CS525-Parallel Computing HW2

1. Estimating the diameters for Ring, Mesh and Tree Networks for different values of p with k=4,16.

Following are the plots for all Networks which estimate the diameter of the networks based on no of processors involved in the network.

Here, we observe that for Ring Network, the value of diameter increases as no of processors/8.

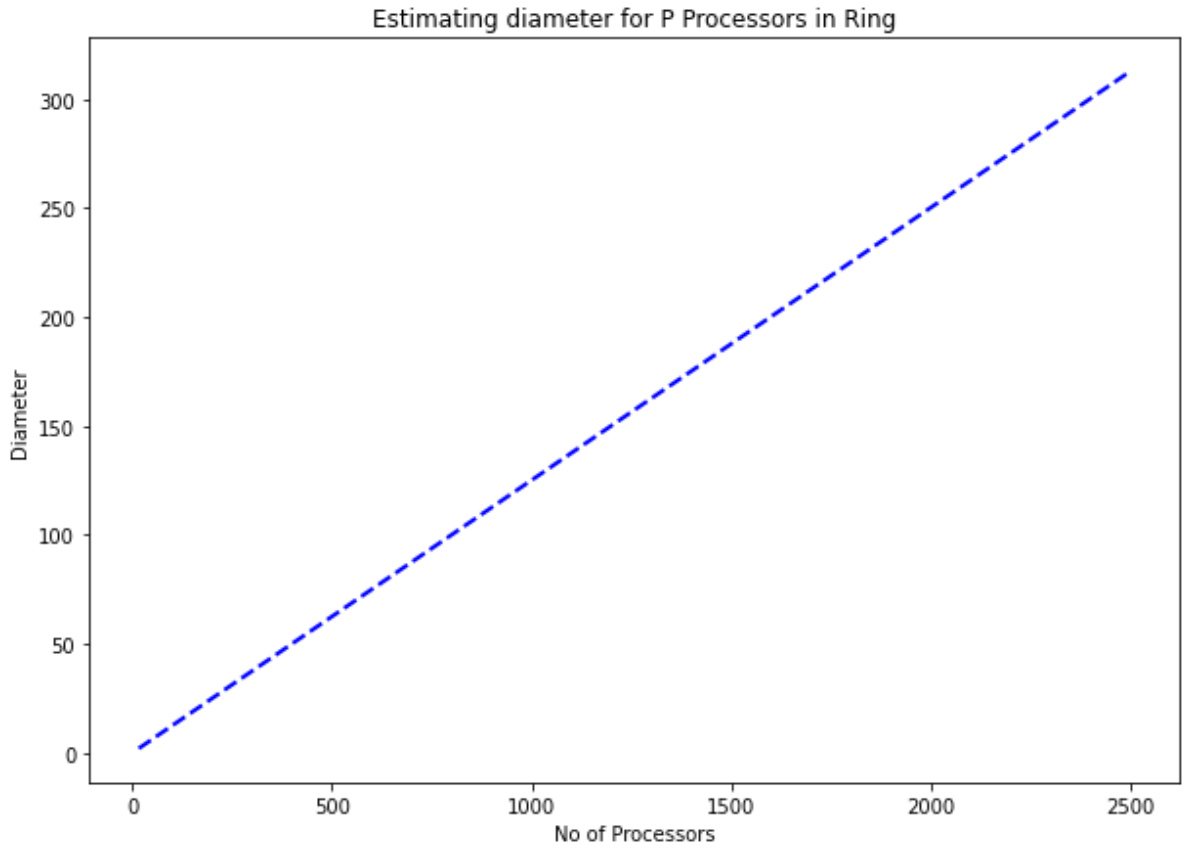
The line is fitted using python's polyfit function and plotted using matplotlib library.

Ploy fit function will take no of processors(x axis values) and diameter value(y axis values) as input and generate the x and y intercept. Therefore, based on x values and intercepts, we plot the below graphs.

Ex. a, b = np.polyfit(x, y, 1)

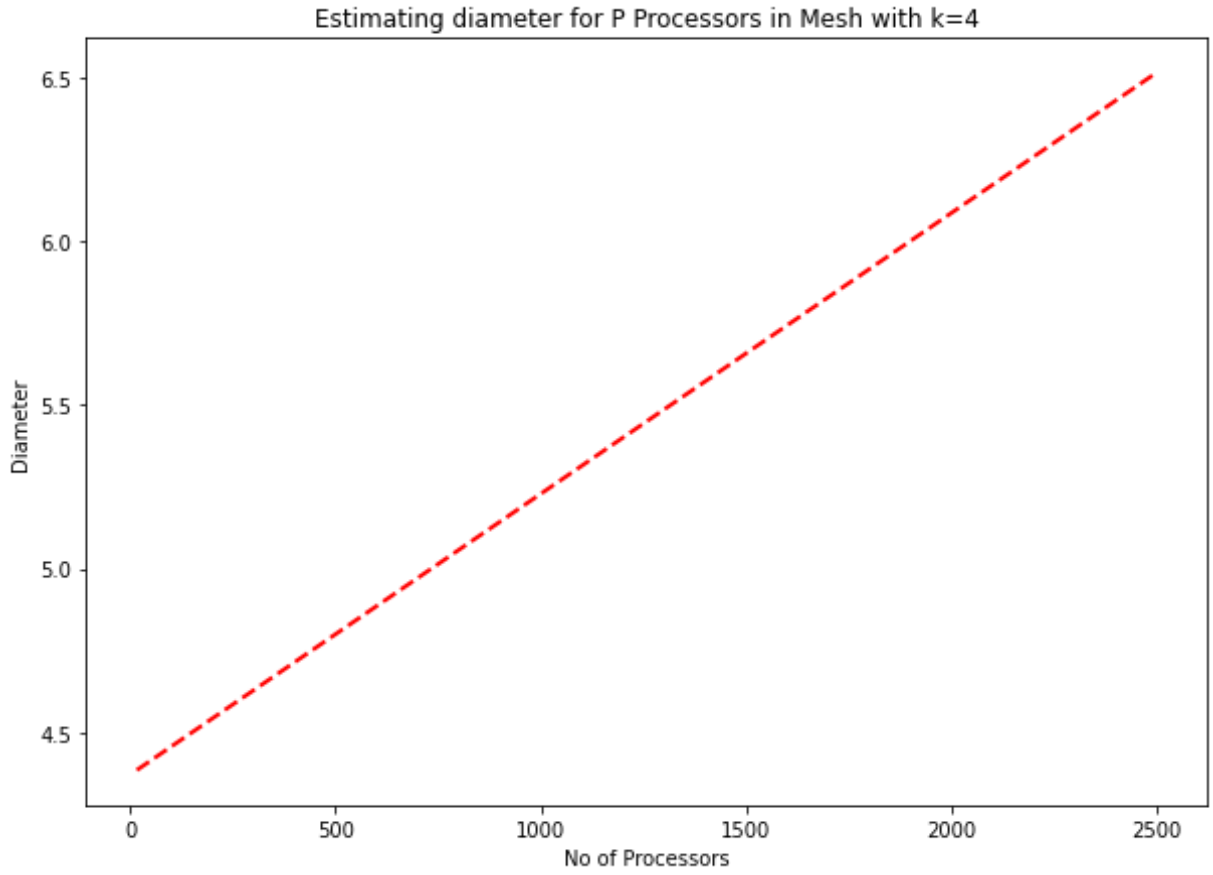
plt.plot(x, a*x+b, color='blue', linestyle='--', linewidth=2)

a. Ring Network

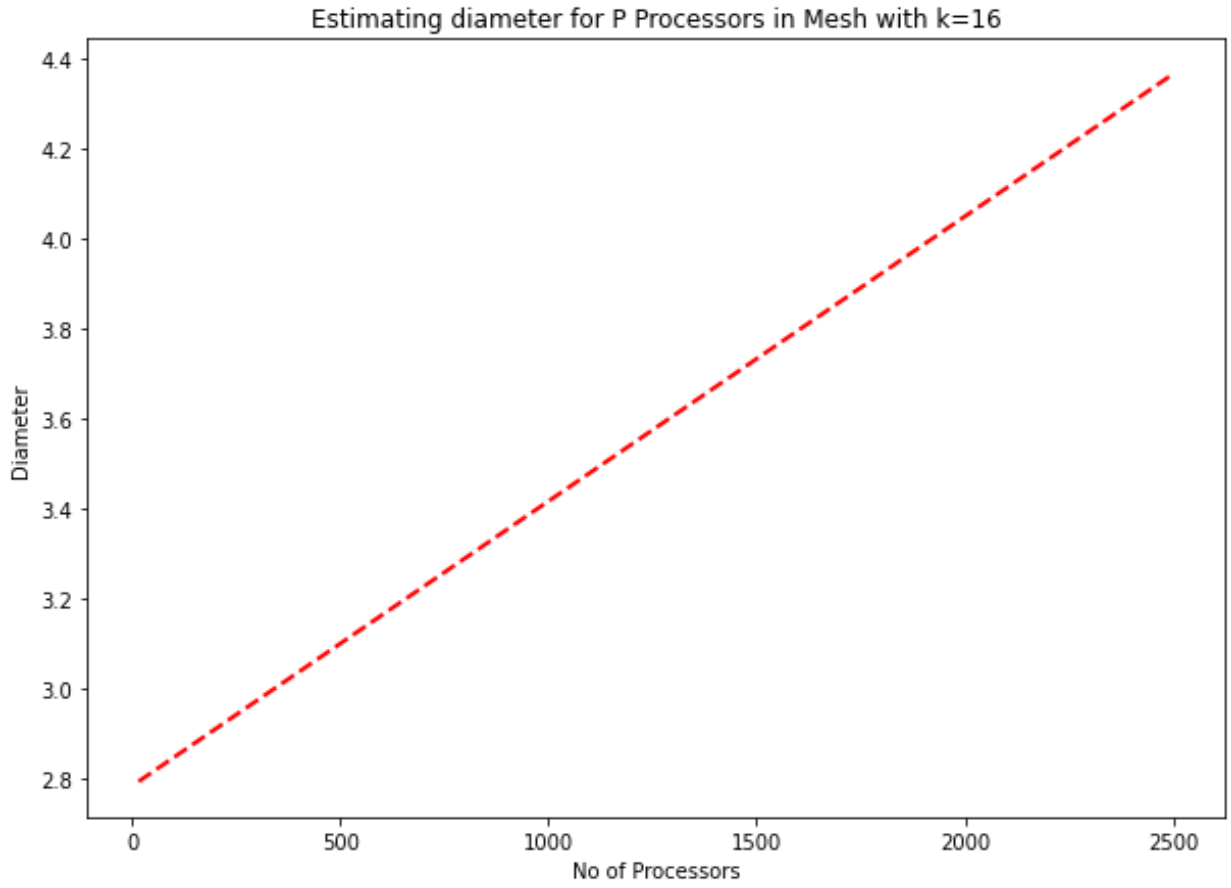


b. Mesh Network

K = 4

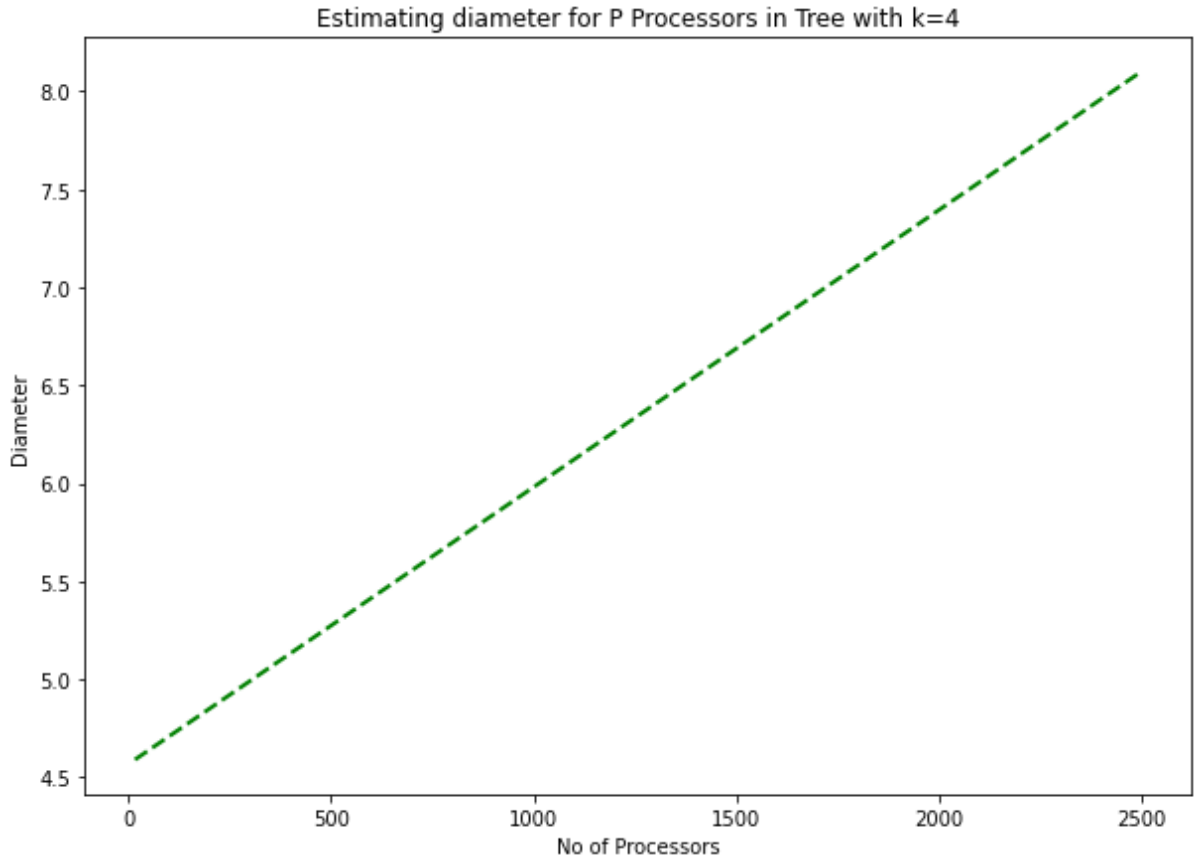


K =16

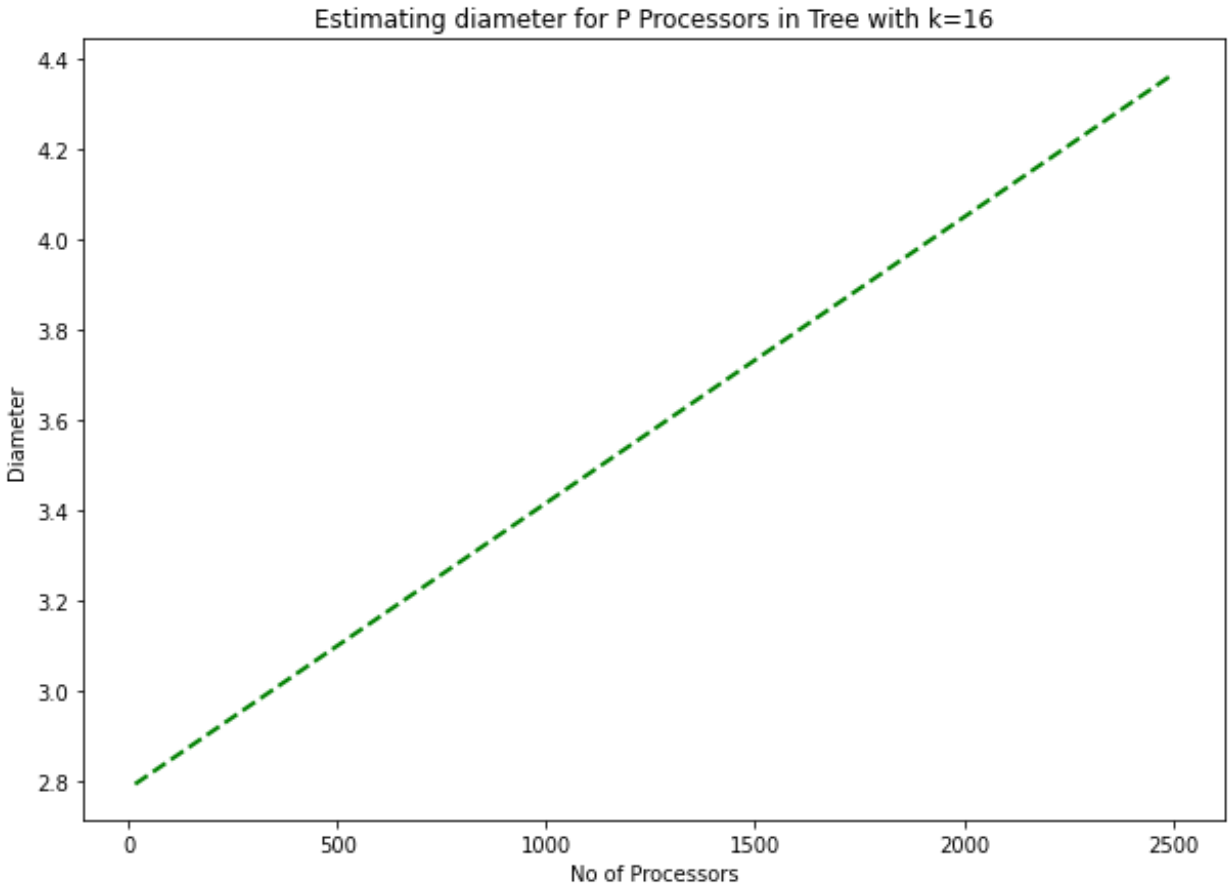


c. Tree Network

K = 4


Estimating diameter for P Processors in Tree with k=4
Diameter
No of Processors
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
0
500
1000
1500
2000
2500

K=16



Output of the code:

Ring Network:

Processors:16 Diameter:2

Processors:32 Diameter:4

Processors:64 Diameter:8

Processors:100 Diameter:13

Processors:128 Diameter:16

Processors:225 Diameter:28

Processors:256 Diameter:32

Processors:512 Diameter:64

Processors:1024 Diameter:128

Processors:2048 Diameter:256

Processors:2500 Diameter:313

Mesh Network k:4

Processors:16 Diameter:3

Processors:32 Diameter:4

Processors:64 Diameter:4

Processors:100 Diameter:4

Processors:128 Diameter:5

Processors:225 Diameter:5

Processors:256 Diameter:5

Processors:512 Diameter:6

Processors:1024 Diameter:6

Processors:2048 Diameter:6

Processors:2500 Diameter:6

Mesh Network k:16

Processors:16 Diameter:2

Processors:32 Diameter:2

Processors:64 Diameter:3

Processors:100 Diameter:3

Processors:128 Diameter:3

Processors:225 Diameter:3

Processors:256 Diameter:3

Processors:512 Diameter:4

Processors:1024 Diameter:4

Processors:2048 Diameter:4

Processors:2500 Diameter:4

Tree Network k:4

Processors:16 Diameter:3

Processors:32 Diameter:4

Processors:64 Diameter:4

Processors:100 Diameter:5

Processors:128 Diameter:5

Processors:225 Diameter:5

Processors:256 Diameter:6

Processors:512 Diameter:6

Processors:1024 Diameter:7

Processors:2048 Diameter:8

Processors:2500 Diameter:7

Tree Network k:16

Processors:16 Diameter:2

Processors:32 Diameter:2

Processors:64 Diameter:3

Processors:100 Diameter:3

Processors:128 Diameter:3

Processors:225 Diameter:3

Processors:256 Diameter:3

Processors:512 Diameter:4

Processors:1024 Diameter:4

Processors:2048 Diameter:4

Processors:2500 Diameter:4

Execute the code:

Code will be found inside hw2/question1/ folder

Compile: g++ diameter.cpp -o ./diameter

Execute: ./diameter

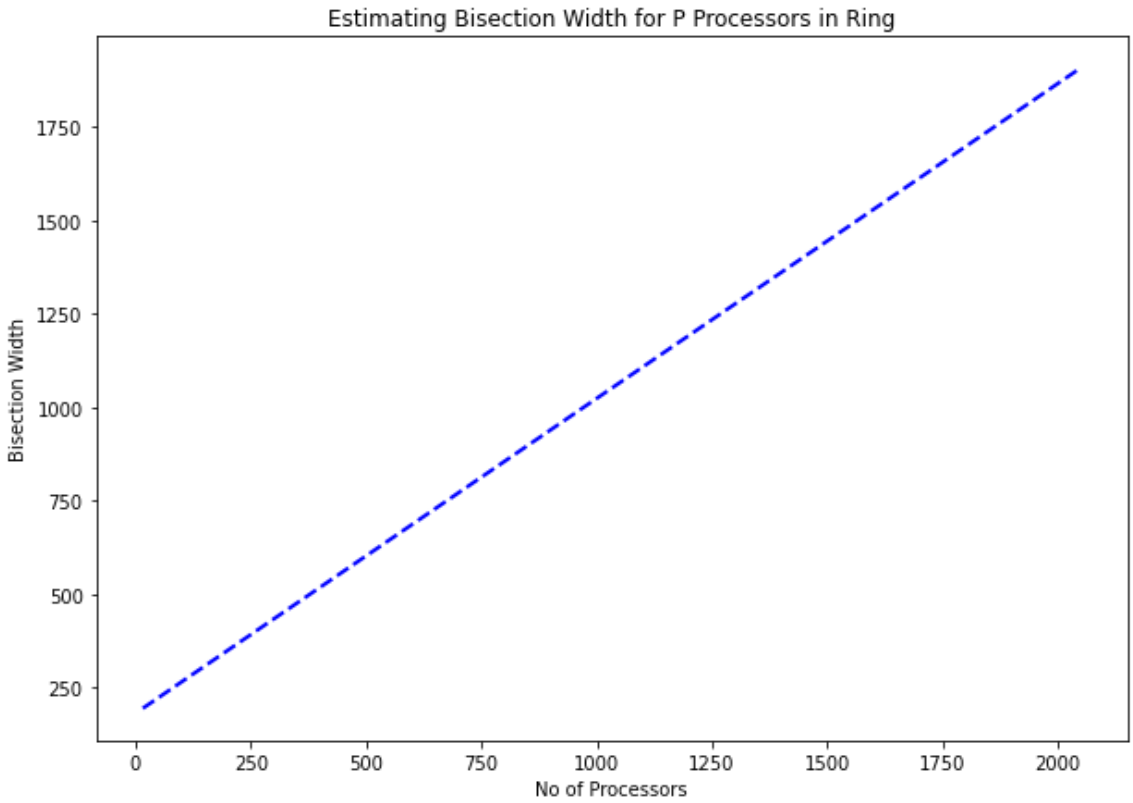
Output: output will be saved in output.txt file

2. Estimating the Bisection width for Ring, Mesh and Tree Networks for different values of p with k=4,16.

Following are the plots for all Networks which estimates the bisection width of the networks based on no of processors involved in the network.

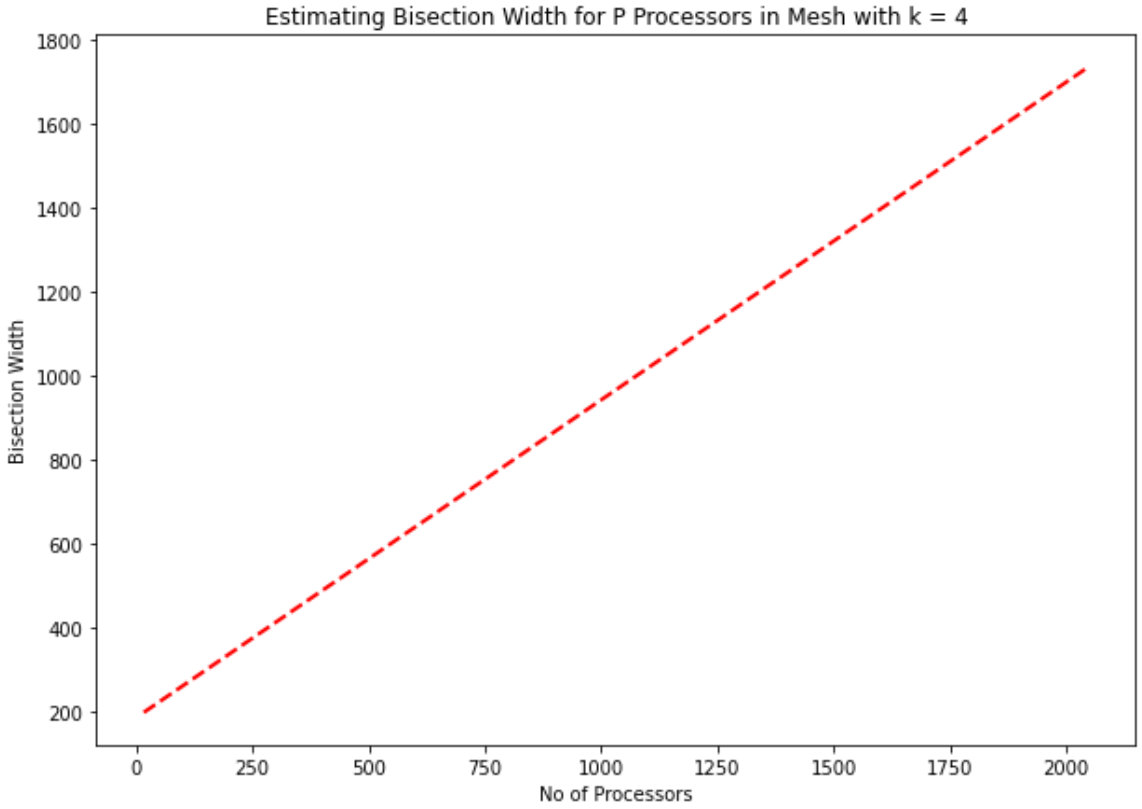
The line is fitted using python's polyfit function and plotted using matplotlib library.

a. Ring Network

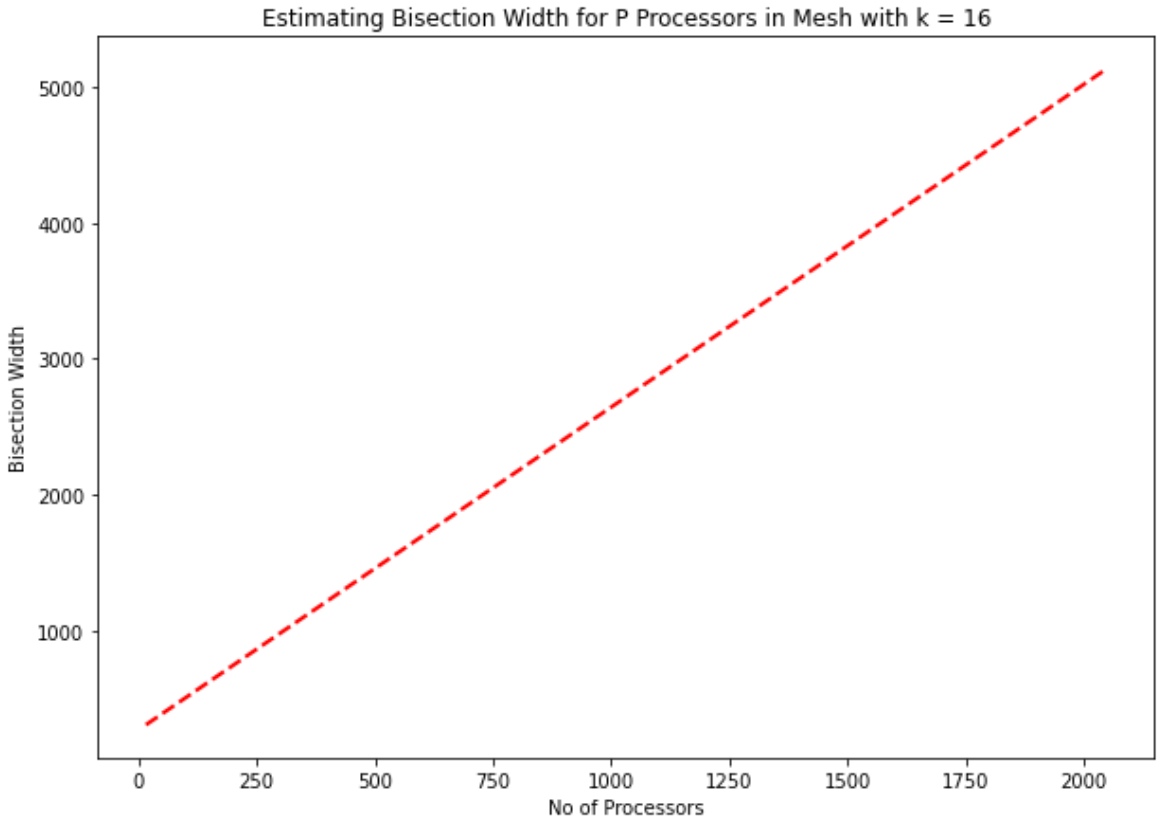


b. Mesh Network

K = 4

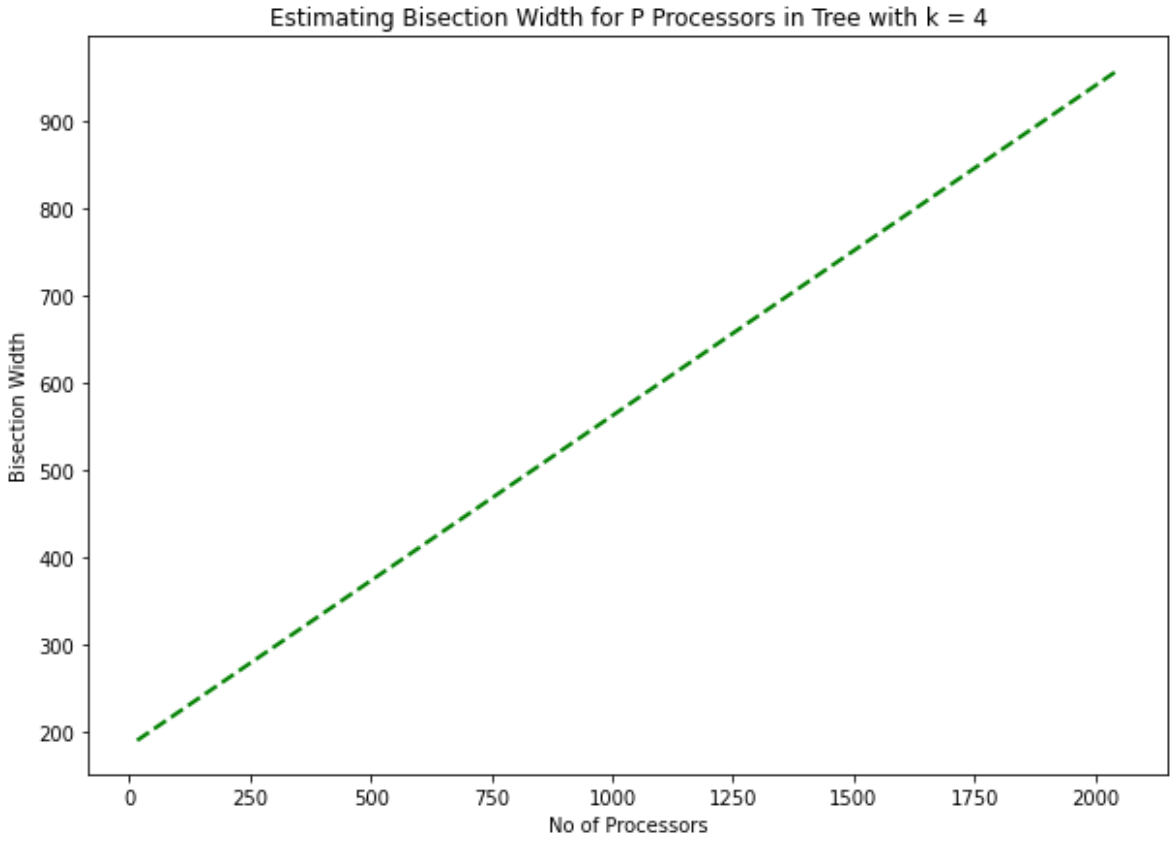

Estimating Bisection Width for P Processors in Mesh with k = 4
Bisection Width
No of Processors
1800
1600
1400
1200
1000
800
600
400
200
0
250
500
750
1000
1250
1500
1750
2000

K = 16

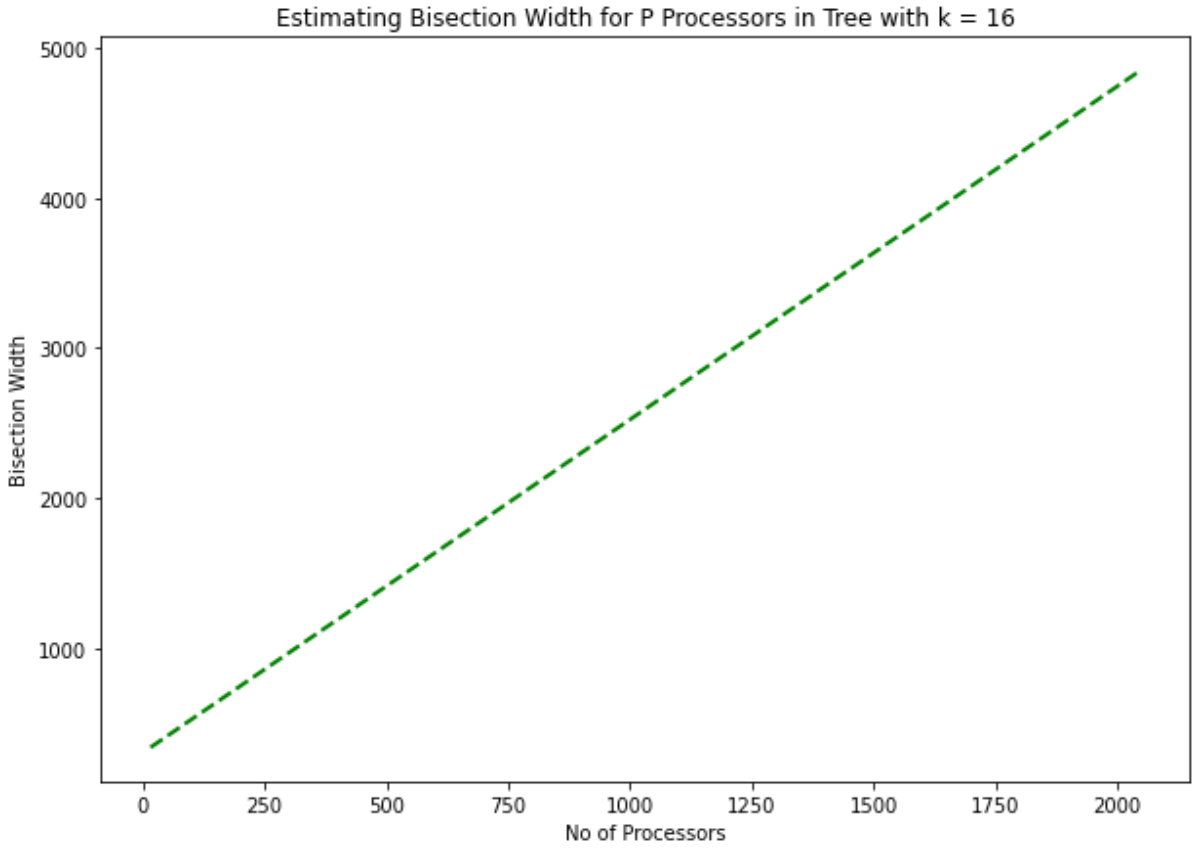


c. Tree Network

K = 4


Estimating Bisection Width for P Processors in Tree with k = 4
Bisection Width
No of Processors
200
300
400
500
600
700
800
900
0
250
500
750
1000
1250
1500
1750
2000

K = 16



Output of the code:

Ring Network:

Processors:16 Bisection Width:35

Processors:32 Bisection Width:78

Processors:64 Bisection Width:182

Processors:128 Bisection Width:208

Processors:256 Bisection Width:408

Processors:512 Bisection Width:788

Processors:1024 Bisection Width:1580

Processors:2048 Bisection Width:1602

Mesh Network k:4

Processors:16 Bisection Width:25

Processors:32 Bisection Width:48

Processors:64 Bisection Width:141

Processors:128 Bisection Width:243

Processors:256 Bisection Width:616

Processors:512 Bisection Width:737

Processors:1024 Bisection Width:1203

Processors:2048 Bisection Width:1552

Mesh Network k:16

Processors:16 Bisection Width:60

Processors:32 Bisection Width:116

Processors:64 Bisection Width:251

Processors:128 Bisection Width:617

Processors:256 Bisection Width:873

Processors:512 Bisection Width:1959

Processors:1024 Bisection Width:3177

Processors:2048 Bisection Width:4781

Tree Network k:4

Processors:16 Bisection Width:34

Processors:32 Bisection Width:52

Processors:64 Bisection Width:87

Processors:128 Bisection Width:205

Processors:256 Bisection Width:306

Processors:512 Bisection Width:466

Processors:1024 Bisection Width:1274

Processors:2048 Bisection Width:590

Tree Network k:16

Processors:16 Bisection Width:55

Processors:32 Bisection Width:133

Processors:64 Bisection Width:303

Processors:128 Bisection Width:653

Processors:256 Bisection Width:1004

Processors:512 Bisection Width:1882

Processors:1024 Bisection Width:2916

Processors:2048 Bisection Width:4560

Execute the code:

Code will be found inside hw2/question2/ folder

Compile: g++ bisection_width.cpp -o ./bisection_width

Execute: ./bisection_width

Output: output will be saved in output.txt file

3. Mapping Ring Network to Mesh and estimating its Dilation and Congestion.

Dilation is the maximum stretch of a particular ring edge in the mesh.

Congestion is maximum no of edges in one network that are mapped to an edge in another network.

Output of the code:

Mapping of Ring to Mesh with k=4:

No of Processors:16

Dilation is :3

Congestion is: 6

No of Processors:32

Dilation is :4

Congestion is: 9

No of Processors:64

Dilation is :4

Congestion is: 8

No of Processors:128

Dilation is :5

Congestion is: 9

No of Processors:256

Dilation is :4

Congestion is: 11

No of Processors:512

Dilation is :5

Congestion is: 14

No of Processors:1024

Dilation is :5

Congestion is: 13

Mapping of Ring to Mesh with k=16:

No of Processors:16

Dilation is :3

Congestion is: 6

No of Processors:32

Dilation is :4

Congestion is: 9

No of Processors:64

Dilation is :4

Congestion is: 9

No of Processors:128

Dilation is :5

Congestion is: 10

No of Processors:256

Dilation is :5

Congestion is: 10

No of Processors:512

Dilation is :5

Congestion is: 12

No of Processors:1024

Dilation is :5

Congestion is: 12

Execute the code:

Code will be found inside hw2/question3/ folder

Compile: g++ -std=c++11 mapping.cpp -o ./mapping

Execute: ./mapping

Output: output will be saved in output.txt file

4. Given that each link in ring network operates at 500 Mb/s and each link in mesh network operates at 200 Mb/s. For p=64 and k=4,

If the networks are congested, the performance rate for a network will be:

bisection width of network * bandwidth of each link

For p=64,

Performance rate for ring network = 46 * 500 Mb/s = 23000Mb/sec

Similarly, the performance rate for mesh network with k=4 = Bisection width * Bandwidth of each link

= 26 * 200 = 5200 Mb/sec

Here, we see the **performance of ring network is much better than mesh**, hence we would prefer ring over mesh in congested network.

For p=1024,

Performance rate for ring network = 806 * 500 Mb/s = 403000Mb/sec

Similarly, the performance rate for mesh network with k=4 = Bisection width * Bandwidth of each link

= 389 * 200 = 77800Mb/sec

Here, we see the **performance of ring network is much better than mesh**, hence we would prefer ring over mesh in congested network.

If the networks are non-congested, the performance rate for a network will be :

no of links in network * bandwidth of each link

For p=64,

Performance rate for ring network = 160 * 500 Mb/s = 80000Mb/sec

Similarly, the performance rate for mesh network with k=4 = Bisection width * Bandwidth of each link

= 240 * 200 = 48000Mb/sec

Here, we see the **performance of ring network is much better than mesh**, hence we would prefer ring over mesh in non-congested network.

For p=1024,

Performance rate for ring network = 2560 * 500 Mb/s = 1.28e+06Mb/sec

Similarly, the performance rate for mesh network with k=4 = Bisection width * Bandwidth of each link

= 4032 * 200 = 806400Mb/sec

Here, we see the **performance of ring network is much better than mesh**, hence we would prefer ring over mesh in non congested network.

Output of the code:

Ring Network:

No of Processors:64

Bisection Width:46

No of links:160

Each link operates at 500Mb/s. Bisection Bandwidth for congestion is: 23000Mb/sec

Each link operates at 500Mb/s. Bisection Bandwidth for non congestion is: 80000Mb/sec

No of Processors:1024

Bisection Width:806

No of links:2560

Each link operates at 500Mb/s. Bisection Bandwidth for congestion is: 403000Mb/sec

Each link operates at 500Mb/s. Bisection Bandwidth for non congestion is: 1.28e+06Mb/sec

Mesh Network k:4

No of Processors:64

Bisection Width:26

No of links:240

Each link operates at 200Mb/s. Bisection Bandwidth for congestion is: 5200Mb/sec

Each link operates at 200Mb/s. Bisection Bandwidth for non congestion is: 48000Mb/sec

No of Processors:1024

Bisection Width:389

No of links:4032

Each link operates at 200Mb/s. Bisection Bandwidth for congestion is: 77800Mb/sec

Each link operates at 200Mb/s. Bisection Bandwidth for non congestion is: 806400Mb/sec

Execute the code:

Code will be found inside hw2/question4/ folder

Compile: g++ main.cpp -o ./main

Execute: ./main

Output: output will be saved in output.txt file