# Testing Deterministic Avionics Networks Using Orthogonal Arrays

**M. Efe ALTUNTAŞ**[1,2], Muhammed Şeker[1,2], Pınar Kışla[1,2], Eyüp Can Akpolat[2,5], İbrahim Hökelek[3,5], M. Selim Demir[3], Erhan Akdoğan[4]

[1]*Yıldız Technical University, Dept. of Avionics Engineering (Turkey)*

[2]*Turkish Aerospace Industries, Inc.(TAI) (Turkey)*

[3]*TÜBİTAK, Informatics and Information Security Research Center (BİLGEM)(Turkey)*

[4]*Yildiz Technical University, Dept. of Mechatronics Engineering (Turkey)*

[5]*İstanbul Technical University, Dept. of Mechatronics Engineering (Turkey)*

*Contact email: muhammedefe.altuntas@tai.com.tr*

**03.10.2021**

# Muhammed Efe ALTUNTAŞ

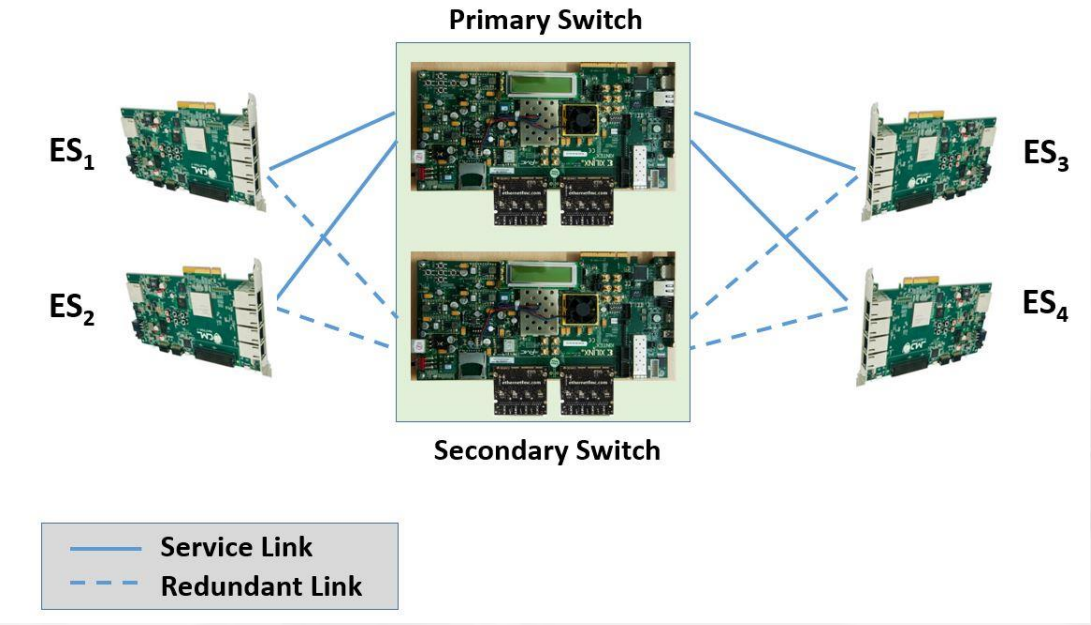
- Undergraduate
  - İstanbul Şehir University – Electrical Electronics Engineering – (2017)
  - İstanbul Şehir University – Computer Engineering – (2018) (Double Major Program)
- Graduate
  - Yıldız Technical University – Avionics Engineering – (present)
- Work Experience
  - Hardware Design and Verification Engineer – Turkish Aerospace Industries, Inc. – (2018 – present)
- Research Interest
  - FPGA
  - UVM
  - ARINC 664
  - Avionic test systems
  - RISC-V

# Motivation

- To verify deterministic avionics network device
  - Thousands of virtual links with large number of configuration options
- Orthogonal Array (OA): formal test generation technique which reduces the number of test cases with respect to coverage of all test space
- Orthogonal array types:
  - Same level OA
  - Multi-level OA
- In this work, we present an approach: **«Chained OA»** is based on multi-level OA
  - Multiple and consecutive applications of multi-level OA's
  - Allows for preserving the number of combinations for a set of selected features

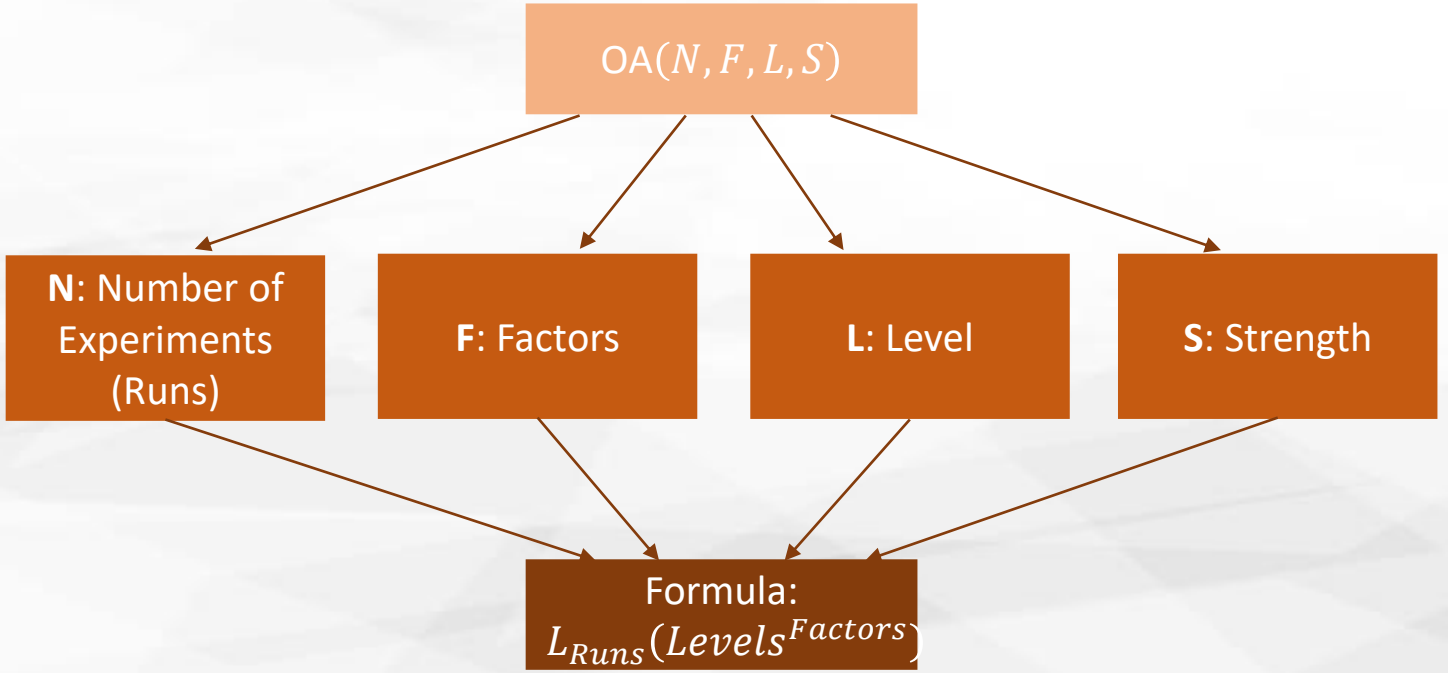
# ARINC 664 – Avionics Deterministic Network

- ARINC 664 has 3 main elements:
  - Switch
  - End system (ES)
  - Virtual Links (VL)
- DTN switches have different functions
  - Filtering
  - Policing
  - Switching
  - End System
- Switch configuration parameters :
  - Numbers of VLs
    - Source and destination end systems
    - Minimum ($L_{min}$) and maximum ($L_{max}$) frame sizes
    - Bandwidth allocation gap (BAG)



# Orthogonal Arrays

- **F** (**Factor**): number of columns in OA, where each column corresponds to a variable of an experiment
- **L (Level),** representing the number of distinct values that a factor can take
- **S (Strength)**, the number of rows needed to cover S-wise combinations of variables
- Same level orthogonal array:
  - All factors have the same numbers of level
- Mixed level orthogonal array:
  - Each factor may have different number of levels



# Example of Same Level OA

- An example OA :

  OA(4,3,2,2) – N: 4, F: 3, L: 2, S: 2

- Number of all combination:
  - $level^{factor}$ = $2^3$ = 8 (see in Table A)
- After applying OA(4,3,2,2), number of all possibilities :
  - $level^{strength}$ = $2^2$ (see in Table B)

Table A

| Runs | $f_1$ | $f_2$ | $f_3$ |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 |
| 3 | 0 | 1 | 0 |
| 4 | 0 | 1 | 1 |
| 5 | 1 | 0 | 0 |
| 6 | 1 | 0 | 1 |
| 7 | 1 | 1 | 0 |
| 8 | 1 | 1 | 1 |

Table B

| Runs | $f_1$ | $f_2$ | $f_3$ |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 1 |
| 3 | 1 | 0 | 1 |
| 4 | 1 | 1 | 0 |

# Example of Multi - Level OA

$$OA\left(N, L_1^{|F_1|} L_2^{|F_2|} \ \dots \ L_v^{|F_v|}, L, S\right)$$

- The example OA is shown:

  OA(12, $2^4$ $3^1$, 2)

  - N = 12, $F_1$ = 4, $F_2$ = 1, $L_1$ = 2, $L_2$ = 3 and S=2
  - The total number of combinations: $Level^{Factor}$: $2^4$ x $3^1$ = 48
  - After OA application: $Level^{Strength}$:

    $2^2$ x $3^1$ = 12 (see in Table)

| Runs | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 1 | 0 |
| 3 | 1 | 0 | 1 | 0 | 0 |
| 4 | 1 | 1 | 1 | 1 | 0 |
| 5 | 0 | 0 | 1 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 |
| 7 | 1 | 0 | 0 | 0 | 1 |
| 8 | 1 | 1 | 0 | 1 | 1 |
| 9 | 0 | 0 | 1 | 1 | 2 |
| 10 | 0 | 1 | 0 | 0 | 2 |
| 11 | 1 | 0 | 0 | 1 | 2 |
| 12 | 1 | 1 | 1 | 0 | 2 |

# Test Space for ARINC 664 Switch

| No. of Ports | BAG (ms) | $L_{min}$ | $L_{max}$ | Priority | Incoming VLs to a Port | Outgoing VLs from a Port |
|---|---|---|---|---|---|---|
| 1 | 1 | 64 | 1518 | High | 192 | 192 |
| 2 | 2 | 128 | 1400 | Low | 384 | 384 |
| 3 | 4 | 150 | 1300 | | 576 | 576 |
| 4 | 8 | 200 | 1200 | | 960 | 960 |
| | 16 | 300 | 1100 | | 1152 | 1152 |
| | 32 | 400 | 1000 | | 1920 | 1920 |
| | 64 | 512 | 900 | | 2880 | 2880 |
| | 128 | 750 | 800 | | 3840 | 3840 |

$$C = 8^{1152} \times 8^{1152} \times 8^{1152} \times 2^{1152} \times 4$$

C is the combination number of VL configuration parameters for 1152 VLs
OA generates only one configuration file for this table

# Chained OA with Regulated Test Space

| Number of VLs for $OA_{g1}$ | | | |
|---|---|---|---|
| Port 1 | Port 2 | **Port 3** | **Port 4** |
| 192 | 192 | 192 | 192 |
| 384 | 384 | 384 | 384 |
| 576 | 576 | 576 | 576 |
| 960 | 960 | 960 | 960 |
| 1152 | 1152 | 1152 | 1152 |
| 1920 | 1920 | 1920 | 1920 |
| 2880 | 2880 | 2880 | 2880 |
| 3840 | 3840 | 3840 | 3840 |

$OA_{g1}(N_{g1}\ L_{g1}^{F_{g1}}\ S_{g1})$

$OA_{g1}$ specify # of VL for each incoming ports

Result of $OA_{g1}$ indicates # of runs for each $OA_{i,j}$

$$\begin{bmatrix} n_{1,1} & \cdots & n_{1,F_{g1}} \\ \vdots & \ddots & \vdots \\ n_{N_{g1},1} & \cdots & n_{N_{g1},F_{g1}} \end{bmatrix}$$

$OA_{g1}(256\ 8^4 2)$

Switch supports 4096 VLs
Runs in $OA_{g1}$ with higher VLs are eliminated
256 rows reduced to 143

| OA Result for $OA_{g1}$ | | | | |
|---|---|---|---|---|
| # of Runs | Port 1 | Port 2 | Port 3 | Port 4 |
| Runs 1 | 192 | 384 | 960 | 1920 |
| Runs 2 | 576 | 1152 | 192 | 384 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| Runs 143 | 1920 | 960 | 576 | 192 |

# Chained OA with Regulated Test Space

| VL Parameters for $OA_{i,j}$ | | | | |
|---|---|---|---|---|
| No. of Ports | BAG (ms) | $L_{min}$ | $L_{max}$ | Priority |
| 1 | 1 | 64 | 1518 | High |
| 2 | 2 | 128 | 1400 | Low |
| 3 | 4 | 150 | 1300 | |
| 4 | 8 | 200 | 1200 | |
| | 16 | 300 | 1100 | |
| | 32 | 400 | 1000 | |
| | 64 | 512 | 900 | |
| | 128 | 750 | 800 | |

$OA_{i,j}(n_{i,j}\ L_{g2}^{F_{g2}}\ S_2)$ Each $OA_{i,j}$ is VL configuration file for specified port

Total # of OA configuration file $\sum_{i=1}^{N}\sum_{j=1}^{F_{g1}} OA_{i,j}$

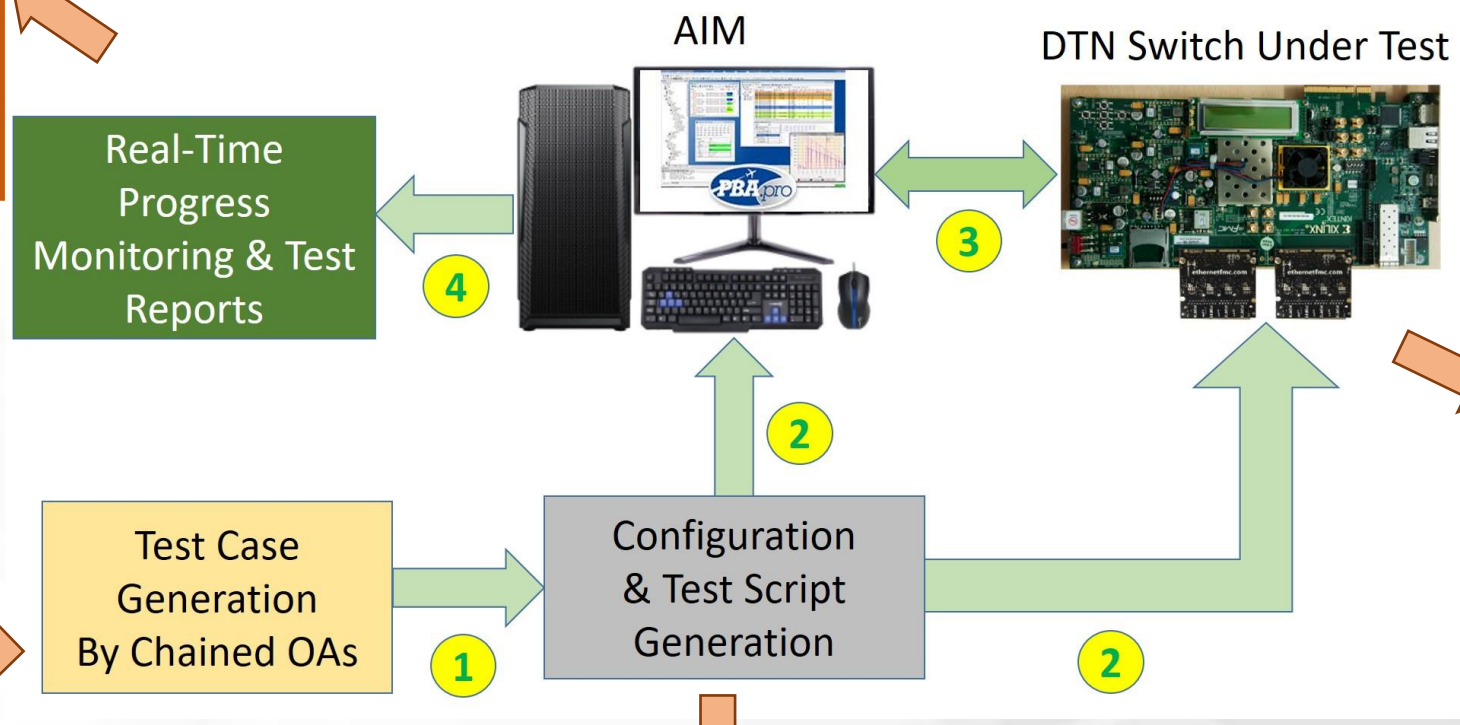
$OA_{i,j}(n_{i,j}\ 4^1\ 8^3 2^1\ 2)$

$OA_{g1}(256\ 8^4 2)$ → $OA_{i,j}(n_{i,j}\ 4^1\ 8^3 2^1\ 2)$ → $\sum_{i=1}^{143}\sum_{j=1}^{4} OA_{i,j}$ → 143 x 4 = 572 Total OA

# System Architecture of DTN Test Tool

4

Results are reported to monitoring tool




3

Tests are run with generated OA

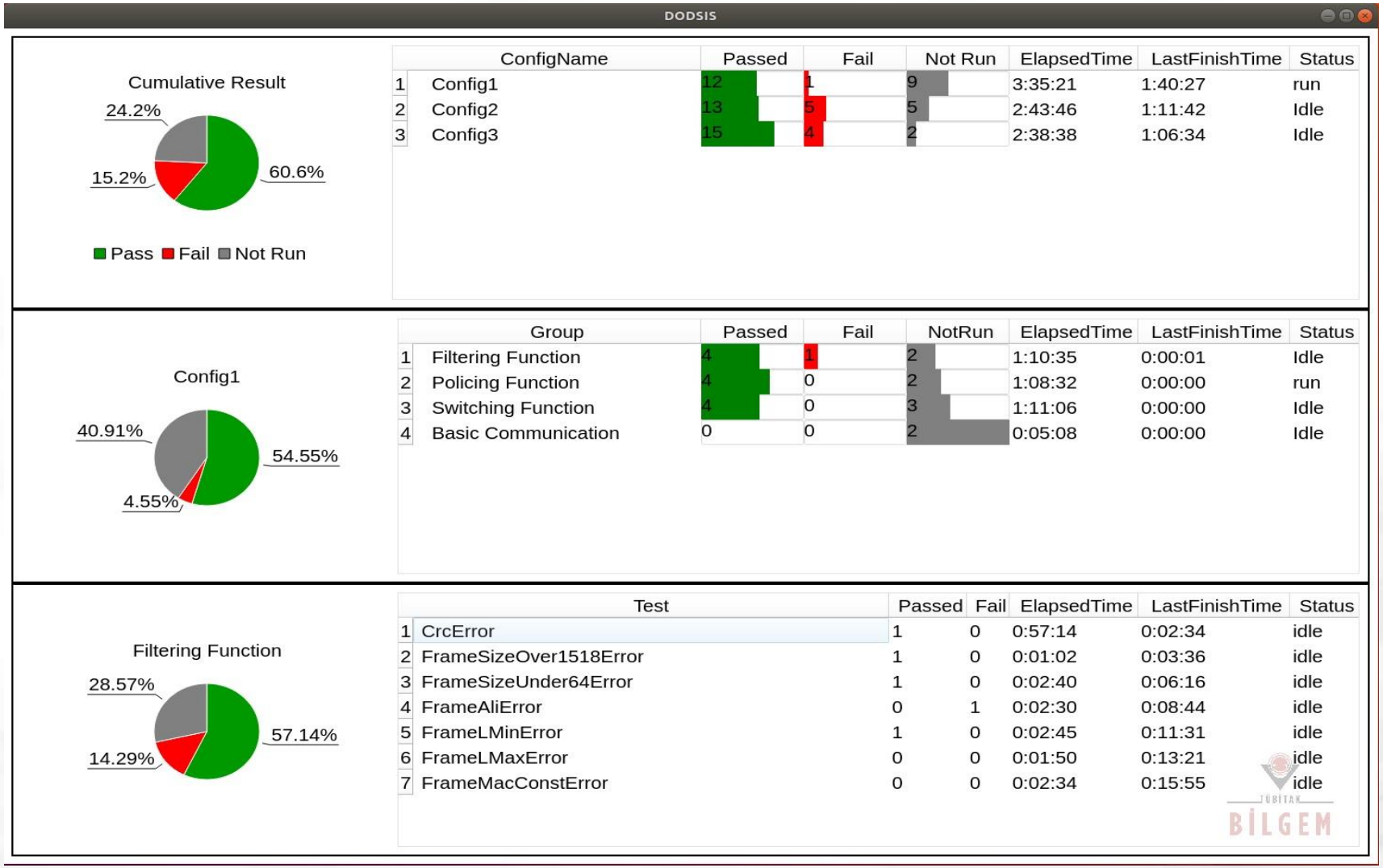
1

OAs are generated for test cases by chained method

2

OA files format are converted to configuration file format to upload AIM and switch

# GUI of Test System




| | ConfigName | Passed | Fail | Not Run | ElapsedTime | LastFinishTime | Status |
|---|---|---|---|---|---|---|---|
| 1 | Config1 | 12 | 1 | 9 | 3:35:21 | 1:40:27 | run |
| 2 | Config2 | 13 | 5 | 5 | 2:43:46 | 1:11:42 | Idle |
| 3 | Config3 | 15 | 4 | 2 | 2:38:38 | 1:06:34 | Idle |

| | Group | Passed | Fail | NotRun | ElapsedTime | LastFinishTime | Status |
|---|---|---|---|---|---|---|---|
| 1 | Filtering Function | 4 | 1 | 2 | 1:10:35 | 0:00:01 | Idle |
| 2 | Policing Function | 4 | 0 | 2 | 1:08:32 | 0:00:00 | run |
| 3 | Switching Function | 4 | 0 | 3 | 1:11:06 | 0:00:00 | Idle |
| 4 | Basic Communication | 0 | 0 | 2 | 0:05:08 | 0:00:00 | Idle |

| | Test | Passed | Fail | ElapsedTime | LastFinishTime | Status |
|---|---|---|---|---|---|---|
| 1 | CrcError | 1 | 0 | 0:57:14 | 0:02:34 | idle |
| 2 | FrameSizeOver1518Error | 1 | 0 | 0:01:02 | 0:03:36 | idle |
| 3 | FrameSizeUnder64Error | 1 | 0 | 0:02:40 | 0:06:16 | idle |
| 4 | FrameAliError | 0 | 1 | 0:02:30 | 0:08:44 | idle |
| 5 | FrameLMinError | 1 | 0 | 0:02:45 | 0:11:31 | idle |
| 6 | FrameLMaxError | 0 | 0 | 0:01:50 | 0:13:21 | idle |
| 7 | FrameMacConstError | 0 | 0 | 0:02:34 | 0:15:55 | idle |

# Results and Future Works

- Major OA methods can be inadequate for DTN switch test spaces. That's why, Chained OA is developed to use for DTN parameters.
- Chained OA reduces  the number of configuration with respect to coverage of all test space. It produced 572 different configuration files for DTN switch.
- Future extension of this work includes development of OAs that incorporate parameters defined for single and multiple end systems and reliability test suites for interoperability of switch and end systems for avionics applications.