MAC-layer Security for Time-Sensitive Switched Ethernet Networks

Venesa Watson, Prof. Dr. Christoph Ruland, Dr. Karl Waedt

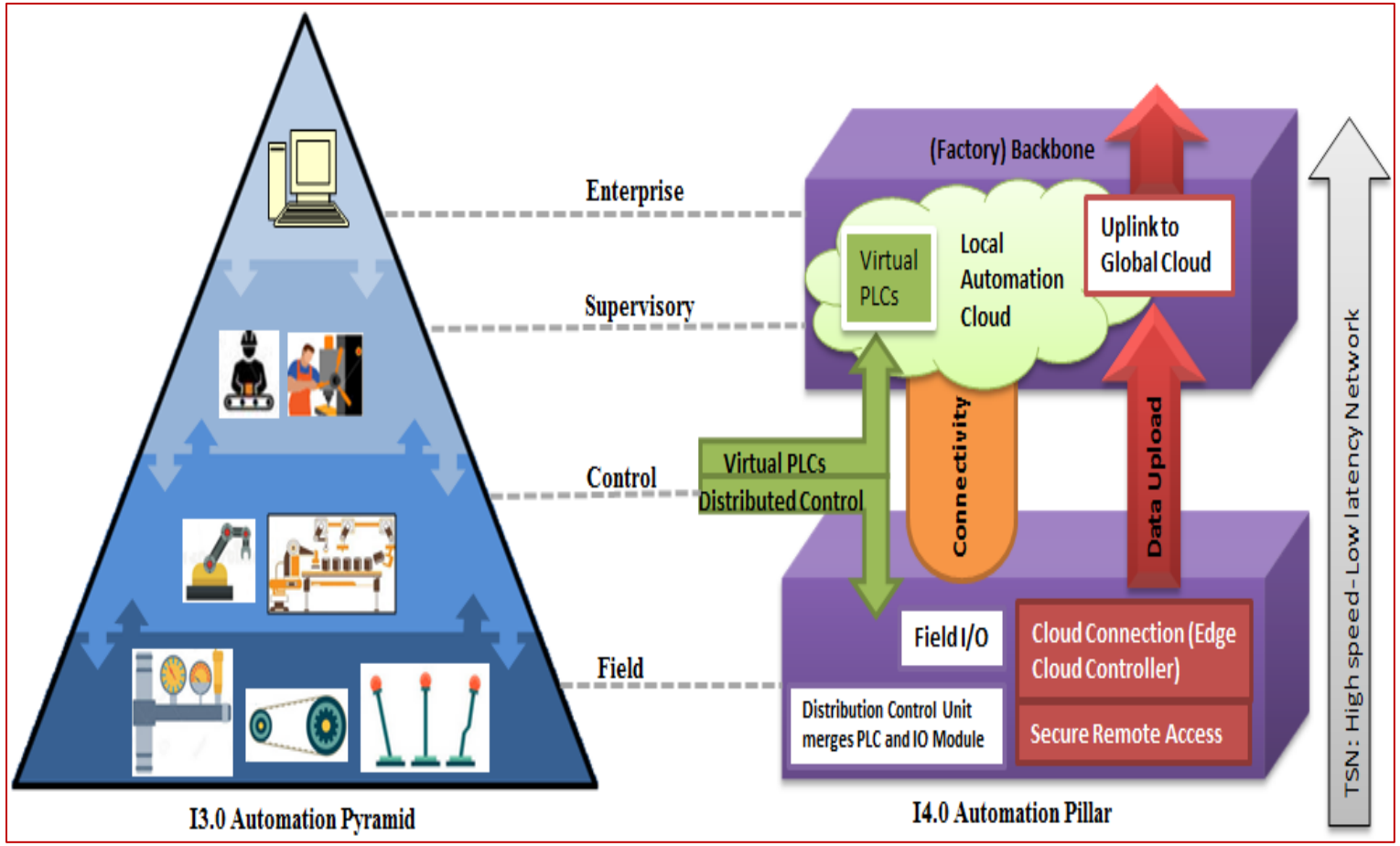
28.09.2020

Contents

- Overview
- Security Scheme (TSN-MIC)
  - Parameters
  - Concept
  - Implementation and Simulation
- Results
- Conclusion

Overview

Overview [1]



I3.0 Automation Pyramid transformation to I4.0 Automation Pillar

- Time-Sensitive Networking (TSNg) IEEE 802.1 sub-standards have been earmarked as the protocols to provide the required time- and mission-critical services in I4.0/Smart manufacturing infrastructure.
- For TSNg and other MAC-layer TSN-based technology found in critical infrastructure (e.g. AFDX in airplanes and TTE in e-vehicles), a dedicated MAC layer security scheme is necessary.

Overview [2]

Time-sensitive Ethernet-based Standards

Standard	Traffic Categorization	Traffic Shaping	Traffic Policing	Time Synchronization and Bounded Latency
ARINC 664 Part 7: AFDX	Customizable through VLs and sub-VLs	Chosen policy at End System; Token bucket and FIFO at the switch	Checks for network allowance and frame format compliance. Checks for unexpected and duplicated traffic.	No time synchronization defined. Upper (500µs) and lower bounds for latency is defined for all traffic.
IEEE 1722-2016 AVTP	4 types defined	Strict Priority Selection and Credit-based Shaper	Checks for network allowance compliance.	gPTP for time synchronization. Bounded delay enforced for time-critical traffic.
SAE AS6802 TTE	4 types defined	Chosen strict priority and non-pre-emptive algorithms	Checks for network allowance compliance. Checks for unexpected traffic.	Time synchronization is defined. Bounded delay enforced for TT and RC traffic.

Overview [3]

- Considerations for the development and demonstration of a dedicated MAC layer security scheme for TSN-based networks include:

1. Target security attribute → **Integrity**
2. Context → **Resource-restrictive environment**
3. Innovation → **Improvement on current solutions**

Security Scheme –TSN-MIC

Security Scheme – TSN-MIC

Time-Sensitive Network – Message Integrity Code (TSN-MIC)

- Lightweight integrity protection designed specifically for MAC layer TSN services
- Additional mechanisms that are included for improved security, while still observing the performance requirements of time-critical transmissions
- Designed with an online key management and key change-over mechanism, with feedback mechanisms to detect and restrict the propagation of error related to intentional and unintentional actions

Security Scheme –TSN-MIC: Parameters

TSN-MIC – Parameters [1]

- The key parameters of the TSN-MIC scheme are:
  - the lightweight cryptography algorithms for the generation of the MIC (or Message Authentication Code/MAC) and
  - the key management protocol for the key lifecycle processes (key generation, key deactivation, key update and key change-over)

TSN-MIC – Parameters [2]

ISO/IEC lightweight cryptographic standards for message integrity

ISO/IEC 29192-6 MIC

LightMAC	**Tsudik's Keymode**	**Chaskey-12**
An n-bit block cipher from ISO/IEC 29192-2 or ISO/IEC 18033-3	A lightweight hash-function from ISO/IEC 29192-5	No additional requirements
A length t in bits of the MIC	---	
A counter size s, i.e. the number of bits allocated to represent the counter value, where $0 \leq s \leq n$	---	

TSN-MIC – Parameters [3]

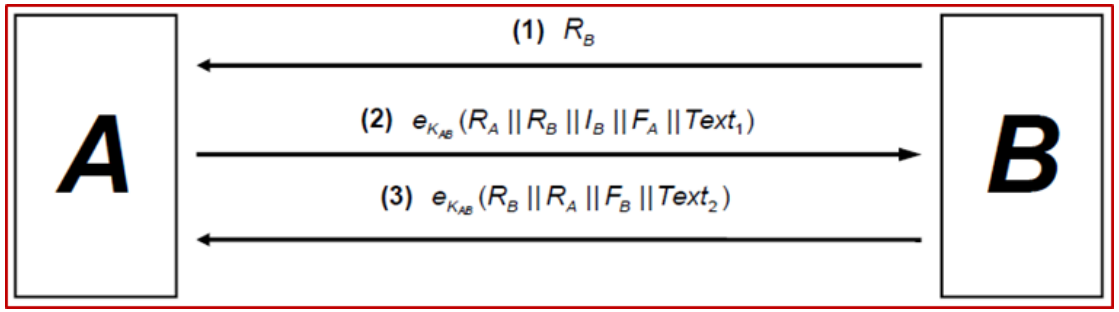
Chaskey-12 Advantages

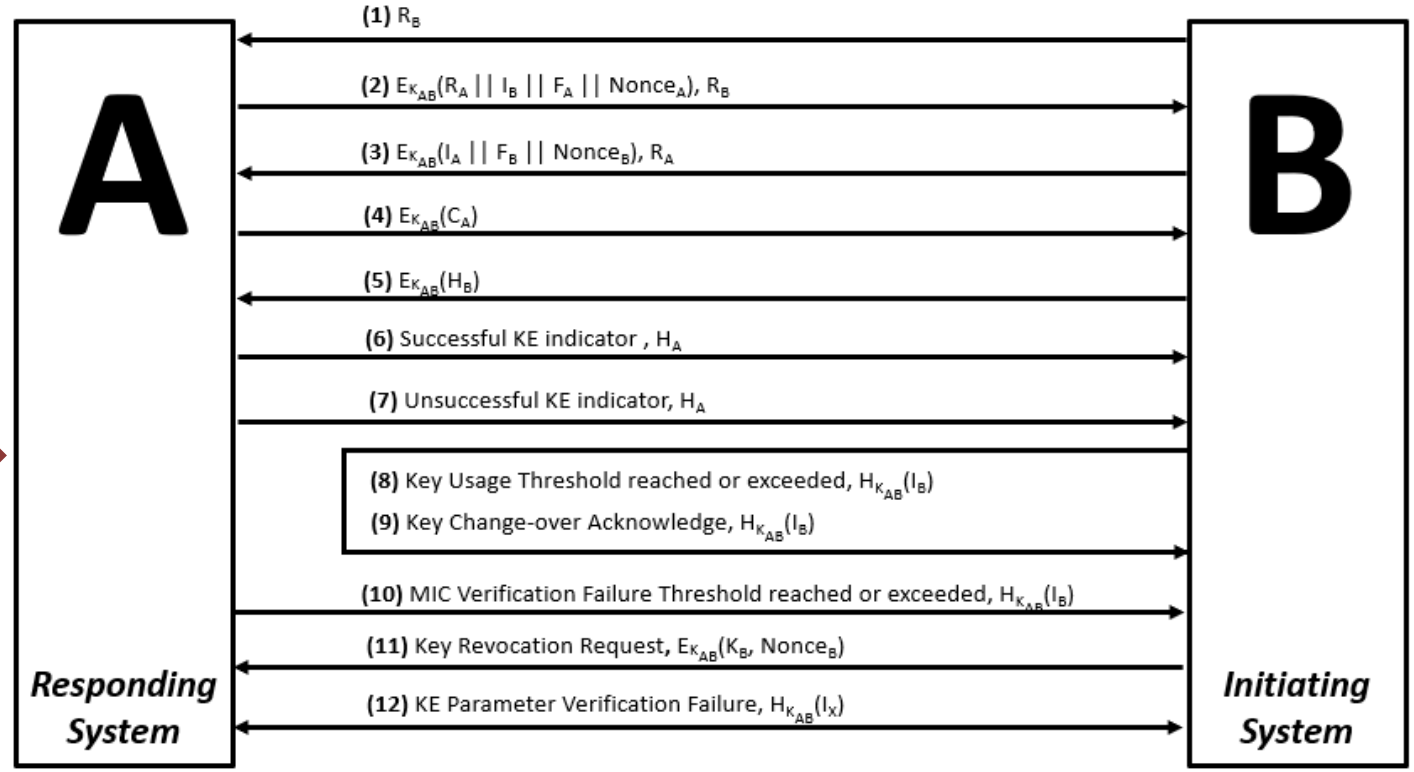
- **Dedicated design**
- **Cross-Platform versatility**
- **Efficient implementation**
- **Resistance against timing attack**
- **Tag truncation**
- **Nonces are optional**
- **Provably secure**
- **Patent-Free**

Algorithm	Speed based on selected microcontrollers	
	ARM Cortex-M3/M4	ARM-Cortex-M0
Chaskey-8	7.0 cycles/byte	16.9 cycles/byte
Chaskey-12	*10.5 cycles/byte*	*25.4 cycles/byte*
AES-128-ECB	66.7 cycles/byte	112.7 cycles/byte
AES-128-CMAC	89.4 cycles/byte	136.5 cycles/byte

TSN-MIC – Parameters [4]

ISO/IEC 11770-2 mechanism 6 - Algorithm for key establishment



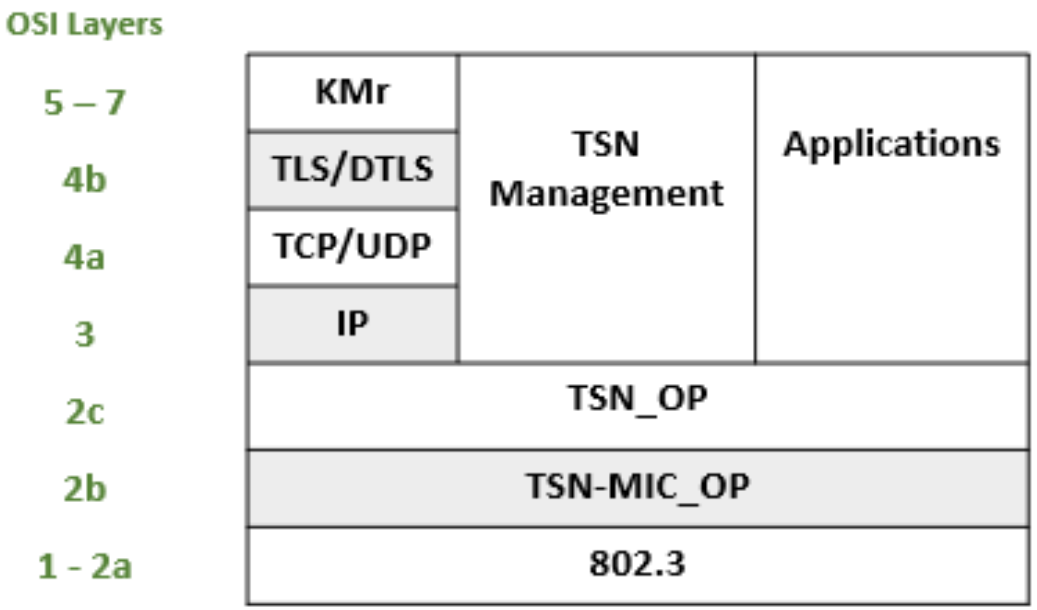


Security Scheme –TSN-MIC: Concept

TSN-MIC – Concept [1]

- The TSN-MIC scheme is implemented at the OSI MAC-layer, just below the services/operations of the selected TSN protocol (e.g. AFDX, AVTP, TTE, etc.)

- This design means that messages are first processed by the TSN-MIC operations before the messages are handled by the TSN operations

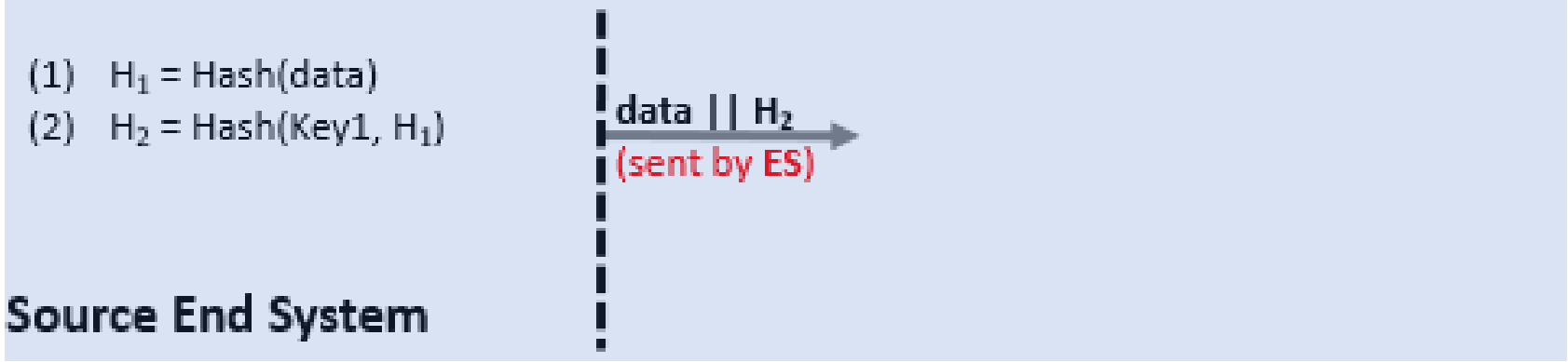


TSN-MIC – Concept [2]

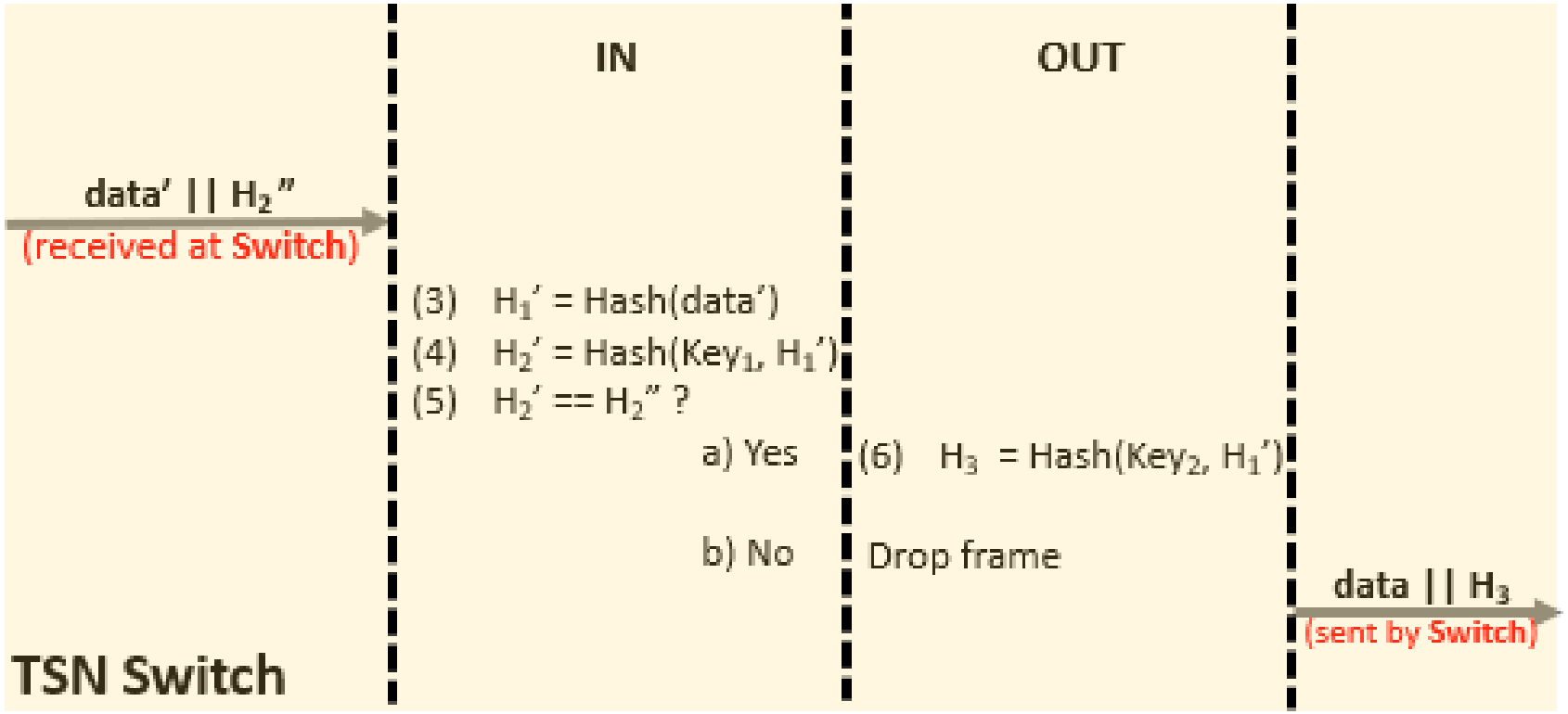
TSN-MIC Calculations

- Across a simple network (a Source End System, a TSN Switch and a Destination End System), there are seven (7) TSN-MIC calculations
- Namely, there are two main calculations that are repeated – that is three (3) Long Hash Calculations (LHCs) and four (4) Short Hash Calculations (SHCs)
- The LHCs are so-called as the hash (unkeyed) output of this calculation is always calculated over the payload of the frame, which is normally between 46 to 1500 bytes
- The SHCs are so-called as the MIC (keyed) output is always calculated over the hash output of the LHCs, which is set at 16 bytes in this description of TSN-MIC but can be larger (recommended) or smaller

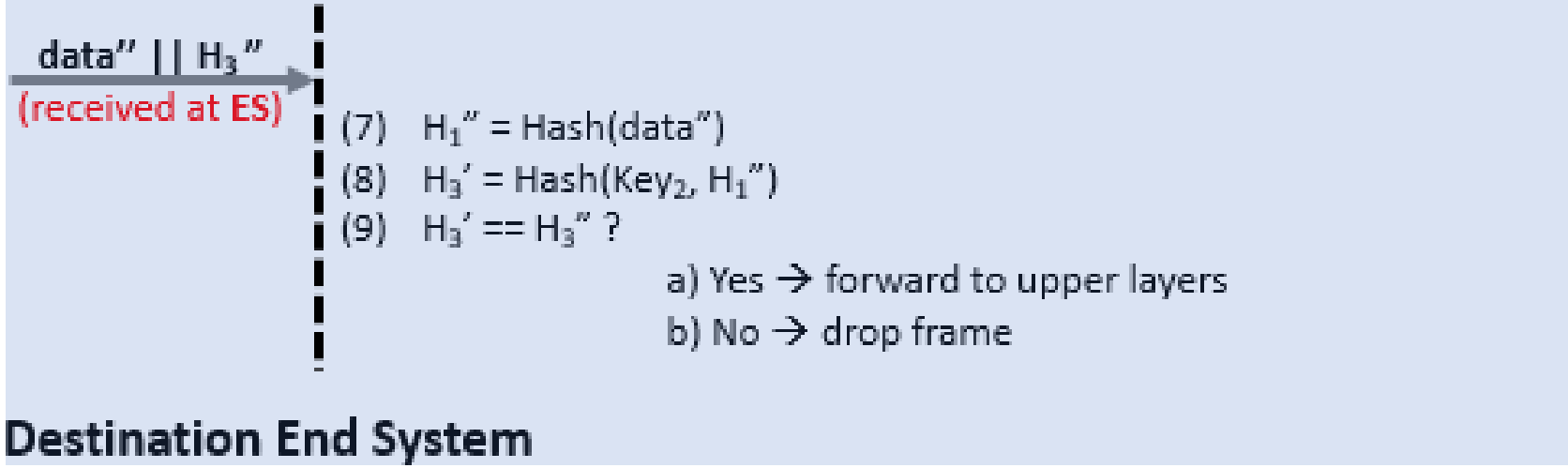
TSN-MIC – Concept [3]



TSN-MIC – Concept [4]



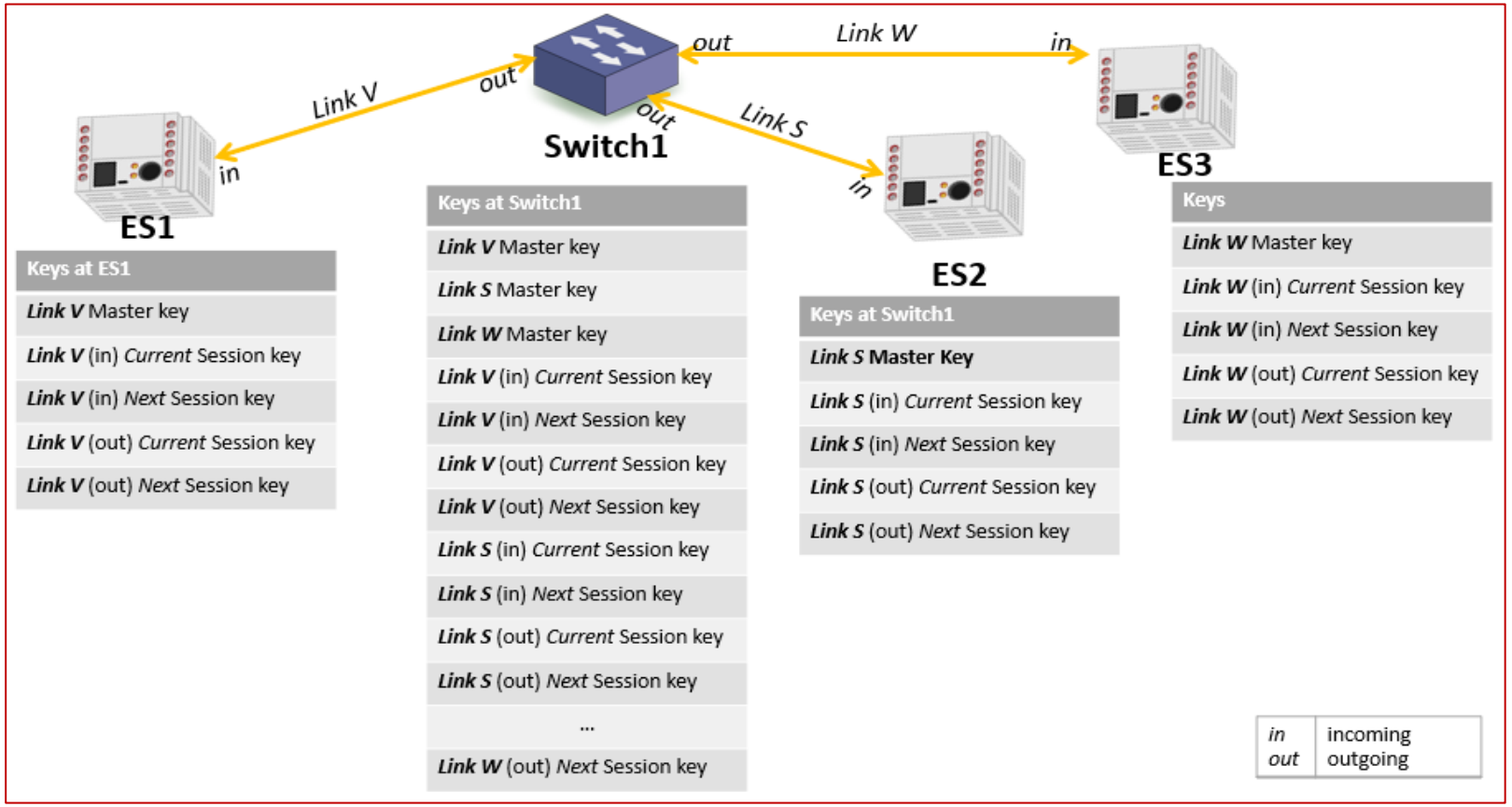
TSN-MIC – Concept [5]



TSN-MIC – Concept [6]

Key distribution

- Two initial session keys: one for incoming messages and one for outgoing messages
- A master key is used for the key establishment procedures to provide confidentiality for the keying material



Security Scheme –TSN-MIC: Implementation and Simulation

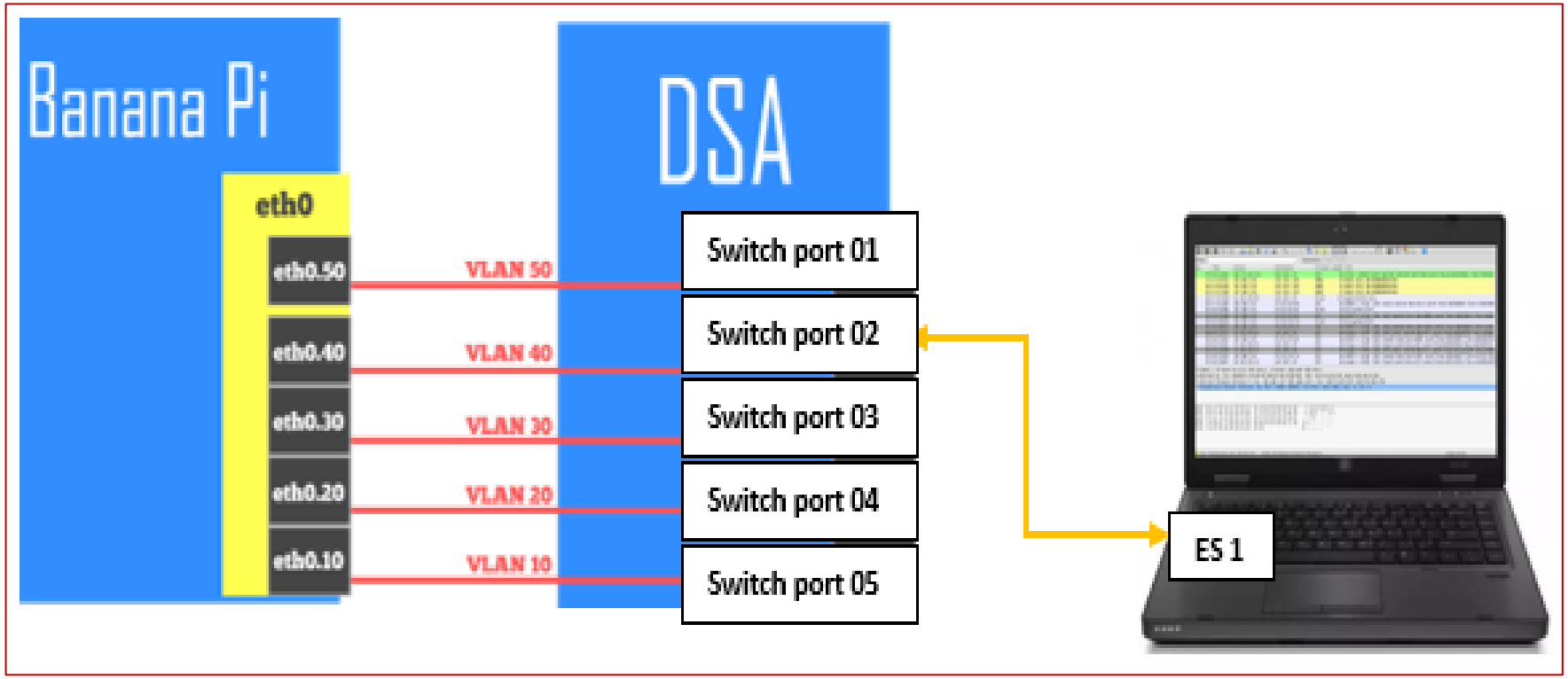
TSN-MIC – Implementation and Simulation [1]

- To test the concept, a software implementation was completed on a Banana Pi, while the efficiency of the TSN-MIC scheme was demonstrated using the OMNeT++ simulator.

- For the former, the Banana Pi was configured as a TSN Switch, modelled after the AFDX specification, and a laptop was then configured as an AFDX End System (ES), serving as both the source ES and the destination ES.

TSN-MIC – Implementation and Simulation [2]

Banana Pi R1 configuration



TSN-MIC – Implementation and Simulation [3]

KE message (1)

```
0000  45 53 31 56 4c 31 45 53  34 56 4c 31 08 00 45 00   ES1VL1ES 4VL1··E·
0010  00 1a 1d 94 00 00 01 11  91 6e c0 a8 b2 5a c0 a8   ····· ·n···Z··
0020  b2 5c 04 16 04 15 00 12  aa ff 01 43 36 30 36 35   ·\····· ···C6065
0030  42 35 46 42 43 36 31 42  31 42 31                  B5FBC61B 1B1
```

Random Number (R_B)

KE message (2)

```
0000  45 53 34 56 4c 31 45 53  31 56 4c 31 08 00 45 00   ES4VL1ES 1VL1··E·
0010  00 1a 1d 94 00 00 01 11  91 6e c0 a8 b2 5c c0 a8   ····· ·n···\··
0020  b2 5a 04 15 04 16 00 12  aa ff 02 38 30 34 37 32   ·Z····· ···80472
0030  34 46 32 37 41 38 43 42  35 33 34 46 45 44 43 42   4F27A8CB 534FEDCB
0040  41 45 44 32 37 37 34 30  36 33 42 41 44 43 36 36   AED27740 63BADC66
0050  30 33 35 36 30 33 35 46  39 32 41 35 42 44 44 36   0356035F 92A5BDD6
0060  37 33 00 43 36 30 36 35  42 35 46 42 43 36 31 42   73·C6065 B5FBC61B
0070  31 42 31                                           1B1
```

Random Number (R_B)

Random Number (R_A)

```
0000  45 53 31 56 4c 31 45 53  34 56 4c 31 08 00 45 00   ES1VL1ES 4VL1··E·
0010  00 1a 1d 94 00 00 01 11  91 6e c0 a8 b2 5a c0 a8   ····· ·n···Z··
0020  b2 5c 07 d0 04 15 00 12  aa ff 0a 44 00 00 6d 00   ·\····· ···D··m·
0030  6d 00 20 61 00 00 00 00  00 00 00 00 68 00 39 00   m· a···· ····h·9·
0040  43 41 00 42 6e 44 00 36  00 5d d7 16 01 cd 8f 8f   CA·BnD·6 ·]·····
0050  db                                                  ·
```

Toggle bit

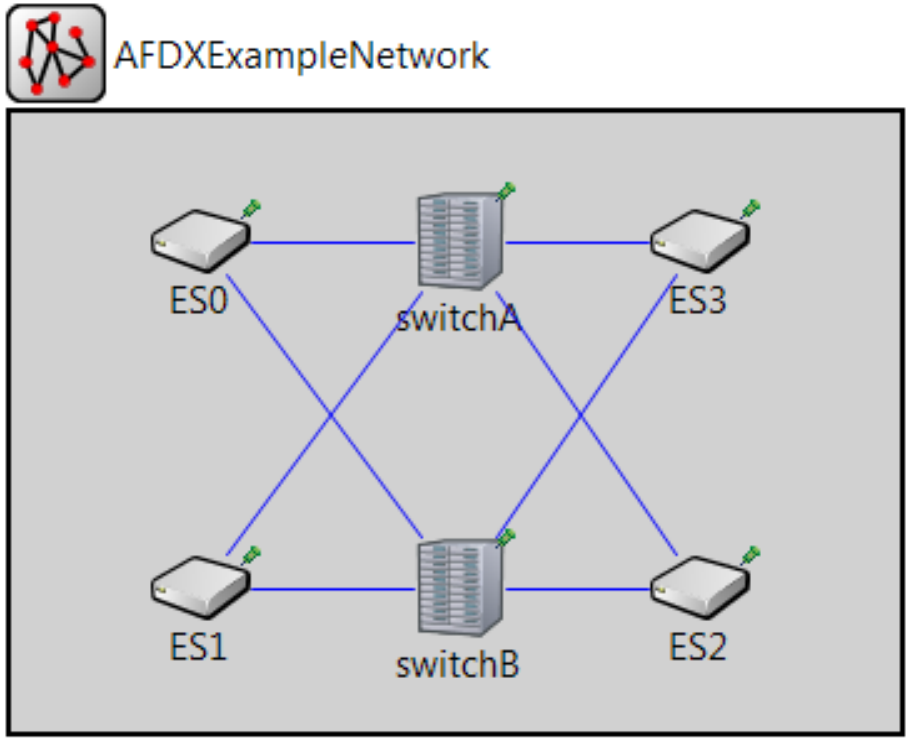
MIC

```
0000  45 53 31 56 4c 31 45 53  34 56 4c 31 08 00 45 00   ES1VL1ES 4VL1··E·
0010  00 1a 1d 94 00 00 01 11  91 6e c0 a8 b2 5a c0 a8   ····· ·n···Z··
0020  b2 5c 07 d0 04 15 00 12  aa ff 0a 44 00 00 6d 00   ·\····· ···D··m·
0030  6d 00 20 61 00 00 00 00  00 00 00 00 68 00 39 00   m· a···· ····h·9·
0040  43 41 00 42 6e 44 00 36  00 1e 5b d0 00 e3 f6 f9   CA·BnD·6 ··[·····
0050  19                                                  ·
```

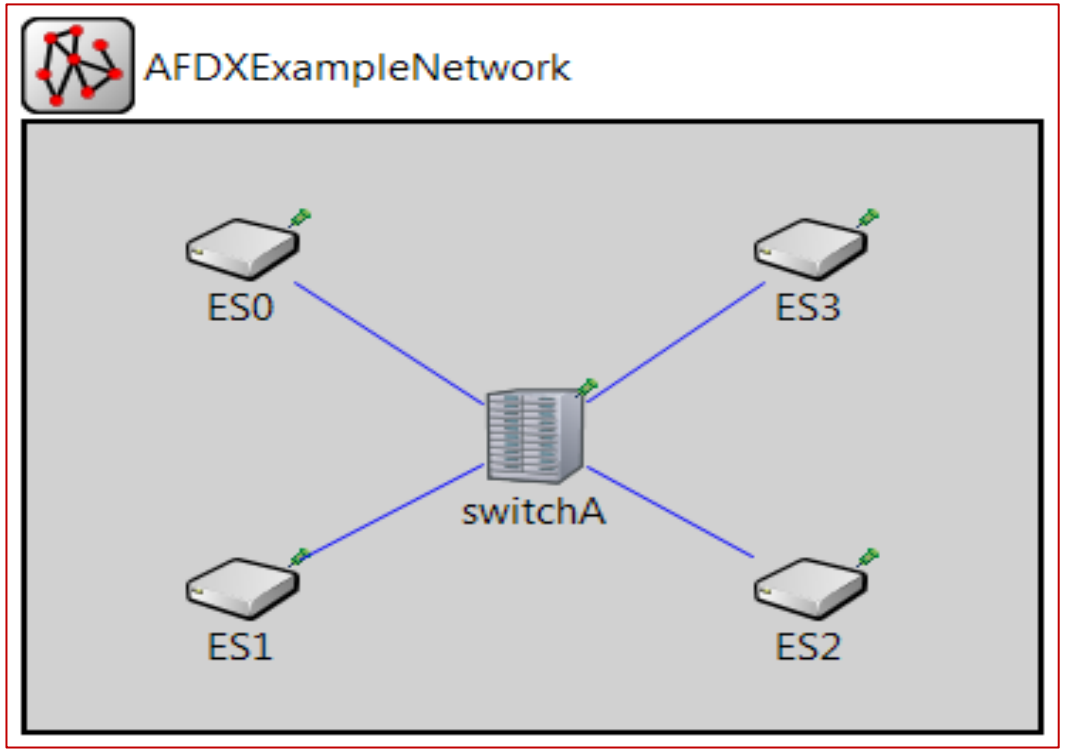
Toggle bit

MIC

TSN-MIC – Implementation and Simulation [4]

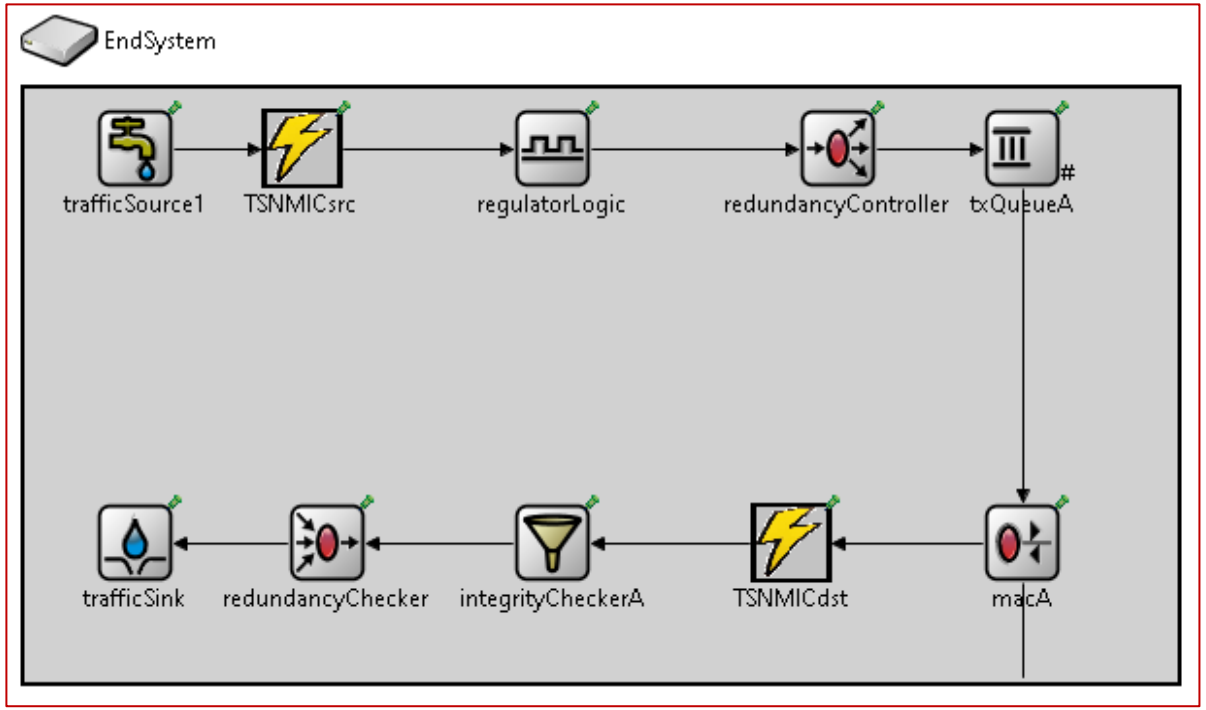


OMNeT++ AFDX model with redundancy

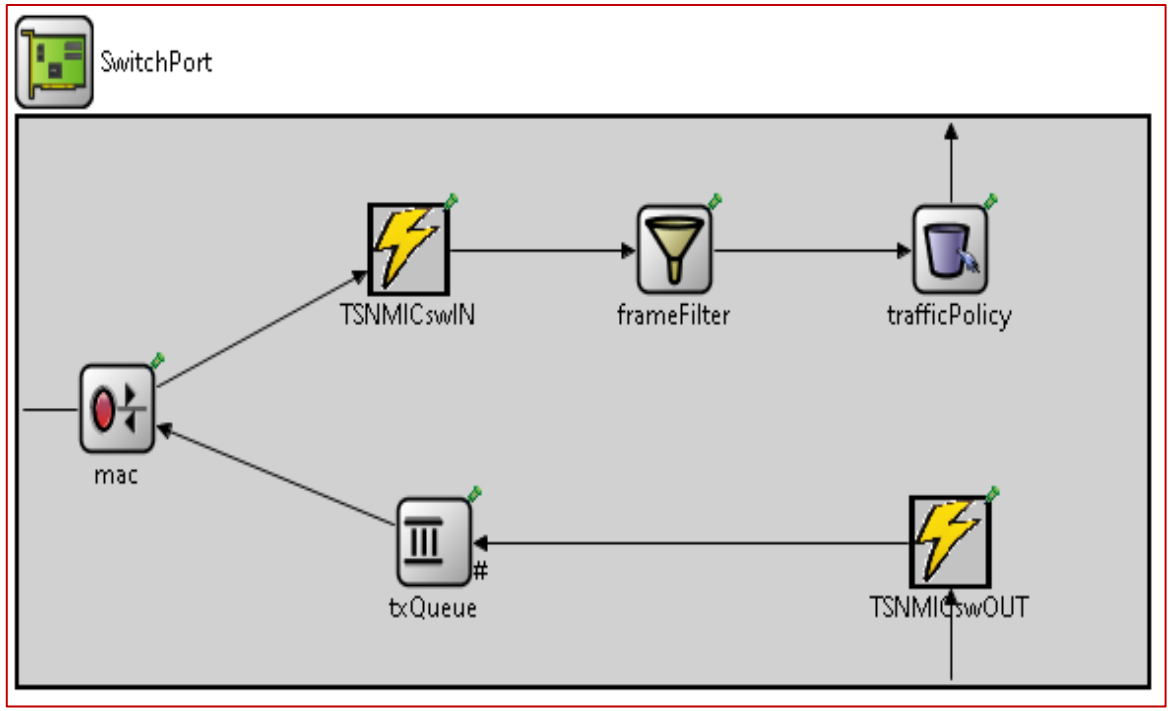


OMNeT++ AFDX model without redundancy

TSN-MIC – Implementation and Simulation [5]



TSN-MIC modules are included for outgoing frames **(TSNMICsrc)** and for incoming frames **(TSNMICdst)**



TSN-MIC modules are included for incoming frames at the TSN Switch ingress port **(TSNMICswIN)** and for outgoing frames at the TSN Switch egress port **(TSNMICswOUT)**

TSN-MIC – Implementation and Simulation [6]

- The TSN-MIC LHC processes frames of between 46 bytes and 1500 bytes, while the TSN-MIC SHC will only ever process 16 bytes (that is, the maximum LHC MIC output)

- Based on published data, Chaskey-12 would process 46 bytes in 5.98 µs to 24.38 µs, and between 195 µs to 795 µs for 1500 bytes

Platform	Algorithm	Speed performance	Processor speed	Performance (bps)
ARM Cortex-M3/M4	Chaskey-8	7.0 cycles/byte	84×10^6 cycles/s	12.0×10^6
ARM Cortex-M3/M4	Chaskey-12	10.5 cycles/byte	84×10^6 cycles/s	8.0×10^6
ARM Cortex-M0	Chaskey-8	16.9 cycles/byte	48×10^6 cycle/s	2.8×10^6
ARM Cortex-M0	Chaskey-12	25.4 cycles/byte	48×10^6 cycle/s	1.9×10^6

TSN-MIC – Implementation and Simulation [7]

- Theoretical calculations indicate that on the ARM Cortex-M3/M4, the expected delay is 8.06 µs to 197.08 µs, while on an ARM Cortext-M0, the TSN-MIC delay is 32.86 µs to 803.48 µs

Platform	Time per byte	TSN-MIC delay for 46-byte frame	TSN-MIC delay for 1500-byte frame
	TSN-MIC Long Hash Calculations		
ARM Cortex-M3/M4	0.13 µs	5.98 µs	195 µs
ARM Cortex-M0	0.53 µs	24.38 µs	795 µs
	TSN-MIC Short Hash Calculations		
ARM Cortex-M3/M4	0.13 µs	2.08 µs	2.08 µs
ARM Cortex-M0	0.53 µs	8.48 µs	8.48 µs

Platform	Chaskey-12 delay	TSN-MIC delay (1 LHC + 1 SHC)	Percentage Change
	Ethernet frame of 46 Bytes		
ARM Cortex-M3/M4	5.98 µs	8.06 µs	+35%
ARM Cortex-M0	24.38 µs	24.38 µs	+35%
	Ethernet frame of 1500 Bytes		
ARM Cortex-M3/M4	195 µs	197.08 µs	+1%
ARM Cortex-M0	795 µs	803.48 µs	+1%

TSN-MIC – Implementation and Simulation [8]

- The efficiency of the TSN-MIC security scheme is then assessed using the OMNeT++ AFDX model
- First, the ***QueryPerformanceFrequency*** function is used to observe the average processing times TSN-MIC LHC and TSN-MIC SHC with the underlying Chaskey-12 algorithm
- The average time take for a TSN-MIC LHC is observed to be 26.6 ms of messages of 128 bytes, and 19.4 ms for a TSN-MIC SHC for messages of 16 bytes

TSN-MIC – Implementation and Simulation [9]

- The OMNeT++ model was then observed to determine the time taken for an end-to-end delivery, which is given as 10.88×10^{-6} simsec (1 simsec ≈ 4.867 seconds) or 53ms in the real world.

- The assumed impact of the TSN-MIC security scheme is given as 157.4 ms (3 x 26.6 ms (LHC) + 4 x 19.4 ms (SHC)) over a simple network.

AFDX Component	**OMNeT++ AFDX** (Actual)	**OMNeT++ AFDX** with **TSN-MIC** (Theoretical)
	Simulation time (simsec)	*Simulation time (simsec)*
Delay at source ES	4.51×10^{-6}	9.50×10^{-3}
Delay at Switch	6.11×10^{-6}	1.49×10^{-2}
Delay at destination ES	2.65×10^{-7}	9.5×10^{-3}
Overall delay	10.88×10^{-6}	3.39×10^{-2}

312%

Results

Results [1]

- The execution of the OMNeT++ TSN-MIC simulation shows that the actual end-to-end delay for a single message is 11.84×10^{-6} simsec, an increase of 8.82% above the baseline of 10.88×10^{-6} simsec.

AFDX Component	Simulation time (simsec)		
	Without TSN-MIC	With TSN-MIC	Percentage change
Delay at source ES	4.51×10^{-6}	4.90×10^{-6}	8.65%
Delay at Switch	6.11×10^{-6}	6.67×10^{-6}	9.17%
Delay at destination ES	2.65×10^{-7}	2.73×10^{-7}	3.02%
Overall delay	10.88×10^{-6}	11.84×10^{-6}	8.82%

Results [2]

- Additional averages were calculated for messages of sizes ranging from 40 bytes to 1500 bytes.
- The overall increase in the simulation time (simsec) is much greater (2,032%) than the change in the TSN-MIC LHC processing time (111%).

Message Size (bytes)	TSN-MIC LHC average processing time (ms)	OMNeT++ processing time (simsec)
40	22.8	3.84×10^{-6}
150	25.4	12.64×10^{-6}
300	29.7	26.64×10^{-6}
450	31.3	36.64×10^{-6}
600	33.4	48.64×10^{-6}
750	36.3	60.64×10^{-6}
900	39.6	72.64×10^{-6}
1150	44.3	92.64×10^{-6}
1500	48.1	119.36×10^{-6}
Percentage increase	+111%	+2,032%

Conclusion

Conclusion

- The TSN-MIC efficiency based on the OMNeT++ shows an increase in the transmission time of 8.82% for each message from source End System to a destination End System, traversing a single TSN Switch.
  - This indicates a 2.52% delay per pair of TSN-MIC calculations (1 LHC and 1 SHC)

- The limitations of the testing environment requires that where more accurate solutions/environments are used, such as with comparable microcontrollers or with an FPGA, an accurate representation of the efficiency TSN-MIC security scheme can be obtained.

- Nevertheless, TSN-MIC demonstrates viability for I4.0/Smart manufacturing and critical infrastructure.

Questions?

Thank you for your attention!