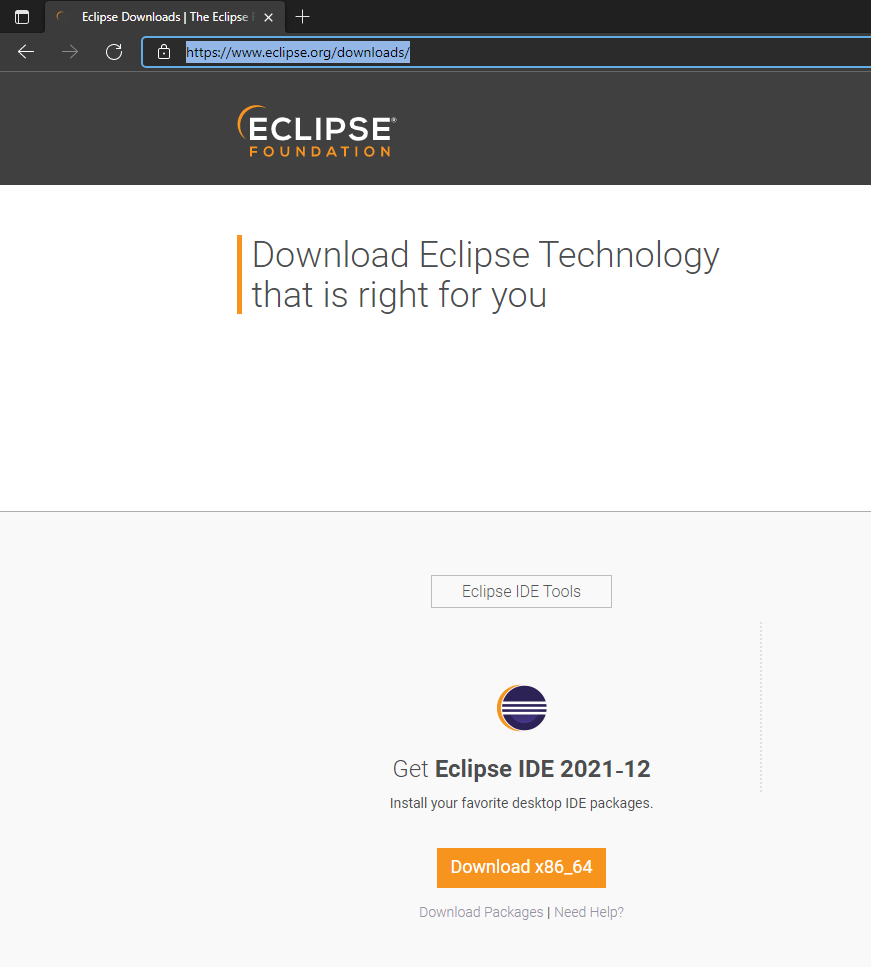
**Reality Conforming Approach vs State of Art method for computing worst-case delays in AFDX.**

**Paper :** [**A Reality-Conforming Approach for QoS Performance Analysis of AFDX in Cyber-Physical Avionics Systems**](https://ieeexplore.ieee.org/document/9521335)

**1. Download the Eclipse IDE from the Internet.**

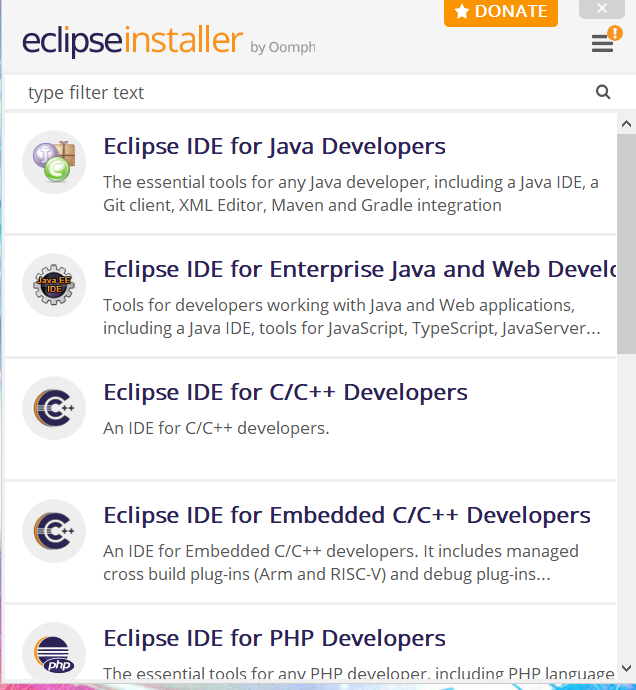
* *Download latest Eclipse IDE from the given link.*

Link : [Eclipse Downloads | The Eclipse Foundation](https://www.eclipse.org/downloads/)

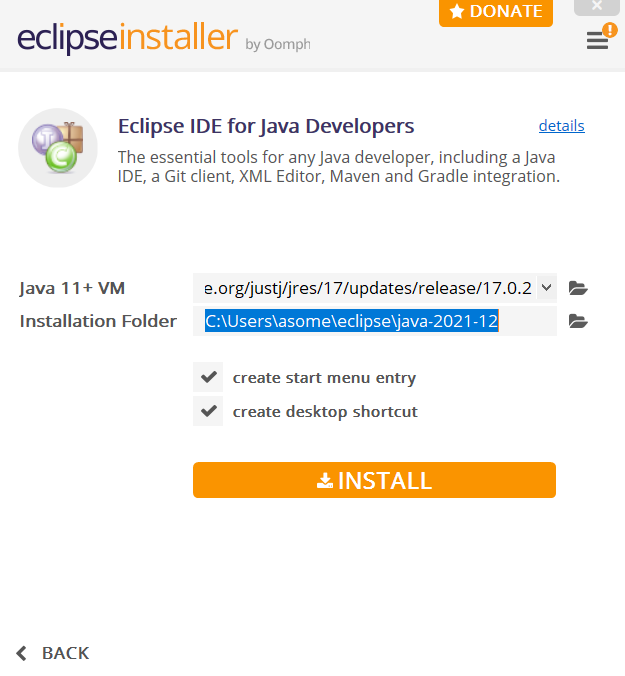


**2. Install the Eclipse into the desired directory.**

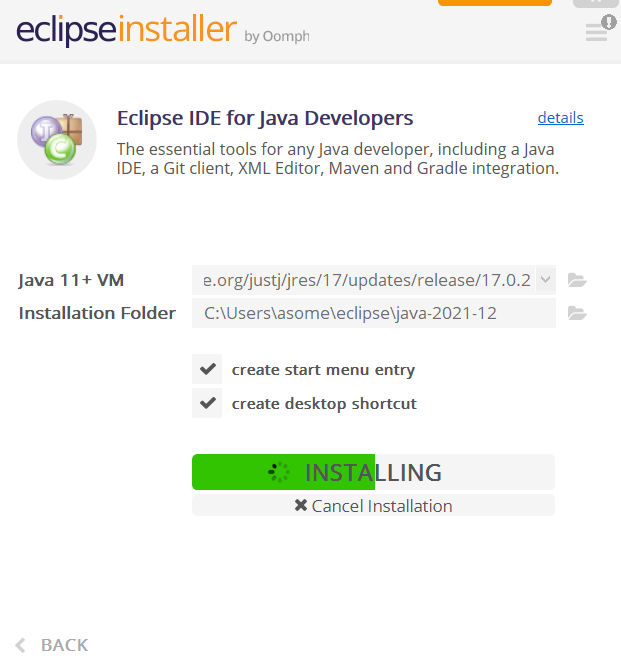
* *Launch the Eclipse Installer.*
* *Select* ***Eclipse IDE for Java Developers****.*



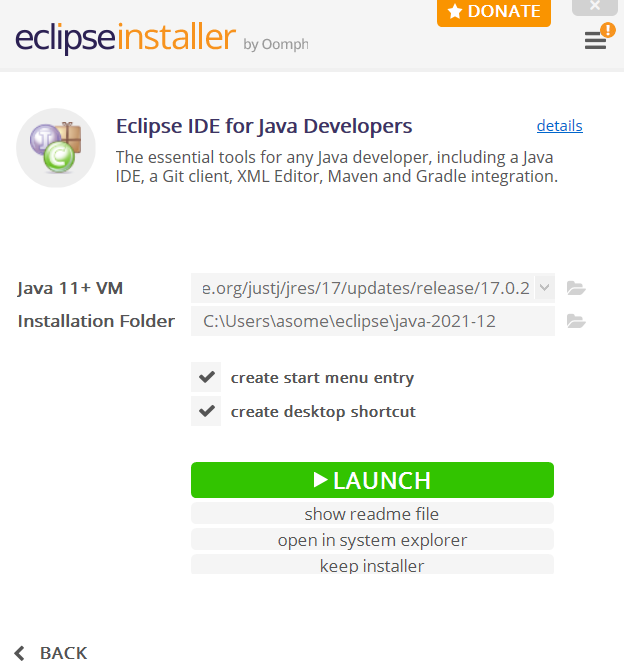
* *Select the directory where you want to install Eclipse.*



* *Click the* ***INSTALL*** *option*

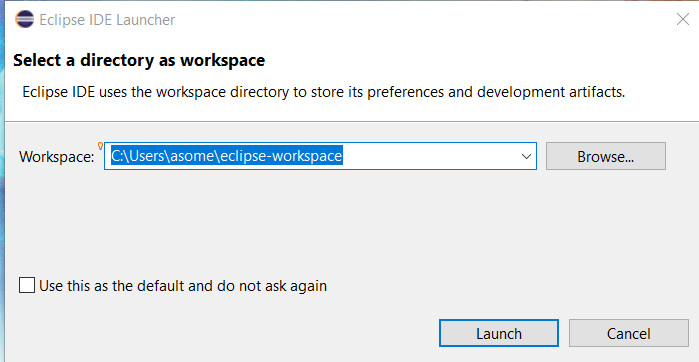


* *After Installation is complete, Click on* ***LAUNCH*** *option.*

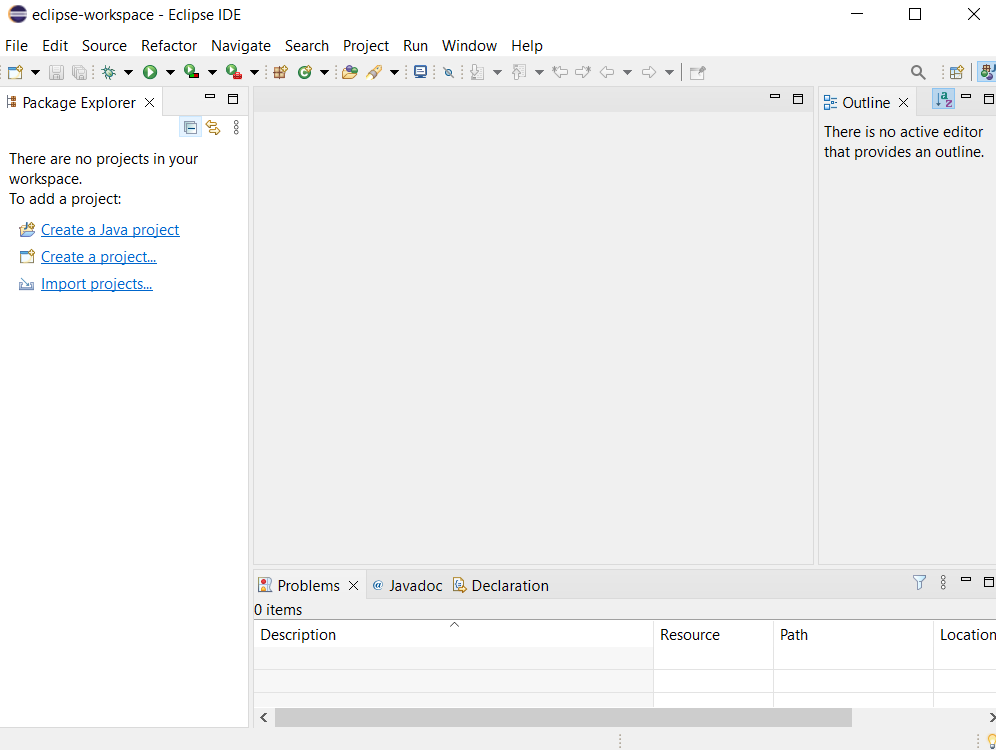


**3. After Installation Launch the eclipse:**

* *Select a directory as a workspace.*



* *Click on* ***Launch*** *Button.*
* *It will launch Eclipse Workspace.*

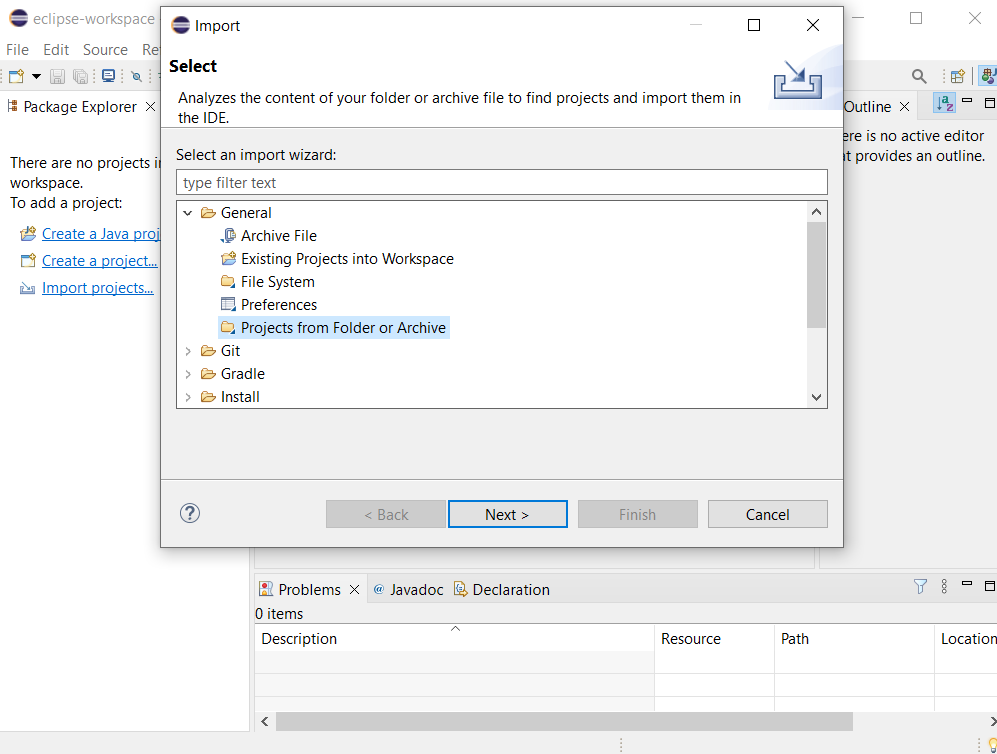


**4. Download the AFDX Code**

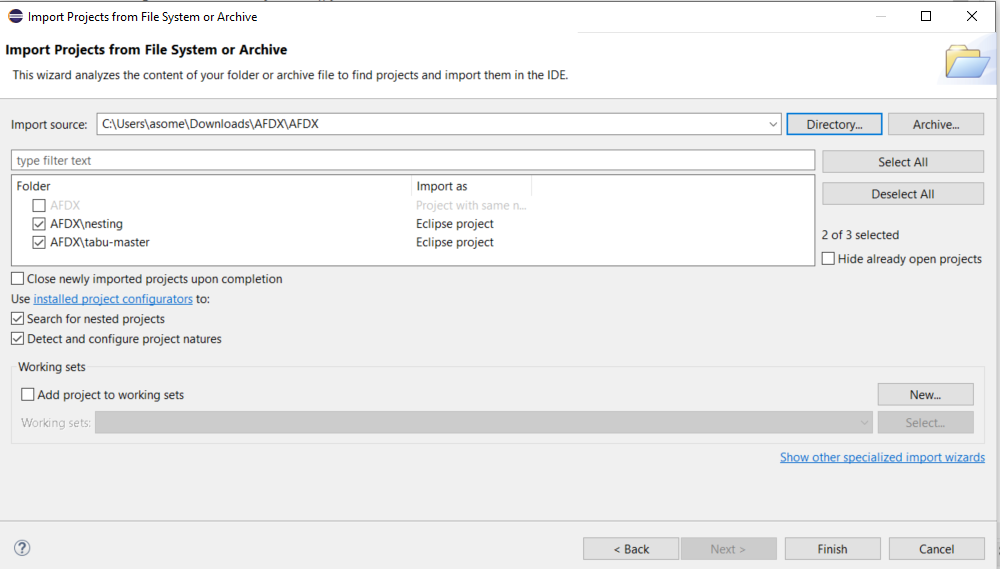
* *Download AFDX from the portal link given below.*
* *Link:*
* *Move the AFDX project to the working directory of your project.*

**5. Import the AFDX folder into the Eclipse**

* *Inside Eclipse Workspace. Go to* ***File****, then select* ***Import****.*
* *Then go to* ***General*** *folder under* ***Select an import Wizard*** *and select* ***Projects from Folder or Archive*** *option.*

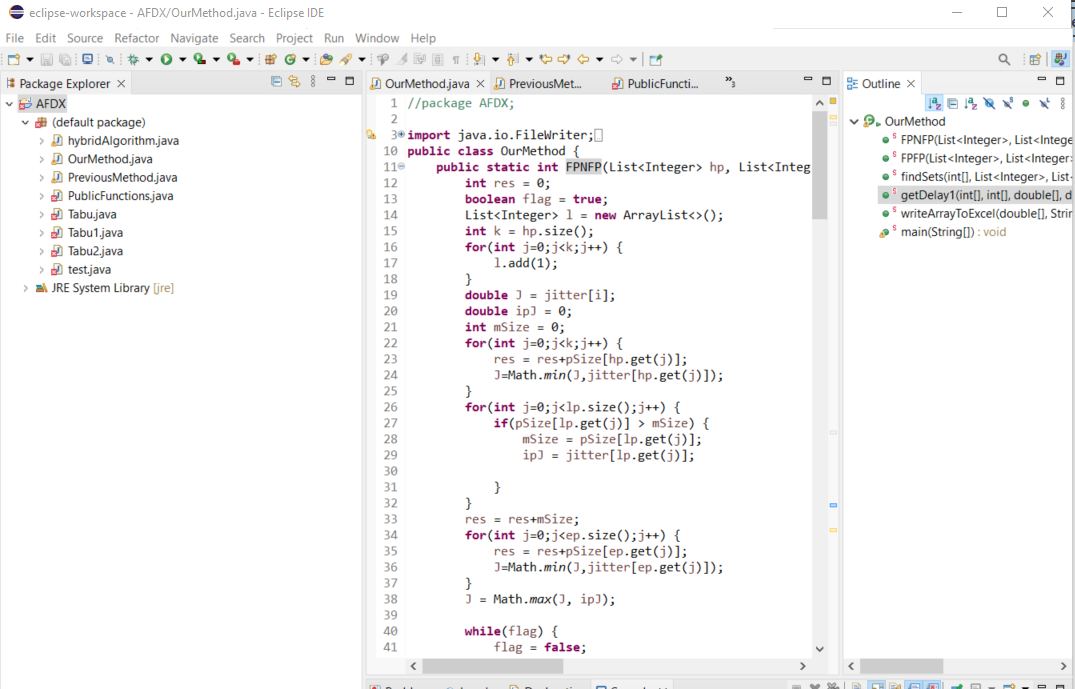


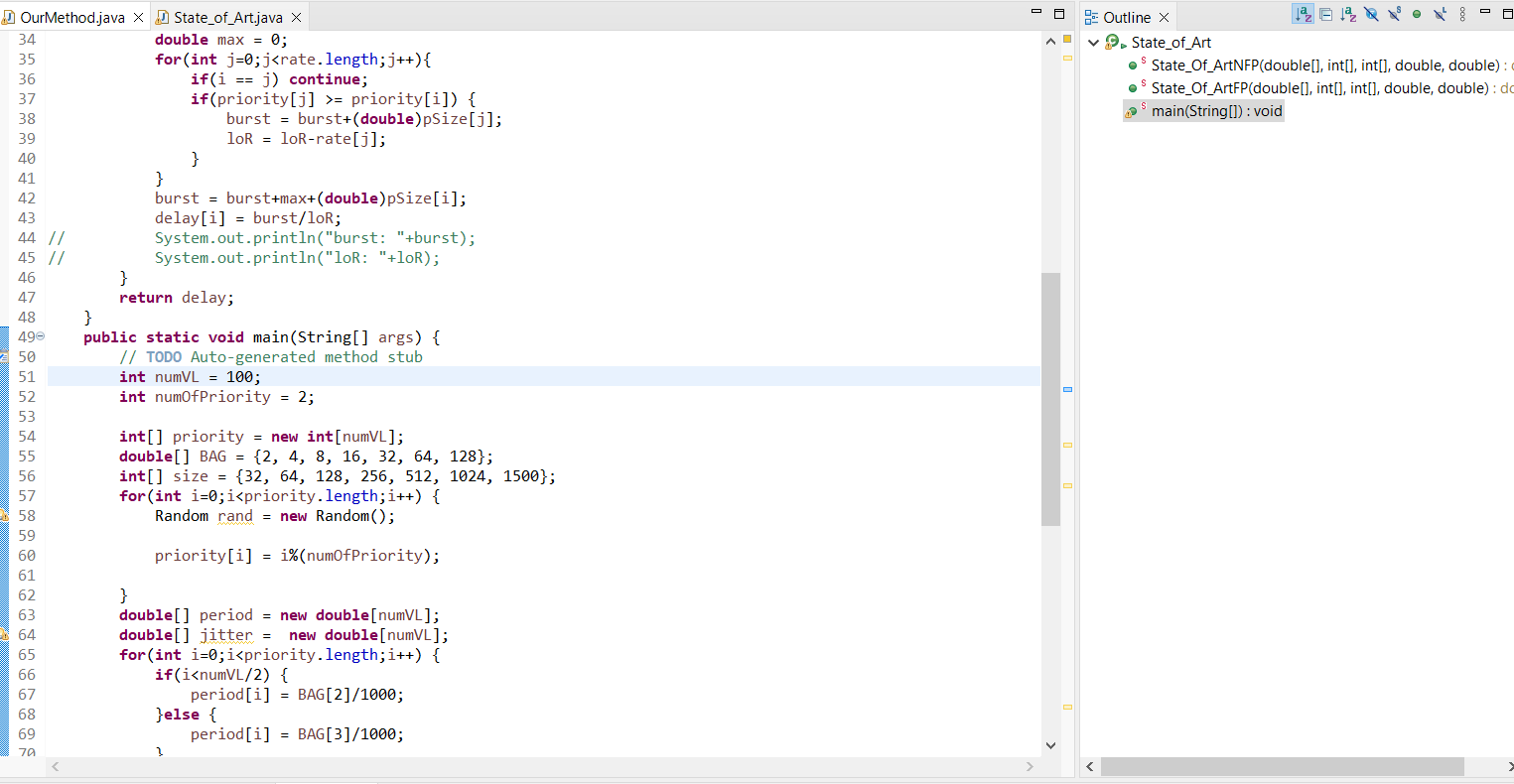
* *Then choose the directory where AFDX folder is located and import it.*



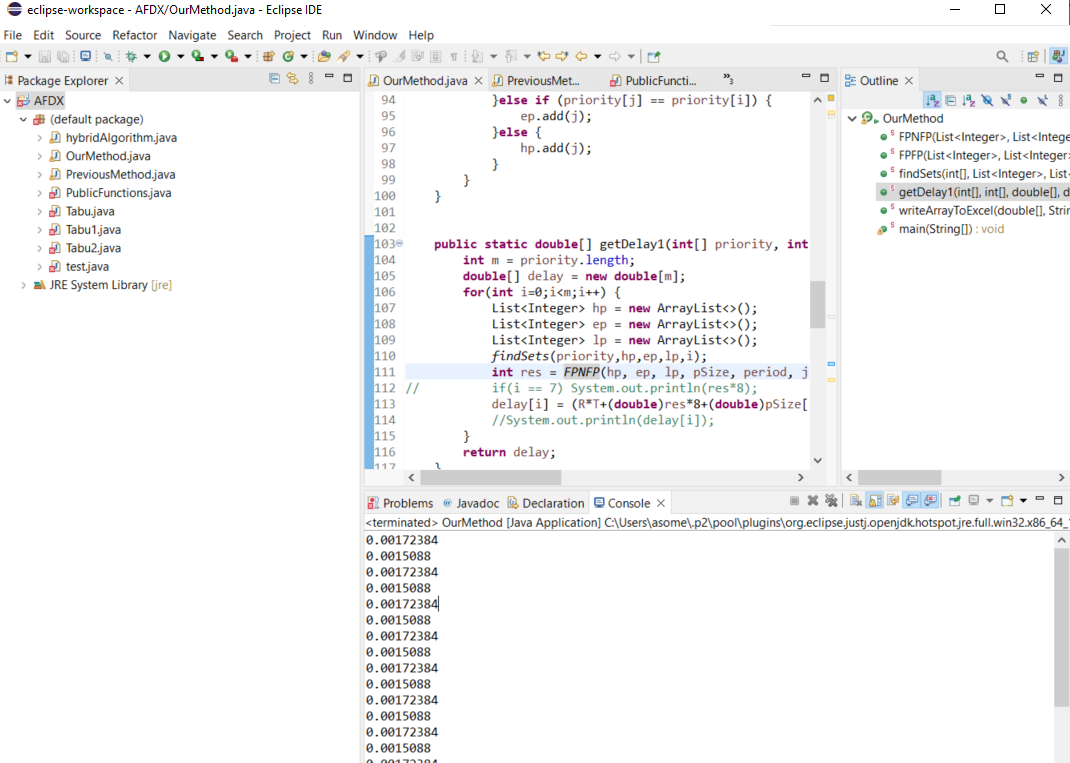
**8. Running the Program.**

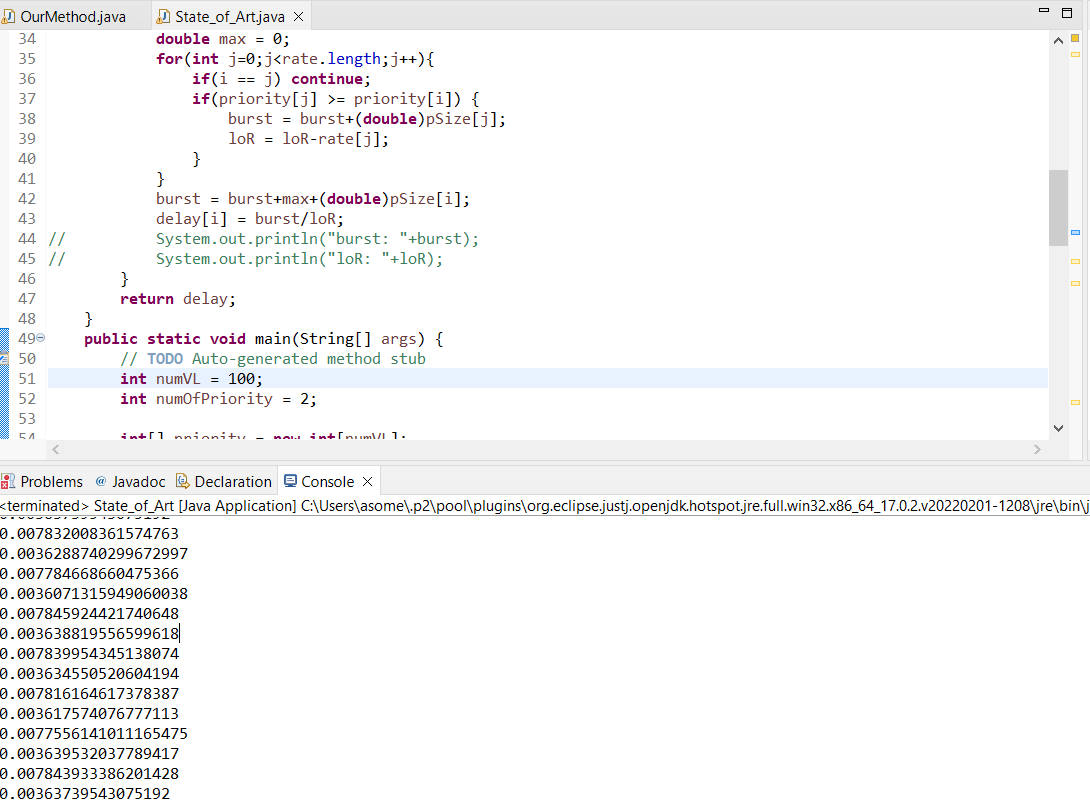
* *Inside* ***AFDX****, go to* ***OurMethod.java*** *and* ***State\_of\_Art.java***





* Then run that file and record the result.





Inside ***OurMethod.java and State\_of\_Art.java,*** *change the required parameters to get the values of delays for several scenarios such as: When all virtual links have the same priority, different priorities with or without frame preemption.*

***A****.* ***Get Delays for Reality Conforming method when all VLS have the same priority.***

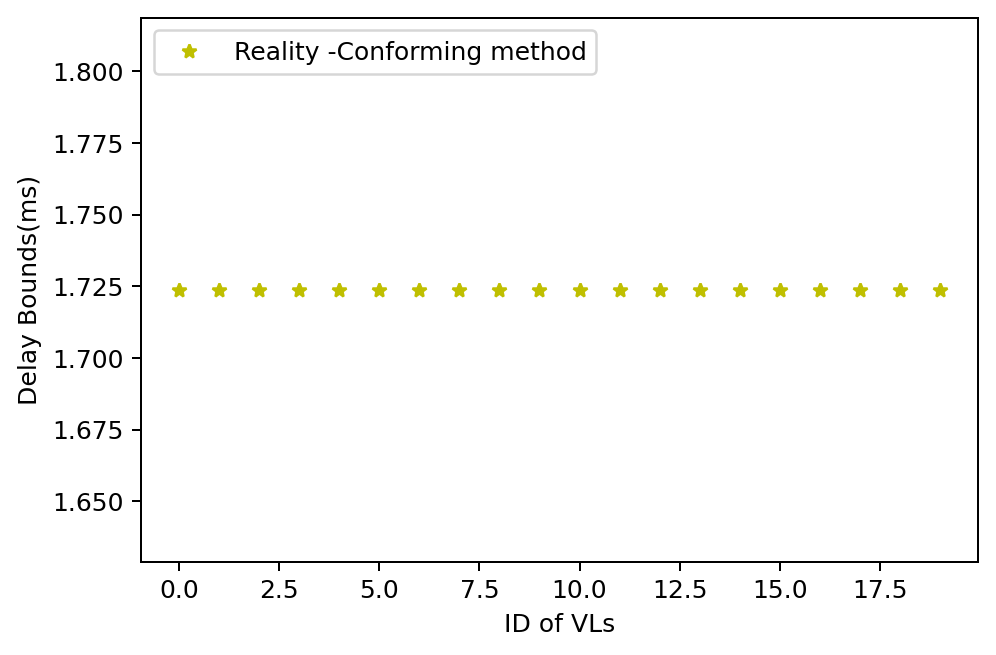
***-*** *Go to* ***OurMethod.java*** *file*

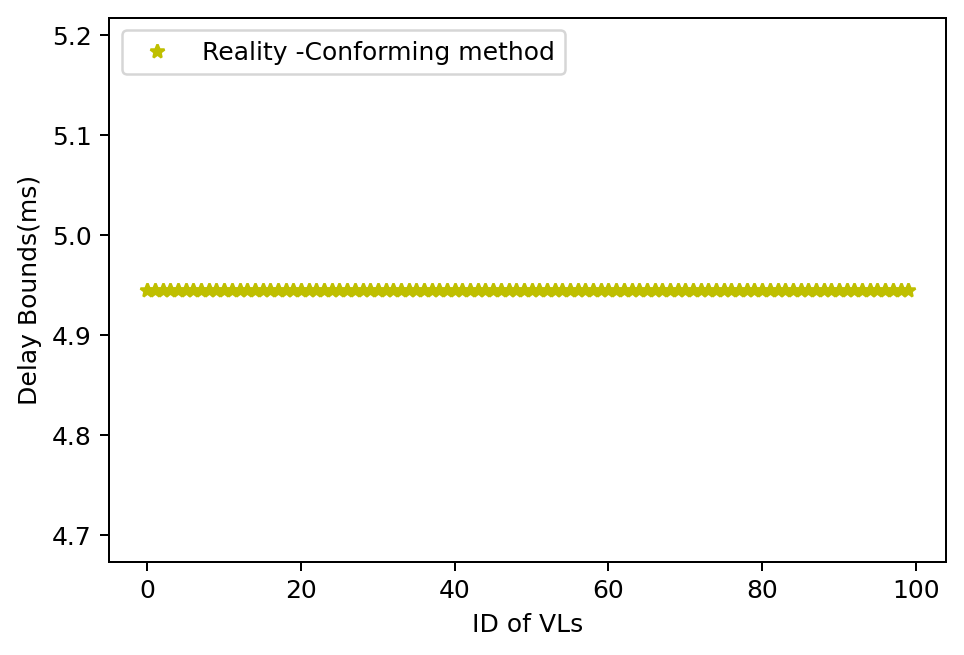
*- Inside* ***getDelay1*** *method, set* ***res*** *=* ***FPNFP*** *or* ***FPFP***

*- In* ***main*** *method, set* ***numVL*** *(no of virtual links) to 20 or 100 depending upon no of VL’s used.*

*- In* ***main*** *method, set* ***numP*** *(priority) to 1, indicate all VLs have same priority.*

*- Run the code and record the delay values to plot the graph.*





***B. Get Delays for State of Art method when all VLS have the same priority.***

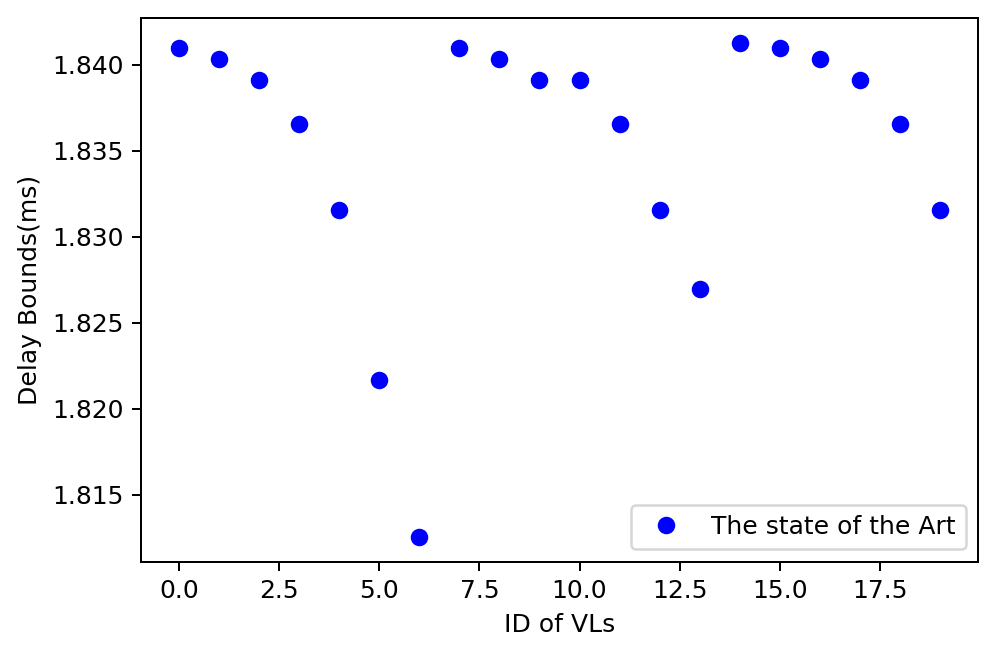
*- Go to* ***State\_of\_Art.java*** *file*

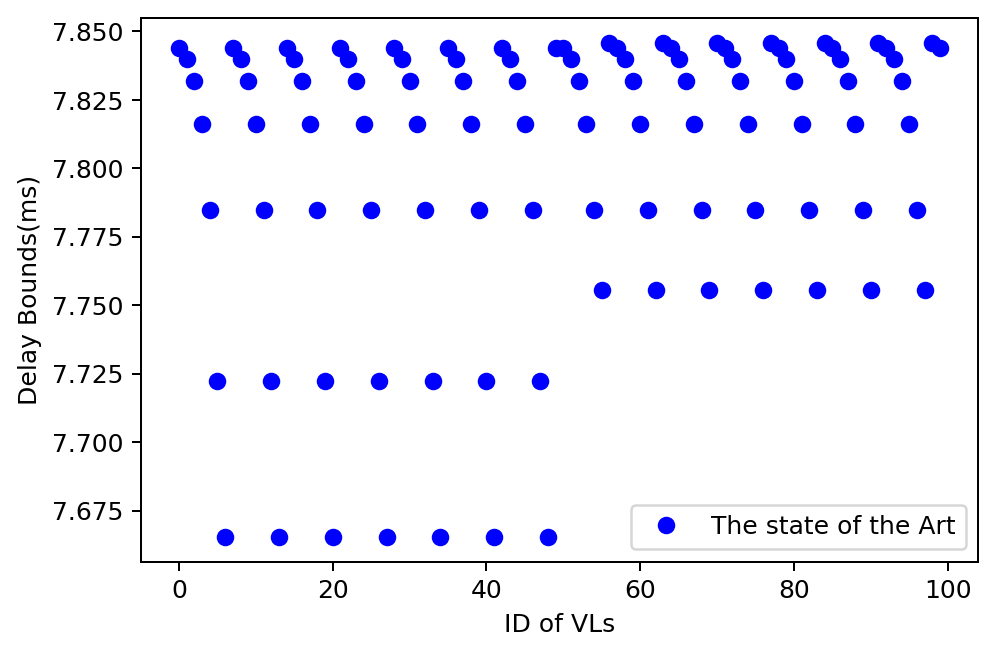
*- In* ***main*** *method, set* ***numVL*** *(no of virtual links) to 20 or 100 depending upon no of VL’s used.*

*- In* ***main*** *method, set* ***numOfPriority*** *(priority) to 1, indicate all VLs have same priority.*

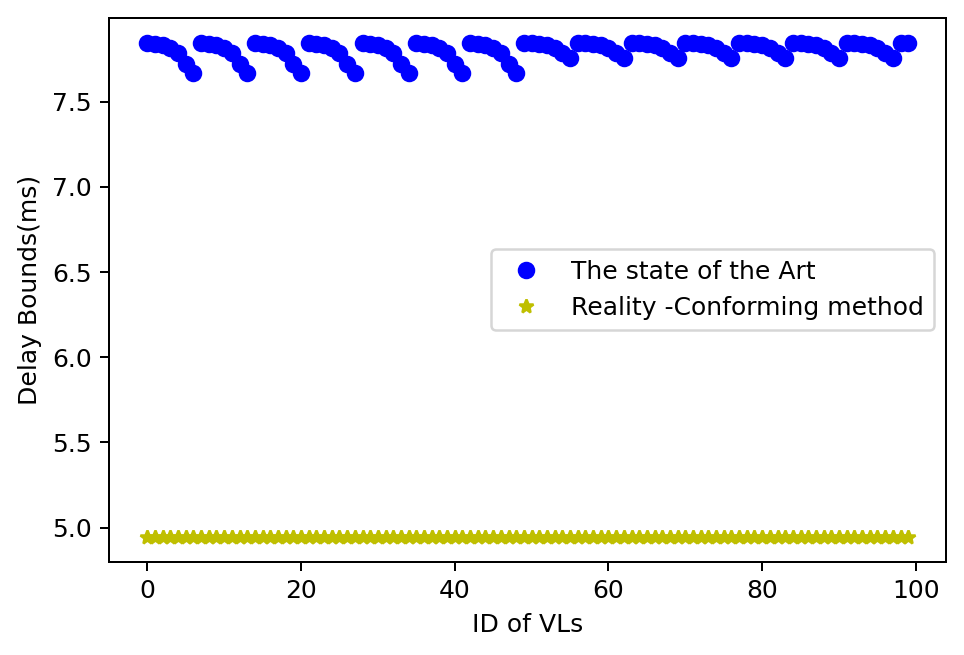
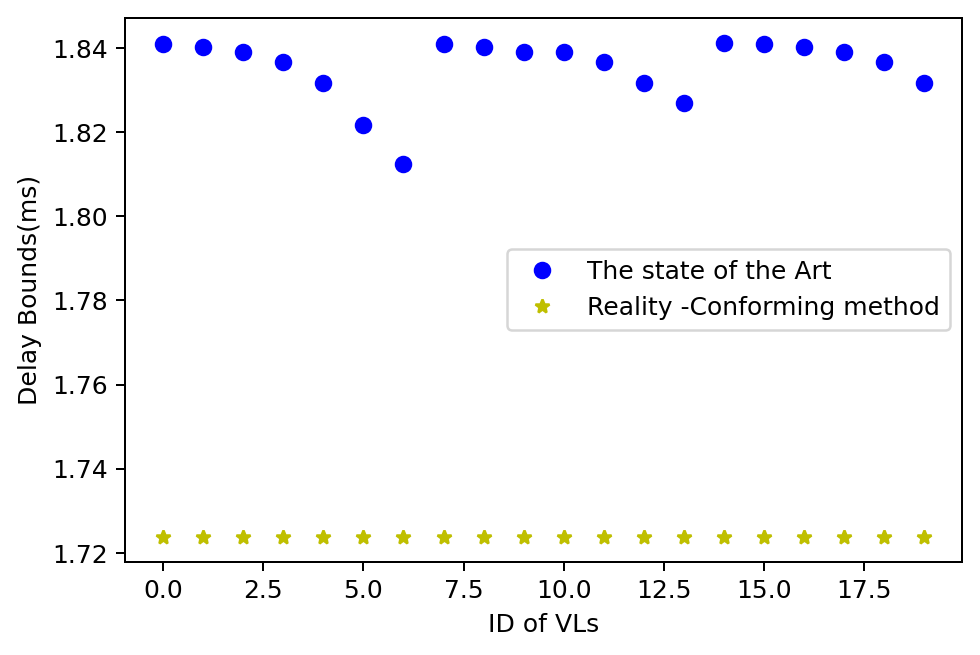
*- In* ***main*** *method, at end of the method, set* ***delay*** *to* ***State\_Of\_ArtFP*** *or* ***State\_Of\_ArtNFP***

*- Run the code and record the delay values to plot the graph.*





**Plotting both State of the Art and Reality Conforming Method, we get following results:**



***C. Get Delays for Reality Conforming method when there are two priorities without frame preemption.***

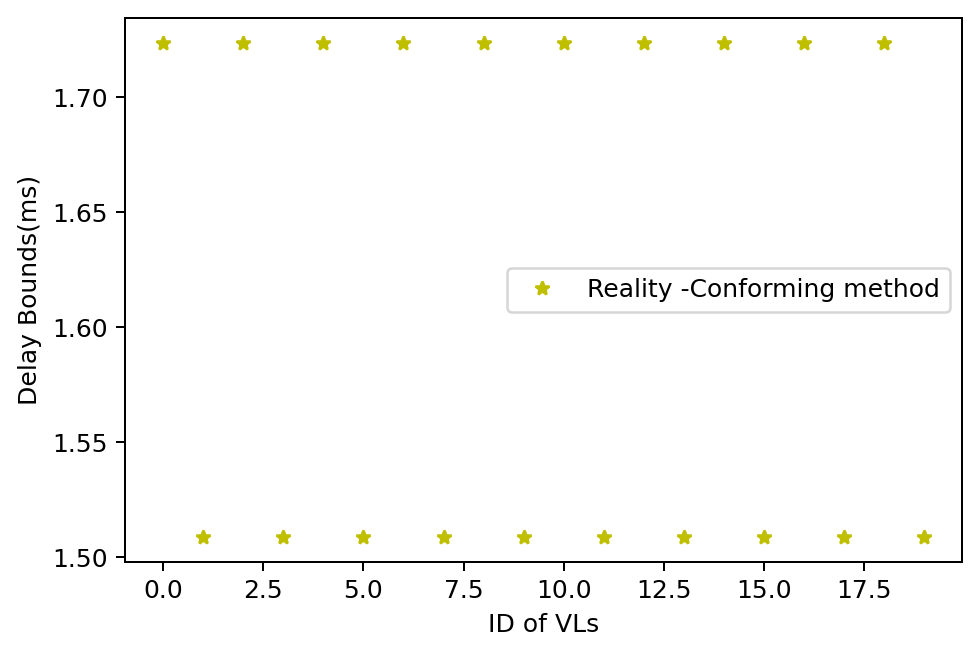
*- Go to* ***OurMethod.java*** *file*

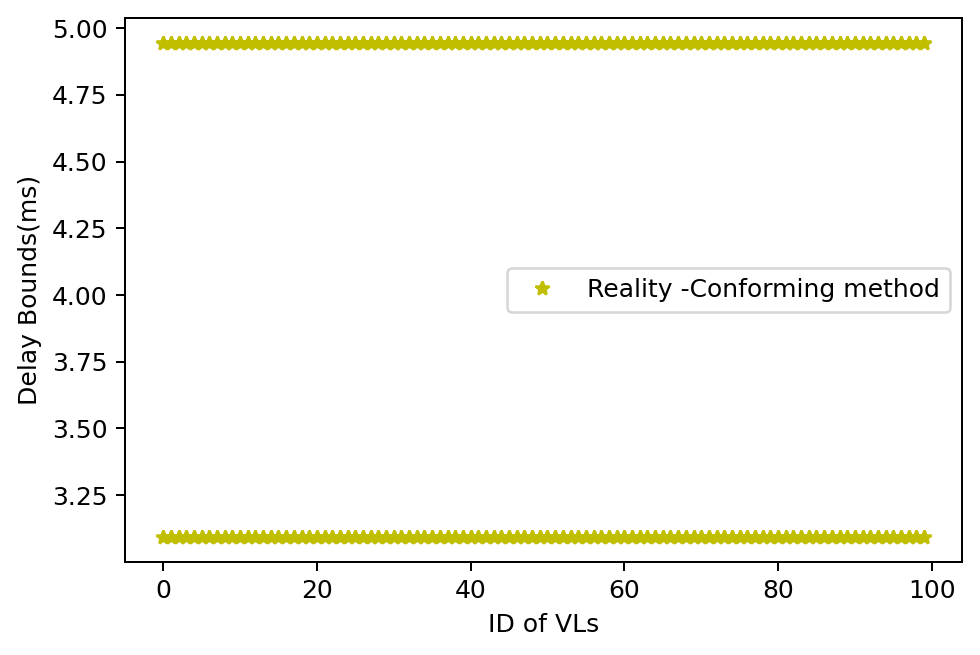
*- Inside* ***getDelay1*** *method, set* ***res*** *=* ***FPNFP***

*- In* ***main*** *method, set* ***numVL*** *(no of virtual links) to 20 or 100 depending upon no of VL’s used.*

*- In* ***main*** *method, set* ***numP*** *(priority) to 2, indicate VLs have different priority*

*- Run the code and record the delay values to plot the graph.*





***D. Get Delays for State of Art Method when there are two priorities without frame preemption.***

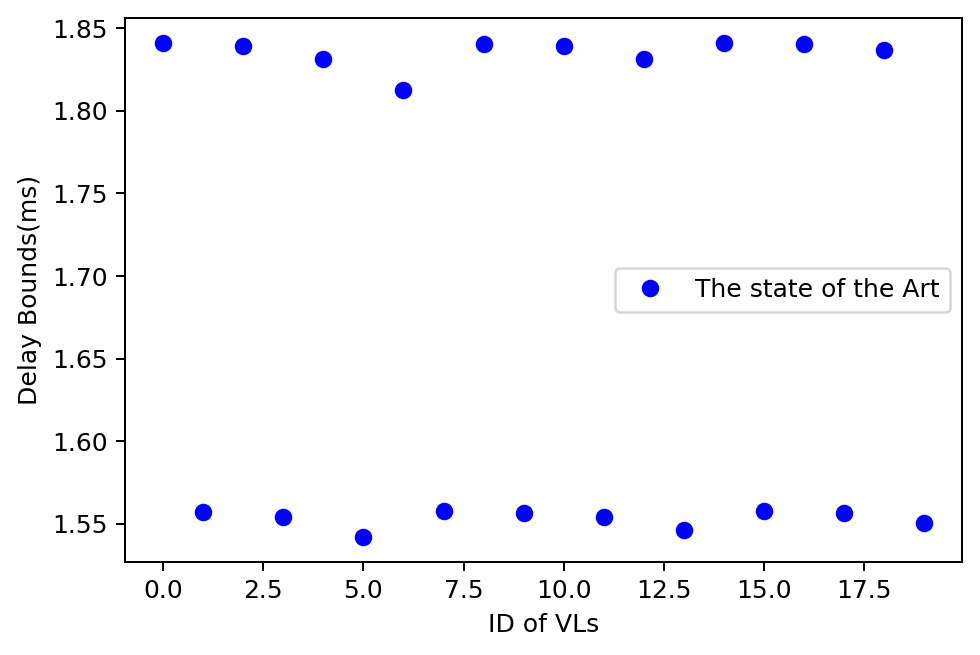
*- Go to* ***State\_of\_Art.java*** *file*

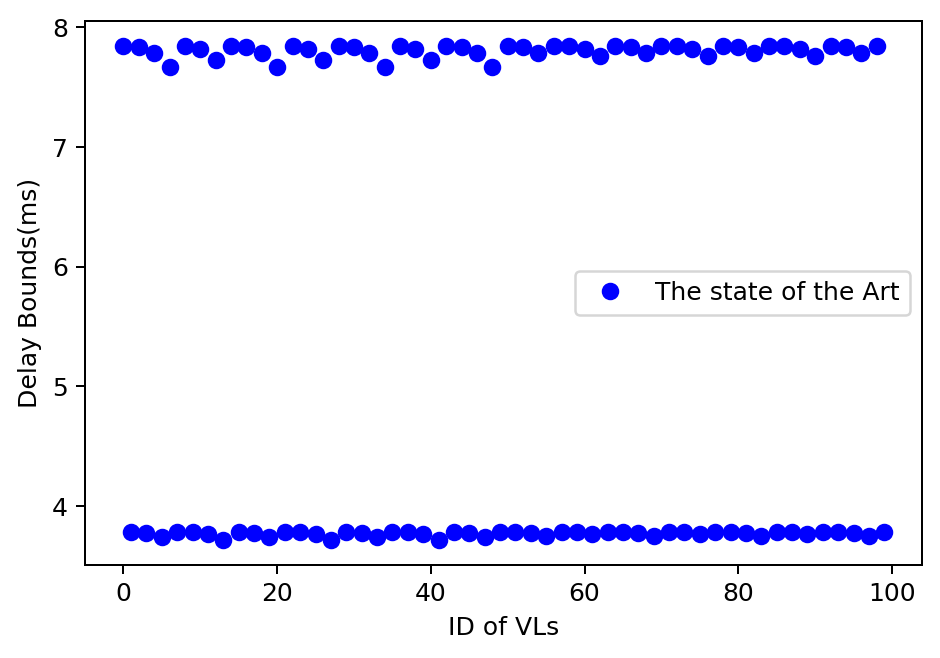
*- In* ***main*** *method, set* ***numVL*** *(no of virtual links) to 20 or 100 depending upon no of VL’s used.*

*- In* ***main*** *method, set* ***numOfPriority*** *(priority) to 2, indicate VLs have different priority.*

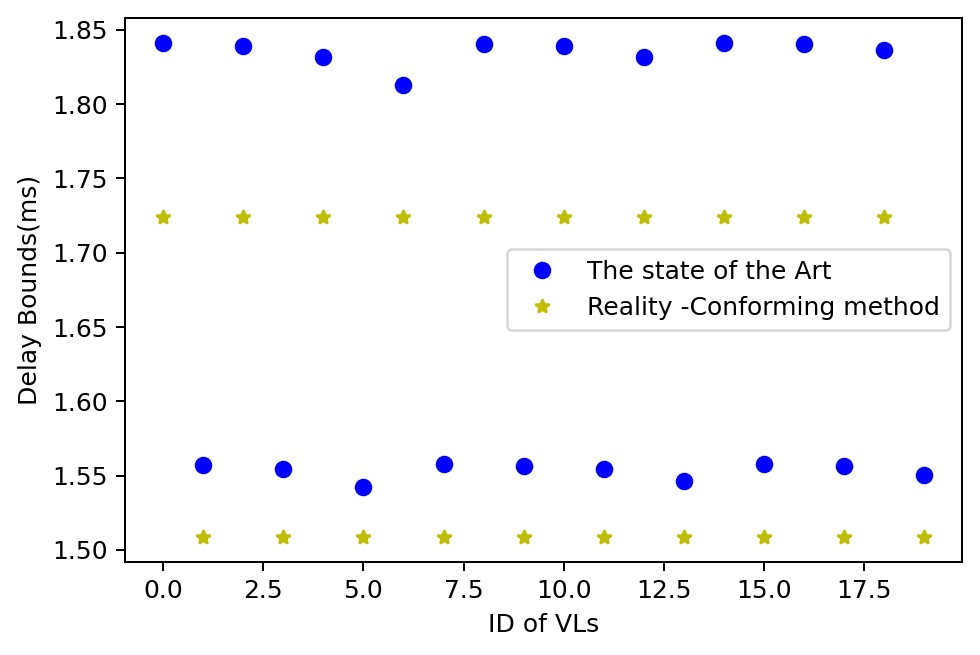
*- In* ***main*** *method, at end of the method, set* ***delay*** *to* ***State\_Of\_ArtNFP***

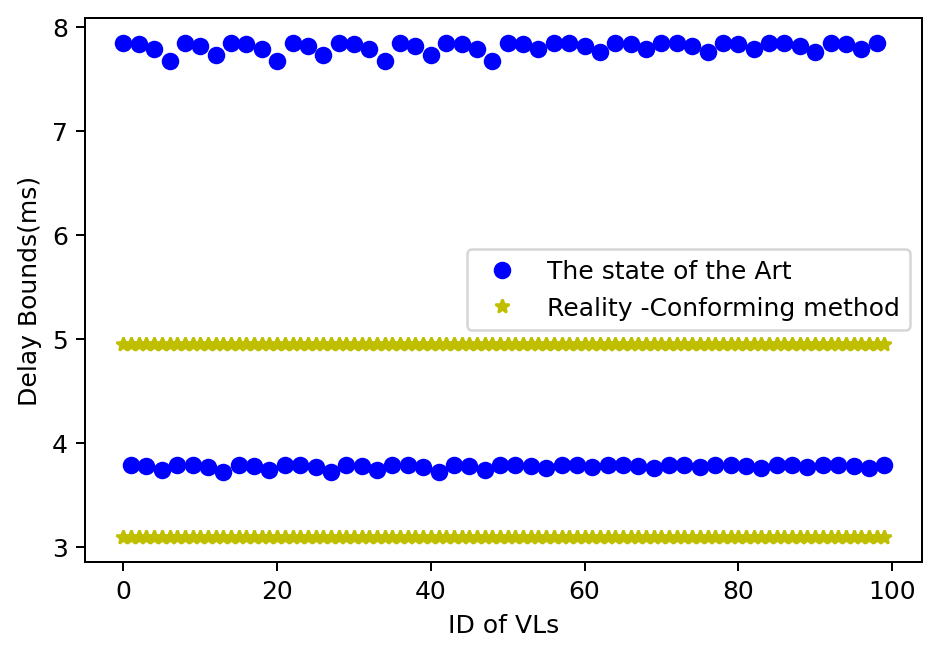
*- Run the code and record the delay values to plot the graph.*





**Plotting both State of the Art and Reality Conforming Method, we get following results:**





***E. Get Delays for Reality Conforming method when there are two priorities with frame preemption.***

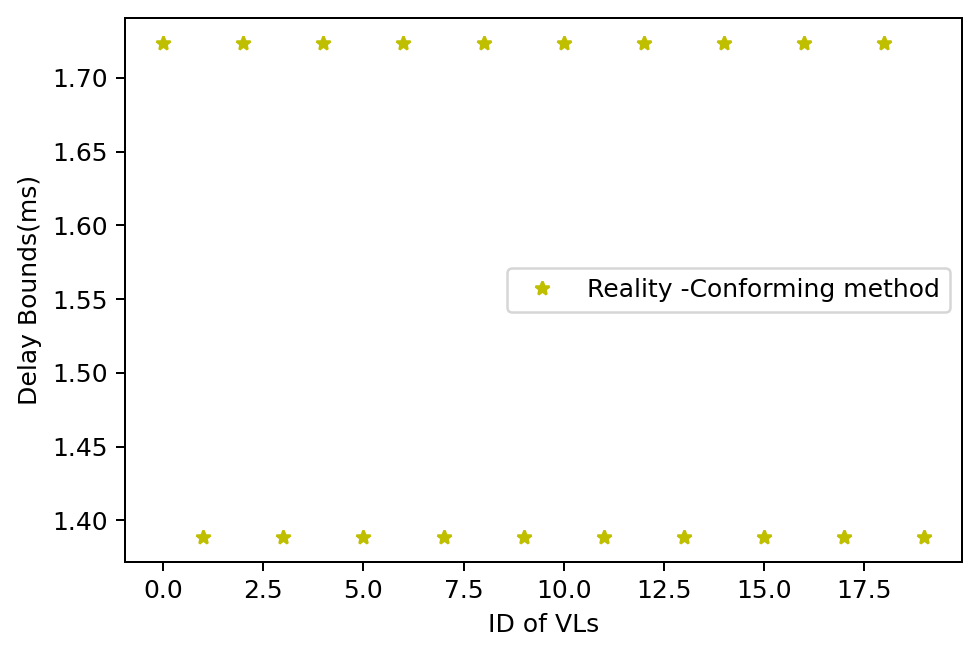
*- Go to* ***OurMethod.java*** *file*

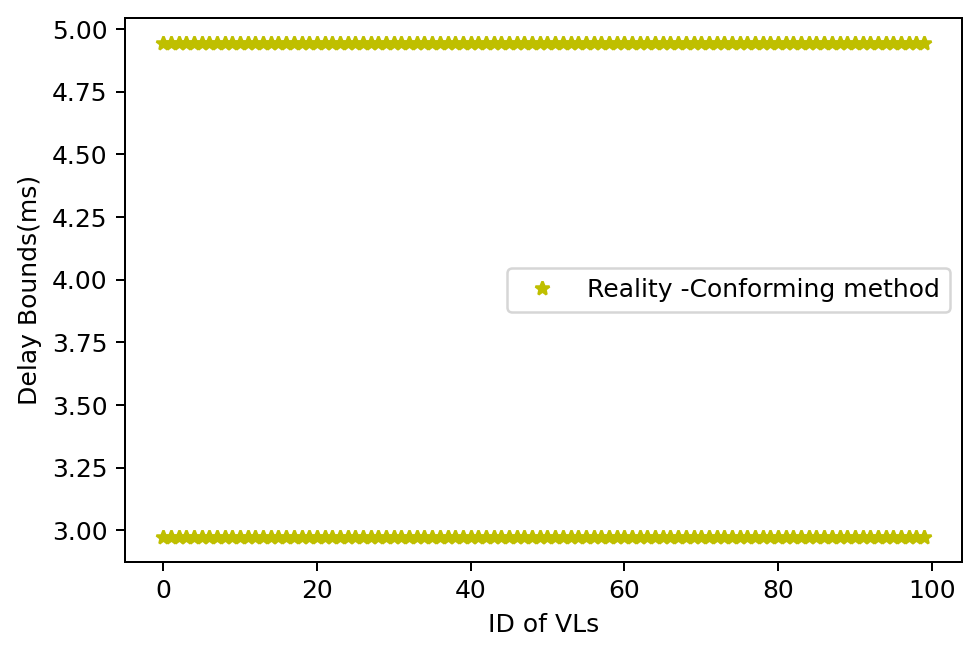
*- Inside* ***getDelay1*** *method, set* ***res*** *=*  ***FPFP***

*- In* ***main*** *method, set* ***numVL*** *(no of virtual links) to 20 or 100 depending upon no of VL’s used.*

*- In* ***main*** *method, set* ***numP*** *(priority) to 2, indicate all VLs have same priority*

*- Run the code and record the delay values to plot the graph.*





***F. Get Delays for State of Art method when there are two priorities with frame preemption.***

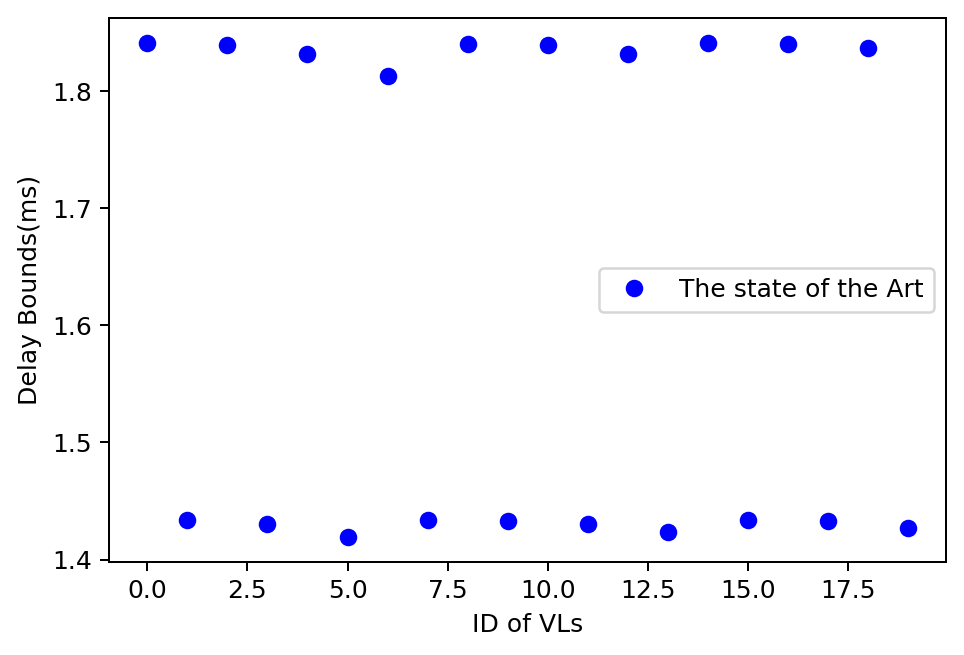
*- Go to* ***State\_of\_Art.java*** *file*

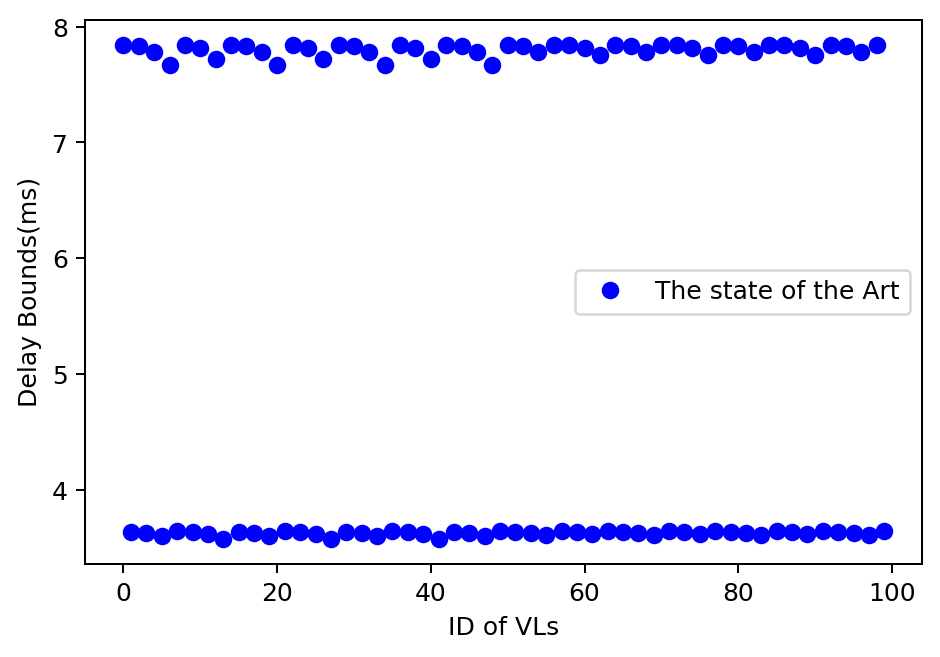
*- In* ***main*** *method, set* ***numVL*** *(no of virtual links) to 20 or 100 depending upon no of VL’s used.*

*- In* ***main*** *method, set* ***numOfPriority*** *(priority) to 2, indicate VLs have different priority.*

*- In* ***main*** *method, at end of the method, set* ***delay*** *to* ***State\_Of\_ArtFP***

*- Run the code and record the delay values to plot the graph.*





**Plotting both State of the Art and Reality Conforming Method, we get following results:**

