Introduction

The models, assumptions and algorithms are explained in the RTN 2016.

Installation

It is recommended to obtain the most recent version on GitHub.
Being based on Java tsncf should run on all major platforms. Following external dependencies are needed and have been verified:

- Apache Commons CLI 1.3.1 for Command Line Parsing

- JGraphT 0.9.1 jgrapht-core for the internal data-structures and algorithms

- JGraphx 2.0.0.1 and JGraph-ext 0.9.1 (Included in the JGraphT download) for visualization (not mandatory)

- JUnit 4.12 and Hamcrest 2.0.0.0 For unit-testing (not mandatory)

- SLF4J 1.7.13 and LogBack for logging (not mandatory)
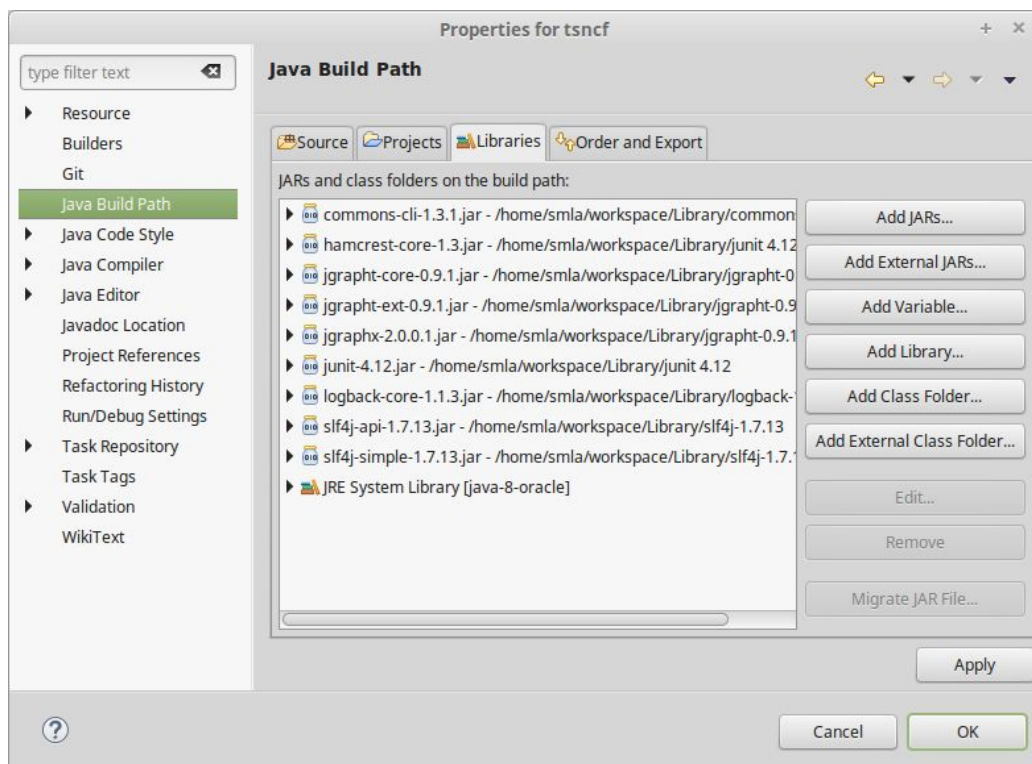
Usage

The folder /resources includes some example files. The solution for these can be displayed using the -display command where the -net argument refers to architecture GraphML files and -app argument for XML application files.

$ Java -jar tsncf -net <file> -app <file>

Following optional arguments can be specified :

Command	Function	Default
-display	Visualizes the solution	Off
-verbose	Verbose logging	Off
-K <val>	Sets the value of K to <val>	50
-Solver <sol>	Use a non default solver (e.g. <exhaustive>) [1]	GRASP

I.e. to run the motivational example from the RTN 2016 paper one would type :

$ Java -jar tsncf -net
./resources/architecture/MOTIV.XML -app
./resources/application/MOTIV_T1.XML -display

To obtain the following visual output :

---

[1] If the exhaustive solver is used, the runtime depends on the value of K. For small examples K = 4 seems reasonable. For larger examples K = 2 should be used.
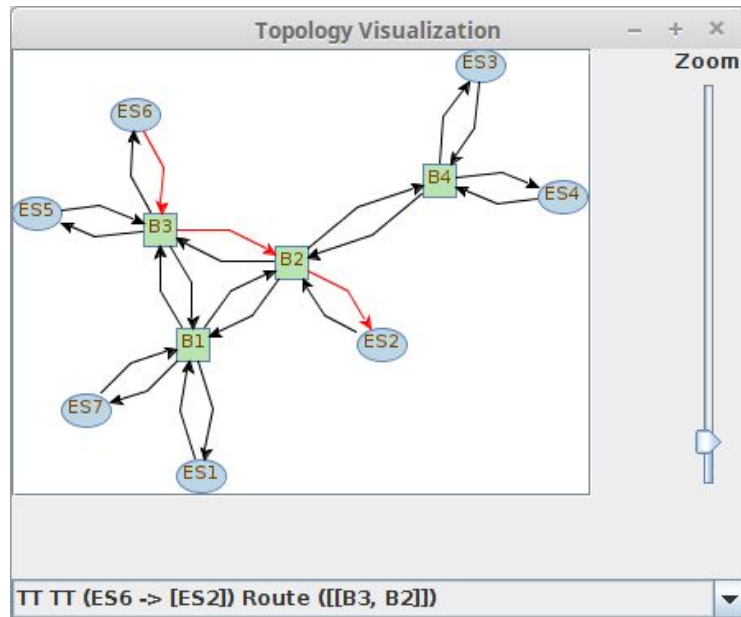
Where the individual routes can be inspected using the drop-down menu in the bottom.

Input files

Architecture
GraphML schema is used to define the architecture. http://graphml.graphdrawing.org/.
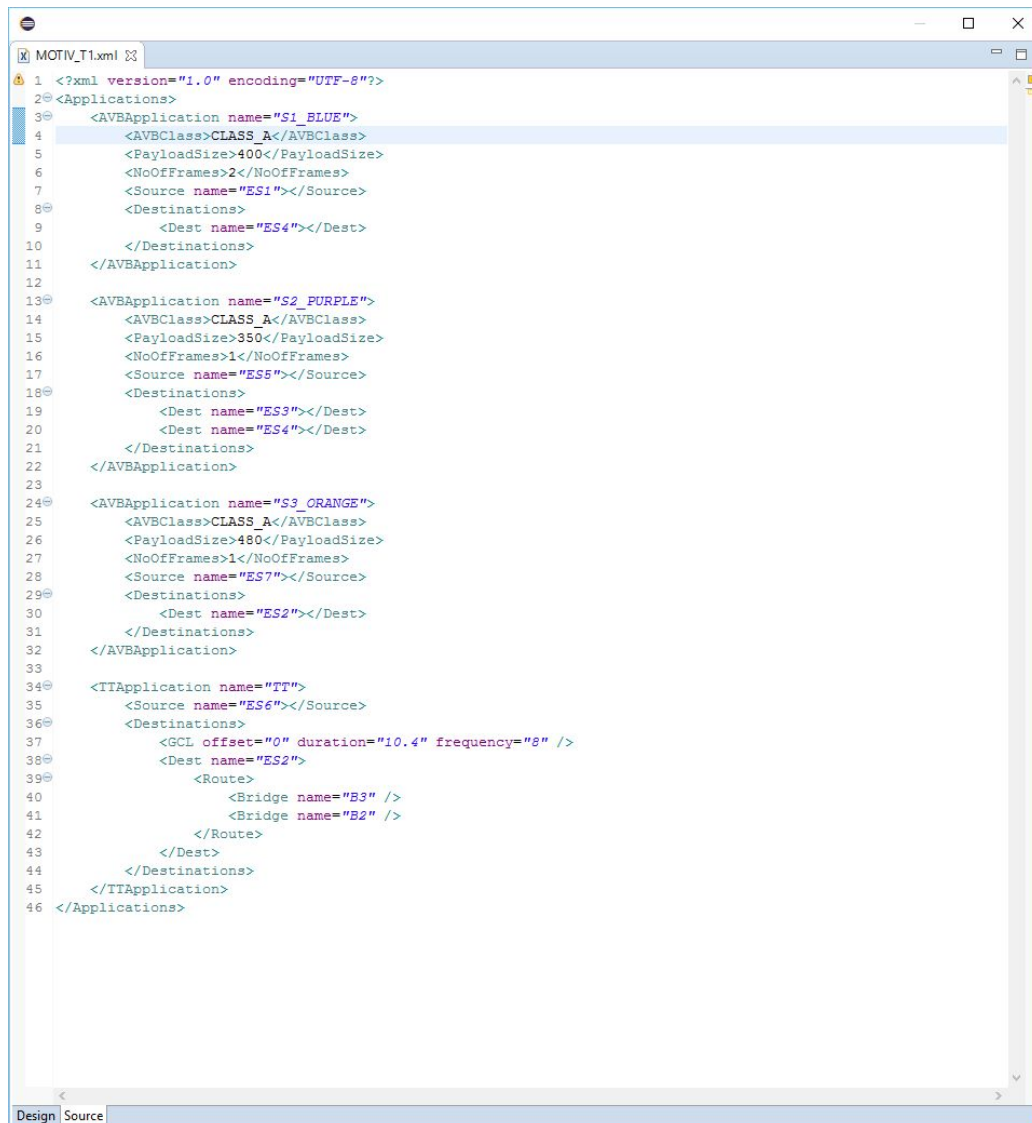


```xml
 1 <?xml version="1.0" encoding="UTF-8"?>
 2 <graphml xmlns="http://graphml.graphdrawing.org/xmlns"
 3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 4      xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
 5          http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
 6
 7   <graph id="G" edgedefault="undirected">
 8     <node id="ES1"/>
 9     <node id="ES2"/>
10     <node id="ES3"/>
11     <node id="ES4"/>
12     <node id="ES5"/>
13     <node id="ES6"/>
14     <node id="ES7"/>
15
16     <node id="B1"/>
17     <node id="B2"/>
18     <node id="B3"/>
19     <node id="B4"/>
20
21     <edge source="ES1" target="B1"/>
22     <edge source="ES7" target="B1"/>
23
24     <edge source="ES2" target="B2"/>
25
26     <edge source="ES5" target="B3"/>
27     <edge source="ES6" target="B3"/>
28
29     <edge source="ES3" target="B4"/>
30     <edge source="ES4" target="B4"/>
31
32     <edge source="B1" target="B2"/>
33     <edge source="B1" target="B3"/>
34     <edge source="B2" target="B3"/>
35     <edge source="B2" target="B4"/>
36   </graph>
37 </graphml>
```

<u>Application</u>

Custom XML format with two types of applications, AVBApplication for AVB Streams and TTApplication for TT Streams.

```xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <Applications>
3      <AVBApplication name="S1_BLUE">
4          <AVBClass>CLASS_A</AVBClass>
5          <PayloadSize>400</PayloadSize>
6          <NoOfFrames>2</NoOfFrames>
7          <Source name="ES1"></Source>
8          <Destinations>
9              <Dest name="ES4"></Dest>
10         </Destinations>
11     </AVBApplication>
12
13     <AVBApplication name="S2_PURPLE">
14         <AVBClass>CLASS_A</AVBClass>
15         <PayloadSize>350</PayloadSize>
16         <NoOfFrames>1</NoOfFrames>
17         <Source name="ES5"></Source>
18         <Destinations>
19             <Dest name="ES3"></Dest>
20             <Dest name="ES4"></Dest>
21         </Destinations>
22     </AVBApplication>
23
24     <AVBApplication name="S3_ORANGE">
25         <AVBClass>CLASS_A</AVBClass>
26         <PayloadSize>480</PayloadSize>
27         <NoOfFrames>1</NoOfFrames>
28         <Source name="ES7"></Source>
29         <Destinations>
30             <Dest name="ES2"></Dest>
31         </Destinations>
32     </AVBApplication>
33
34     <TTApplication name="TT">
35         <Source name="ES6"></Source>
36         <Destinations>
37             <GCL offset="0" duration="10.4" frequency="8" />
38             <Dest name="ES2">
39                 <Route>
40                     <Bridge name="B3" />
41                     <Bridge name="B2" />
42                 </Route>
43             </Dest>
44         </Destinations>
45     </TTApplication>
46 </Applications>
```

For AVB applications, only the source and destinations as well as type, size and noOfFrames within a 500us period is needed. For TT applications, besides the complete path from ''Source'' to ''Dest'', the GCLs are defined in terms of us (microseconds) where "offset" refers to the global offset of the sending node and is automatically calculated as "offset" + X*"duration" for the Xth node on the path. "Duration" is the time it takes to send the frame. "frequency" refers to the amount of frames sent within a 500us period (i.e. frequency = 8 means that evert 500/8 = 62.5us a TT frame is send).

Limitations / Bugs

- Instead of calculating the hyperperiod automatically, it is currently set to 500 us. I think only the GCLEdge.java needs to updated.
- Multicast TT-streams are not supported
- To change the rate and delay of a GCL Edge you need to go to TopologyParser.java and change the constants RATE and DEVICE_DELAY.
- To change weight-values, when searching, you need to go to ModifiedAVBEvaluatorCost.java and change values of w1,w2,w3
- To change/tune the GRASP parameters you need to go to GraspSolver.java
- Preprocessing / validation of input files is practically non-existent, so malformed inputs may (silently) result in "weird" errors