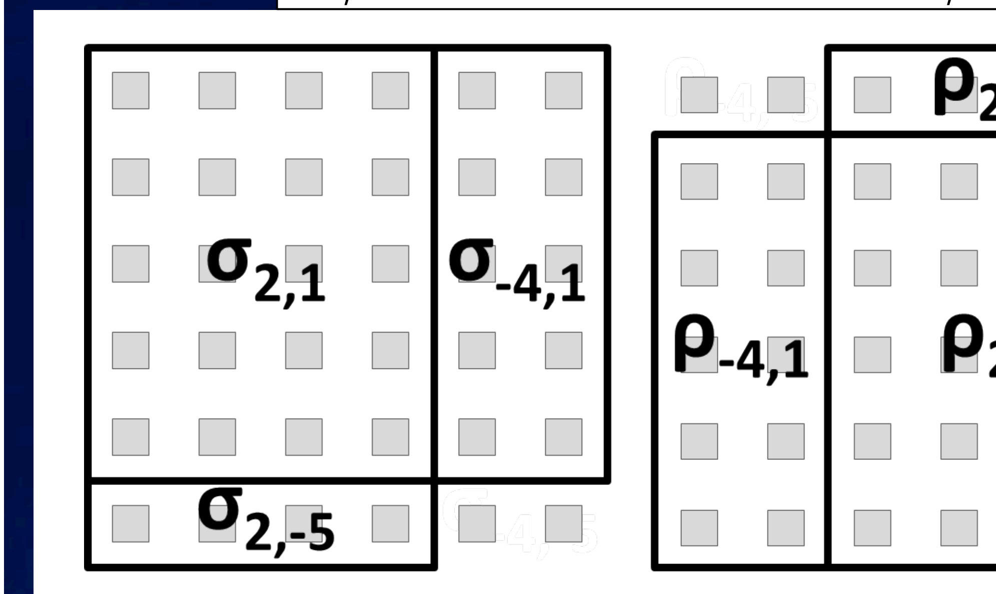


Connection Groups

- No sending or receiving contention
  - Example: $G_{2,1}$

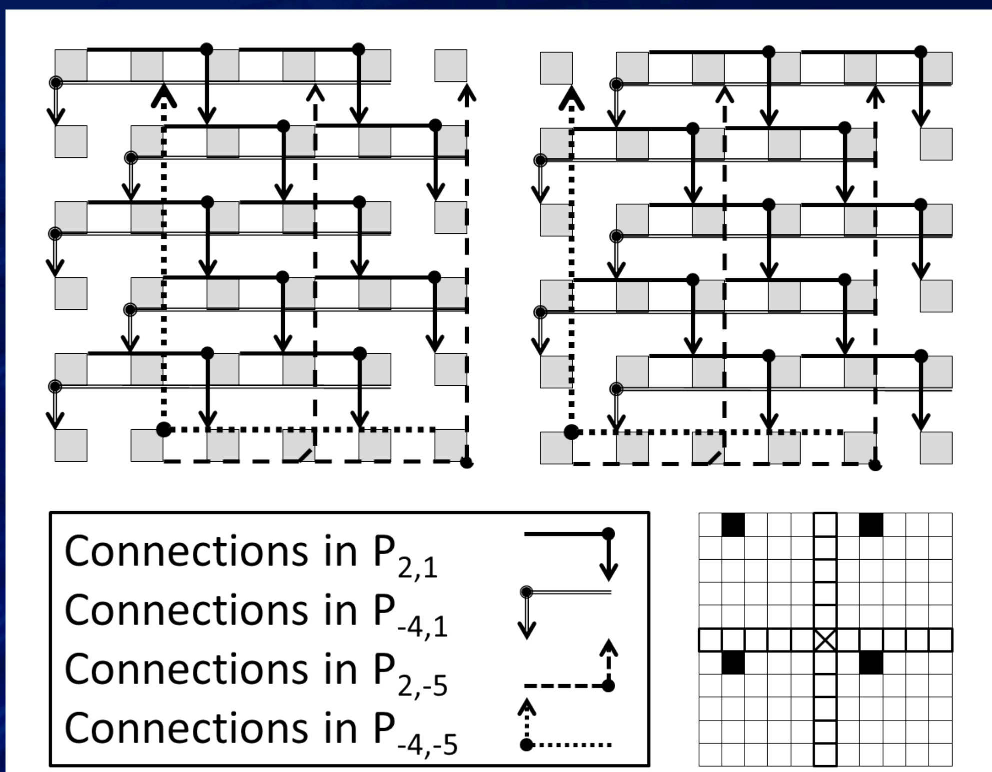


$\sigma_{a,b}$ = set of sending nodes of $P_{a,b}$
$\rho_{a,b}$ = set of receiving nodes of $P_{a,b}$

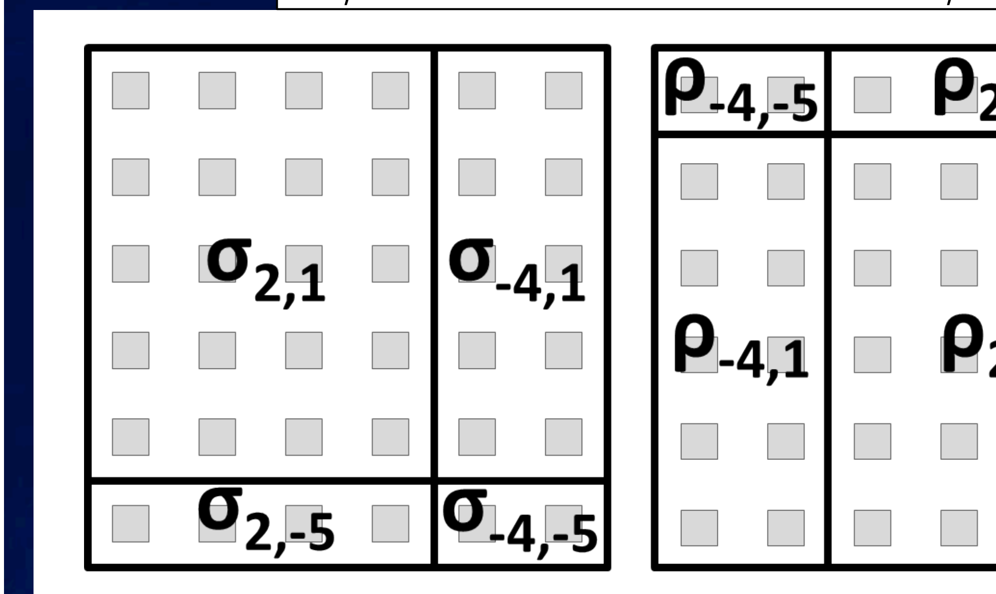


Connection Groups

- No sending or receiving contention
  - Example: $G_{2,1}$

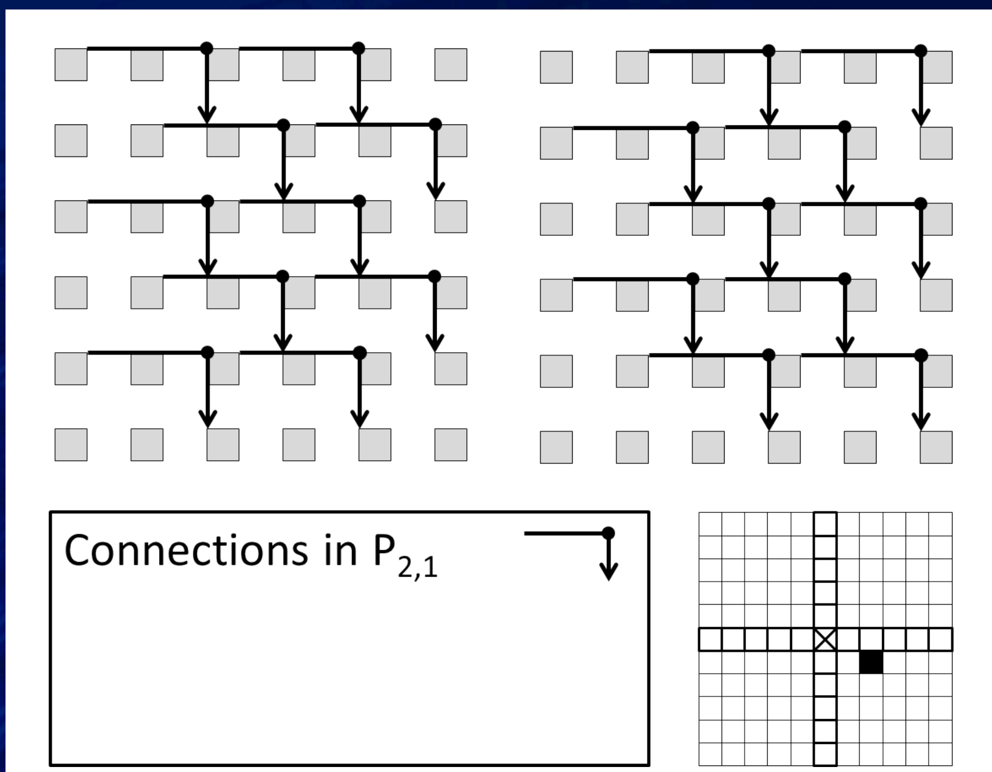


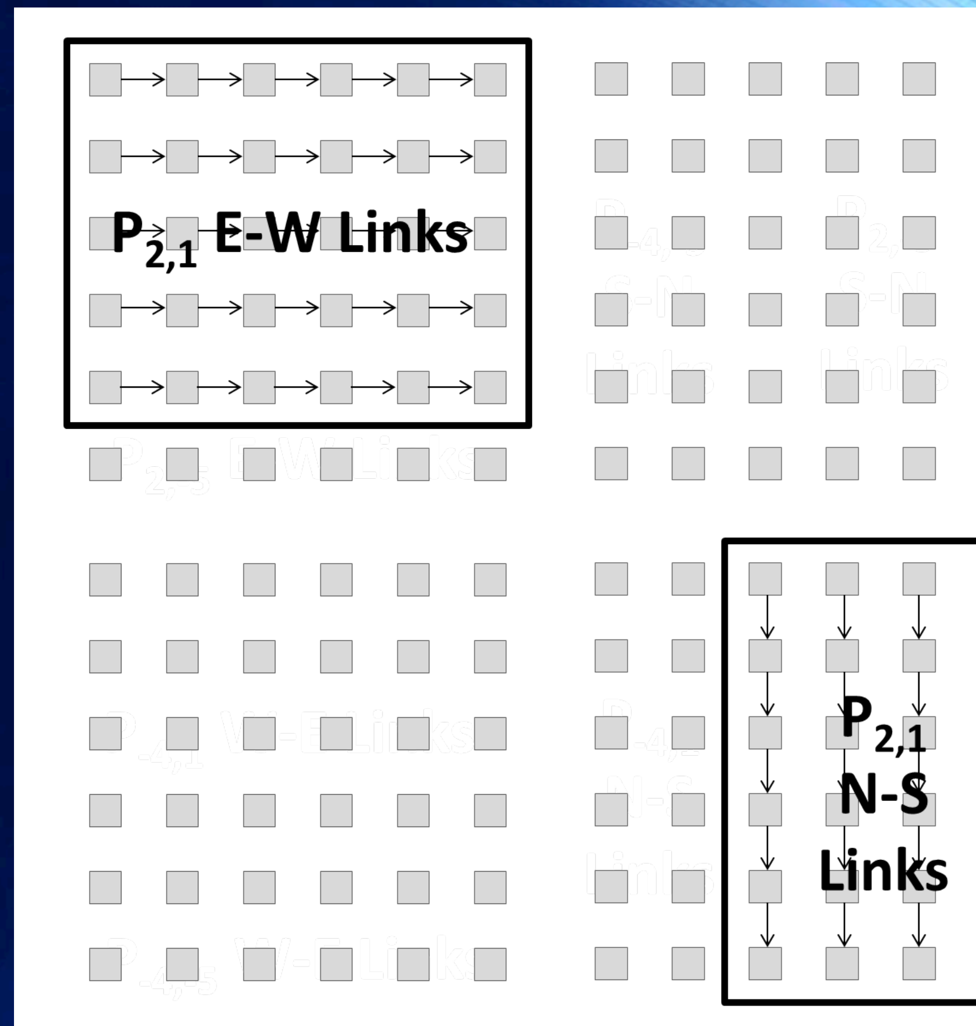
$\sigma_{a,b}$ = set of sending nodes of $P_{a,b}$
$\rho_{a,b}$ = set of receiving nodes of $P_{a,b}$



Connection Groups

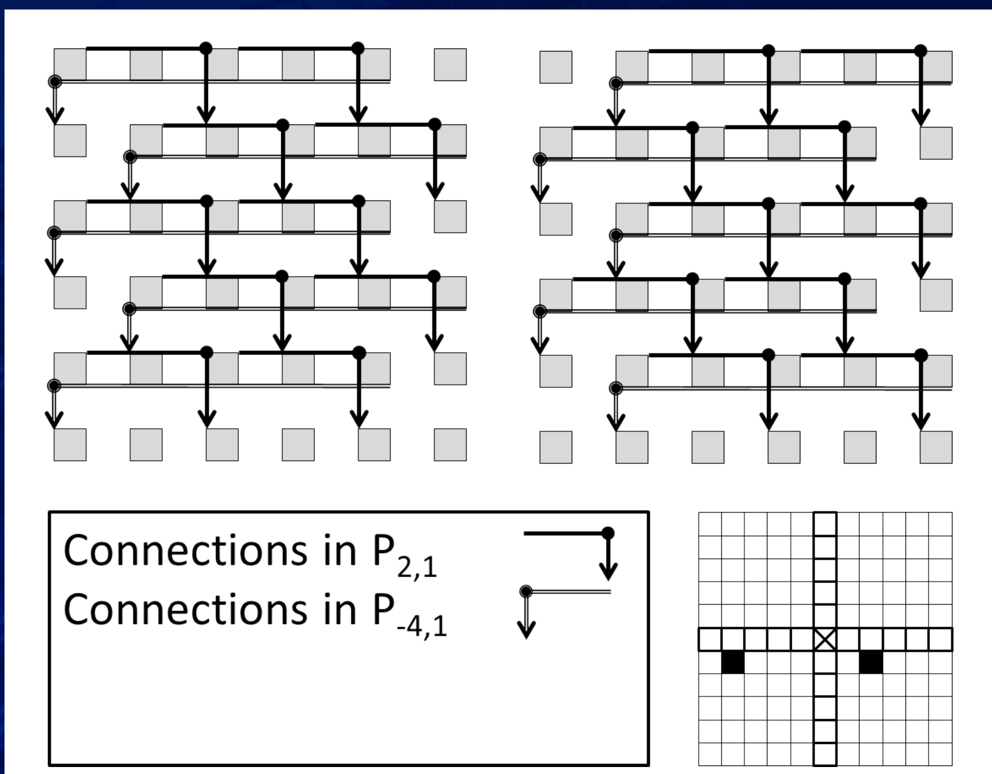
- No link contention
  - Example: $G_{2,1}$

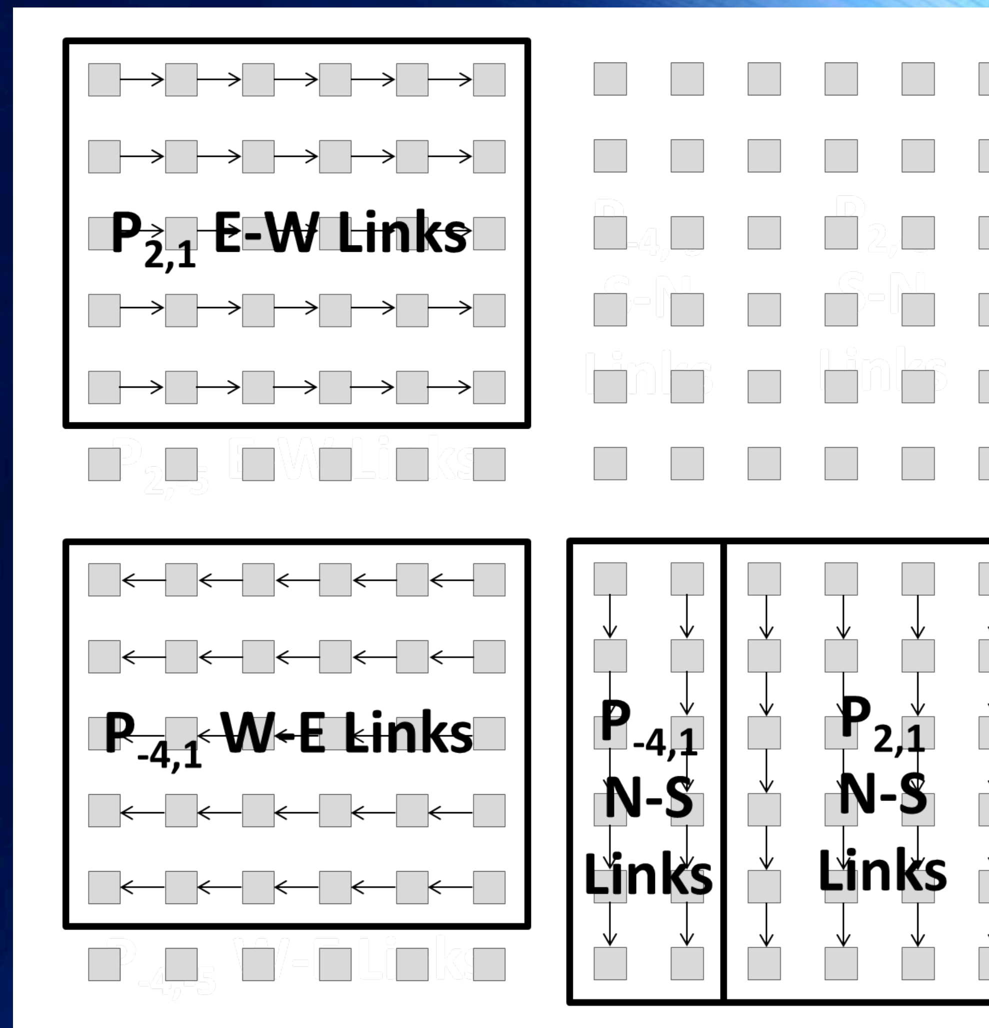




Connection Groups

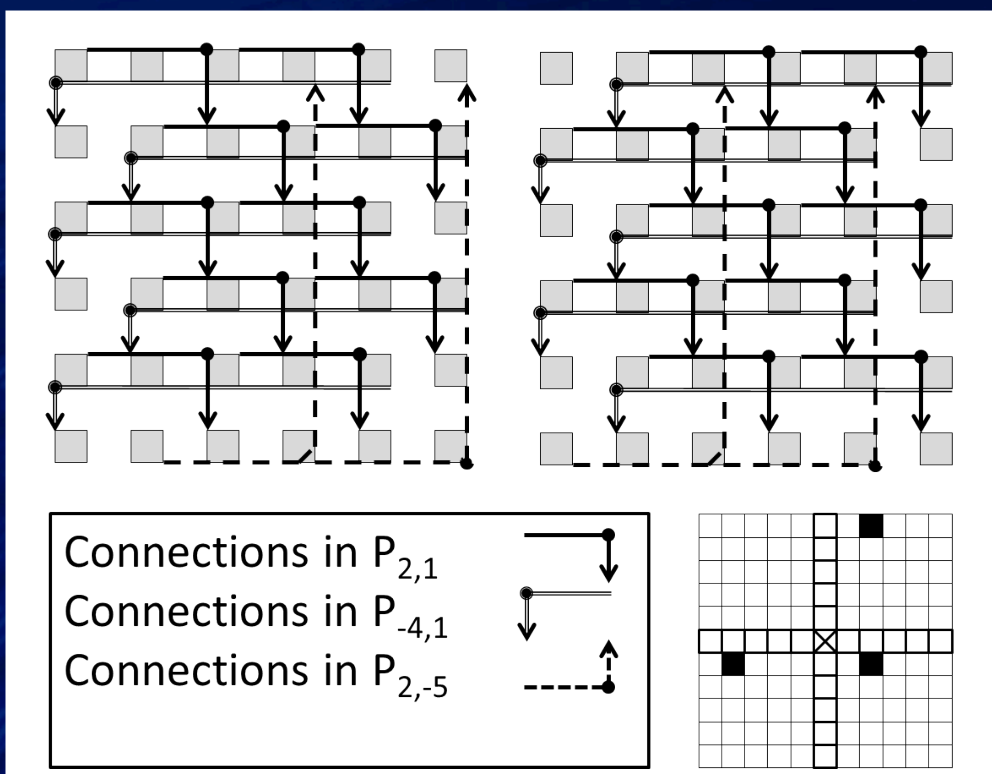
- No link contention
  - Example: $G_{2,1}$

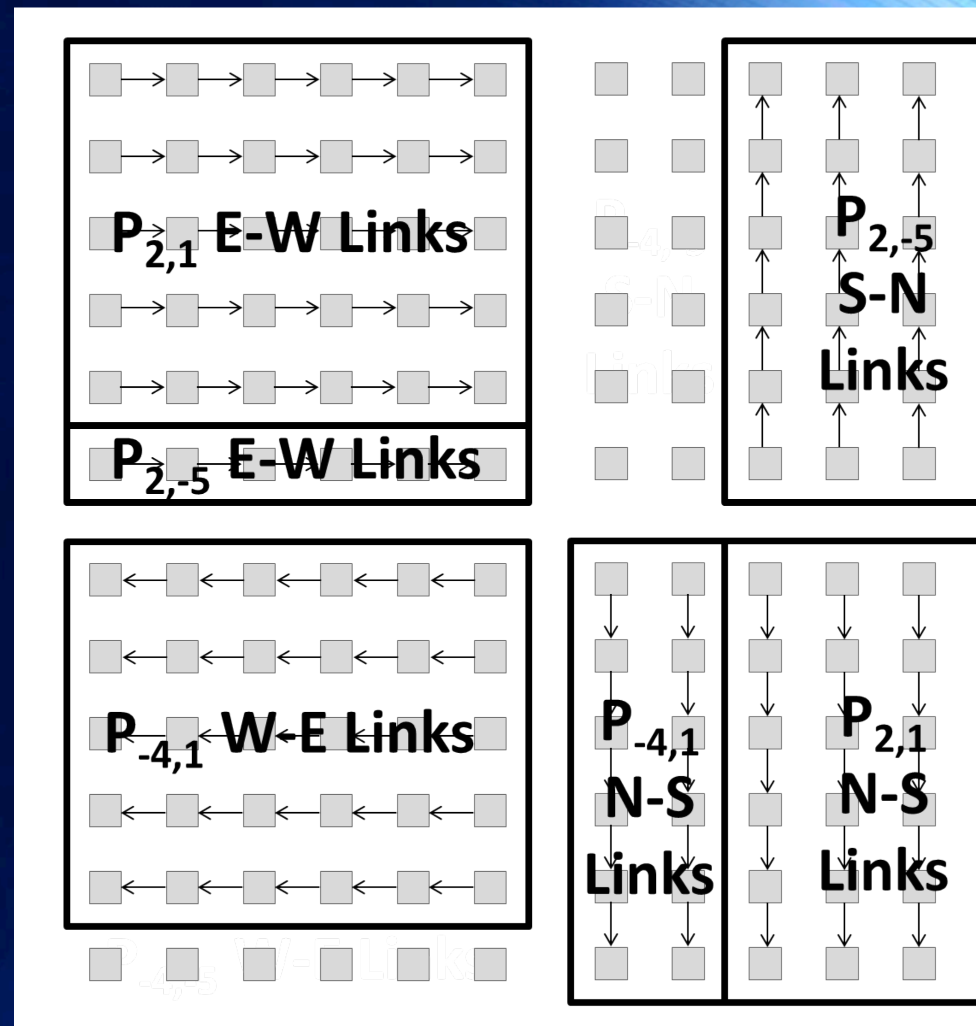




Connection Groups

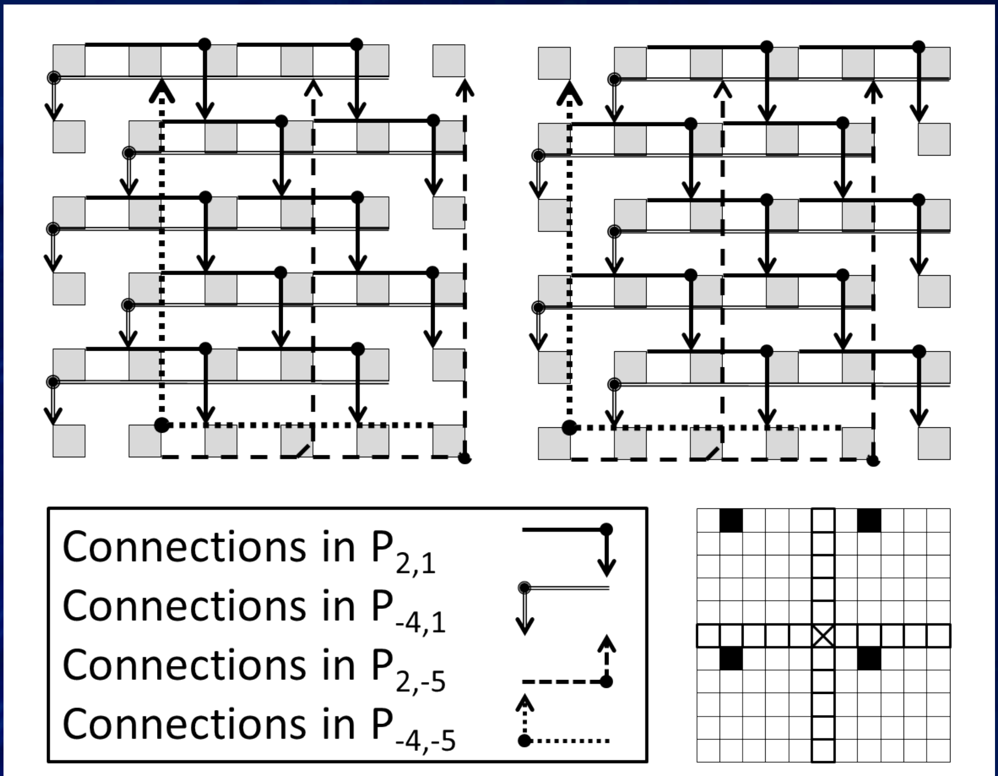
- No link contention
  - Example: $G_{2,1}$

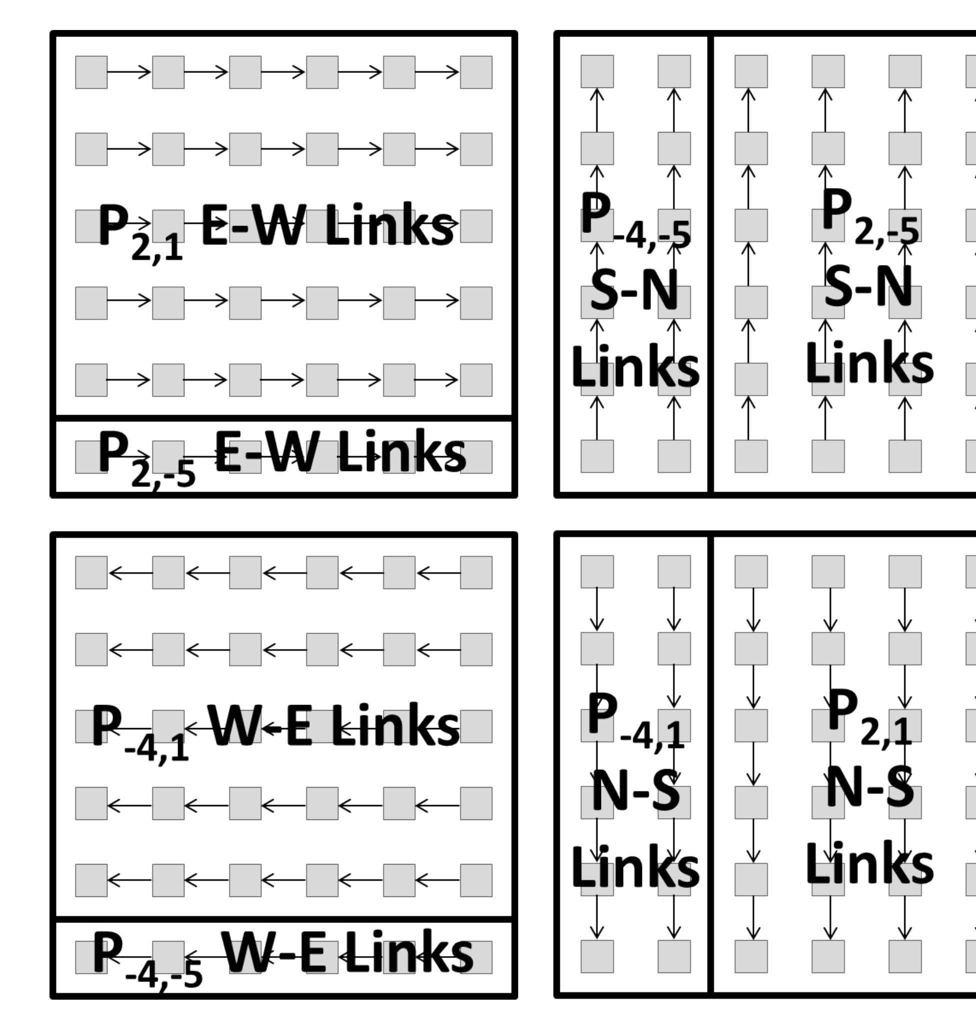




Connection Groups

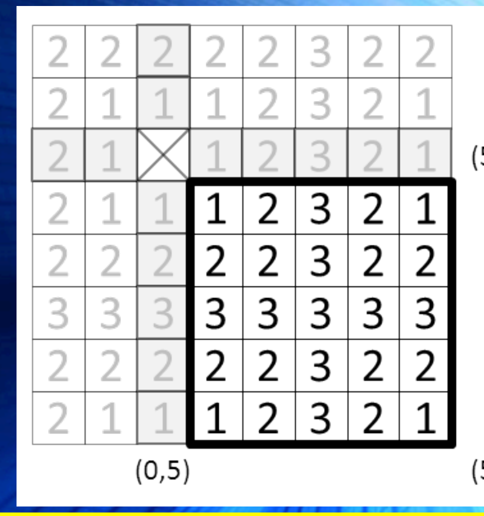
- No link contention
  - Example: $G_{2,1}$





Connection Group based Scheduling

- By separately multiplexing each of the groups, we can achieve a multiplexing degree of 55 for a 6x6 mesh
  - Note that this solution does not yet account for the 1D connections (where a=0, or b=0)
- If we are able to schedule the 1D connections with the others, then we can have a multiplexing degree of 55 for a 6x6 mesh
  - Theoretical min: 54
- Next step: combine 1D connections with the connection groups to realize this low multiplexing degree



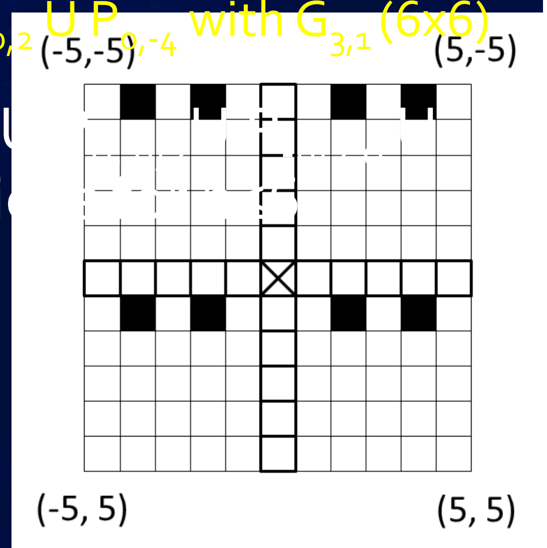
$$\sum_{a=1}^{n-1}\sum_{b=1}^{n-1}\max(\min(a,n-a),\min($$

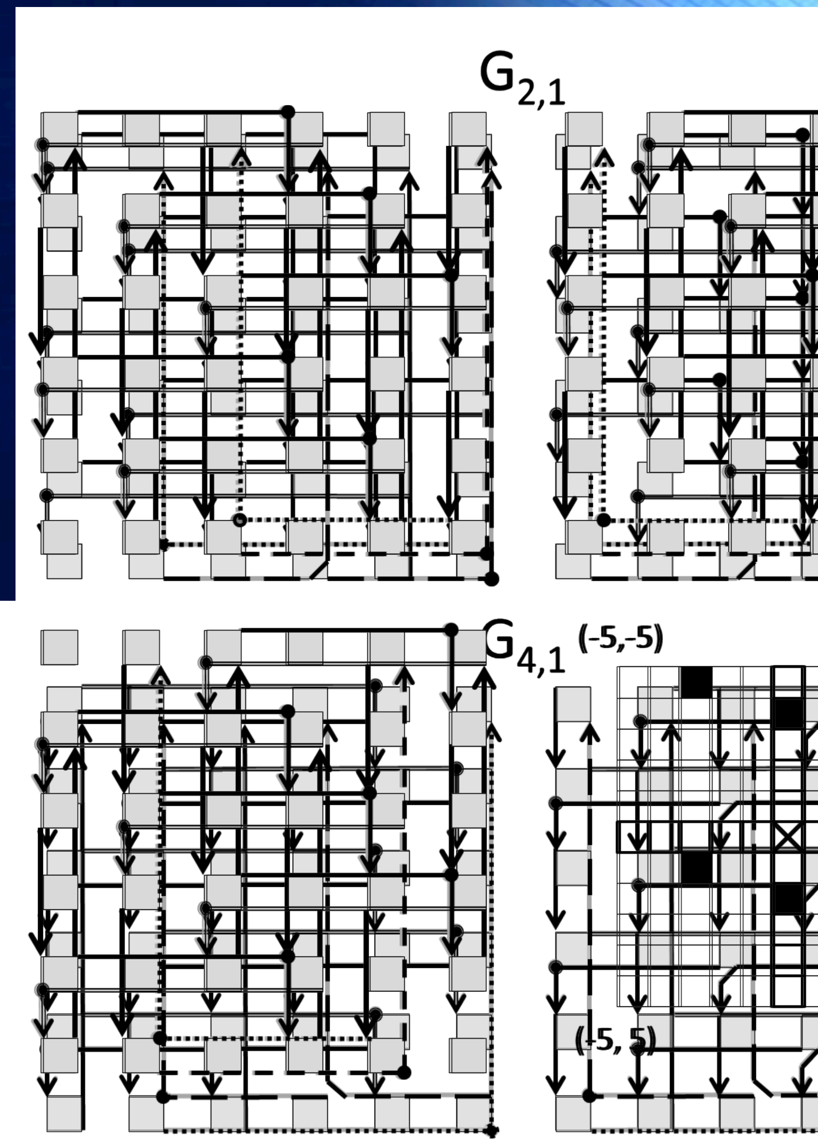
$$=\frac{n^3}{3}-\frac{n^2}{2}+\frac{n}{6}\text{, if n is even}$$

$$=\frac{n^3}{3}-\frac{n^2}{2}-\frac{n}{3}-\frac{1}{2}\text{, if n is odd}$$

Combining 1D Connections with Connection Groups

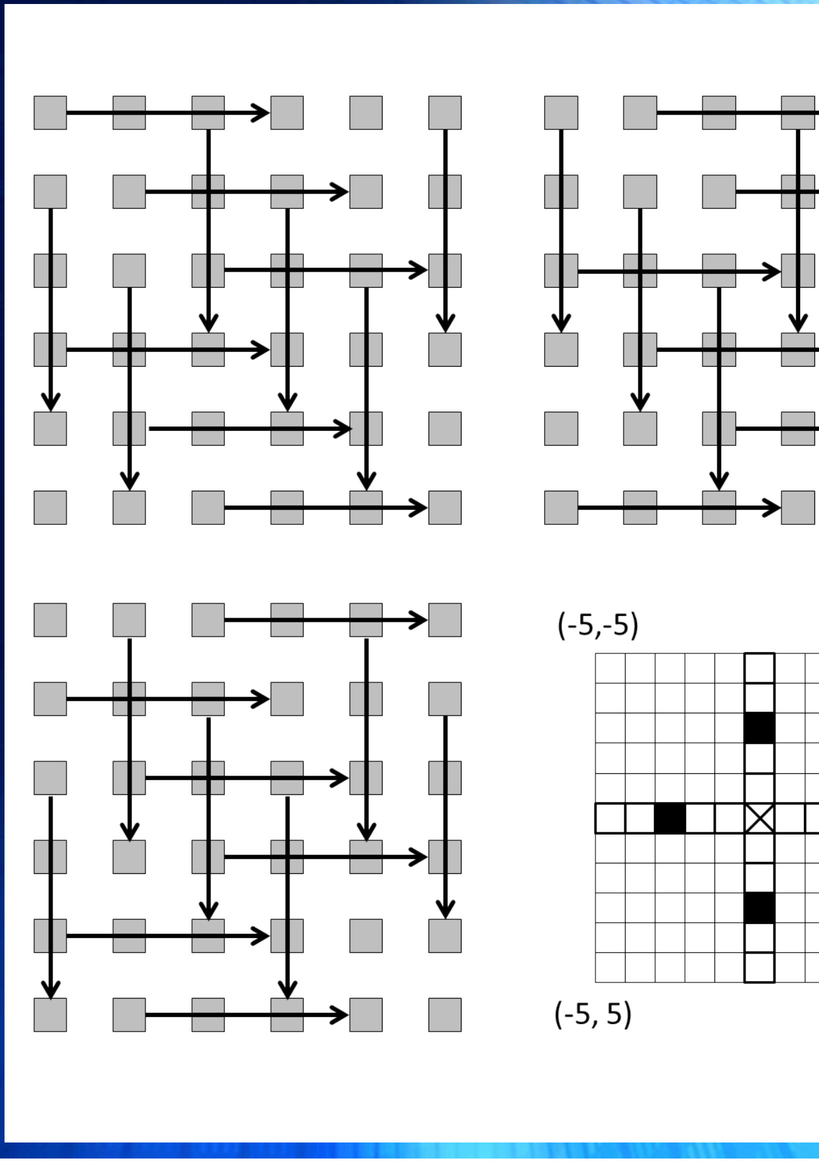
- In general, we combine $P_{0,a}$ U $P_{0,a-n}$ with $G_{a+1,1}$ U $G_{n-a+1,1}$
  - Example 1: combining $P_{0,1}$ U $P_{0,-5}$ with $G_{2,1}$ U $G_{4,1}$
- In the case of a+1 = n/2, we combine $P_{0,a}$ U $P_{0,a-n}$ with <u>just</u> $G_{a+1,1}$
  - Example 2: combining $P_{0,2}$ U $P_{0,-4}$ with $G_{3,1}$ (6x6)
- Cannot schedule $P_{n/2,0}$ U … $P_{0,-n/2}$ with 2D connecti…





The rest of the 1D connections

- We can schedule $P_{n/2,0} \cup P_{0, n/2} \cup P_{-n/2,0} \cup P_{0,-n/2}$ in n/2 time slots
  - For n≤6, we schedule these 1D connections in separate slots from the rest of our schedule
  - For n>6, we can schedule all 1D connections with the 2D connection groups, so we do not need to add any slots to our previous solution



Connection Group based Scheduling, Revised

n≤6

$$\left(\sum_{a=1}^{n-1}\sum_{b=1}^{n-1}\max(\min(a,n-a),\min(b,n-b))\right)+\frac{n}{2}^{*}$$

**Add only if n is even*

$$=\frac{n^3}{3}-\frac{n^2}{2}+\frac{2n}{3}$$, if n is even

$$=\frac{n^3}{3}-\frac{n^2}{2}-\frac{n}{3}-\frac{1}{2}$$, if n is odd

n>6

$$\sum_{a=1}^{n-1}\sum_{b=1}^{n-1}\max(\min(a,n-a),\min(b,n-b))$$

$$=\frac{n^3}{3}-\frac{n^2}{2}+\frac{n}{6}$$, if n is even

$$=\frac{n^3}{3}-\frac{n^2}{2}-\frac{n}{3}-\frac{1}{2}$$, if n is odd

2	2	2	2	2	3	2	2
2	1	1	1	2	3	2	1
2	1	×	1	2	3	2	1
2	1	1	1	2	3	2	1
2	2	2	2	2	3	2	2
3	3	3	3	3	3	3	3
2	2	2	2	2	3	2	2
2	1	1	1	2	3	2	1

(5,0)

(0,5)

(5,5)

Results: Multiplexing Degree Revisited

Size of Network	4x4	6x6	8x8	9x9	10x10
Theoretical Minimum	16	54	128	180	250
Genetic Algorithm	18	61	142	*	*
Deterministic Solution	16	58	140	199	280 **

- * No results given for larger networks by previous work[1]
- ** We can obtain results which scale by $O(n^3)$ for n >> 10, due to the systen nature of our scheduling algorithm
  - Theoretical minimum scales by $O(n^3)$

1. G. Hendry, J. Chan, S. Kamil, L. Oliker, J. Shalf, L. Carloni, and K. Bergman, "Silicon Nanophotonic Network-on-Chip using TDM Arbitration," Proceedings of IEEE Symposium on High-Performance Interconnects, 2010.

Future Work

- Combine Multiplexing Strategies
  - We can leverage the regularity of our schedule and our x-y routing to develop a method of efficiently utilizing TDM in combination with WDM and SDM
- Why will this work?
  - We can leverage regularity to limit the overhead to utilize WDM efficiently
    - Each wavelength needs several microrings within a router tuned to it
    - More wavelengths = more microrings = more complexity = more cost
  - We can divide the all-to-all scheduling into groups with equal multiplexing degrees, then split them between planes/waveguides

Thank You

Any Questions?

Expected Graduation Date (M.S.): December 2013

Looking for jobs!