Can SCA 4.1 Replace STRS in Space Applications?

Ran Cheng, Li Zhou, Qi Tang, Dongtang Ma, Haitao Zhao, Shan Wang and Jibo Wei

National University of Defense Technology

May 17, 2017

Outline

SDR Implementation in Space Environment

Expectations

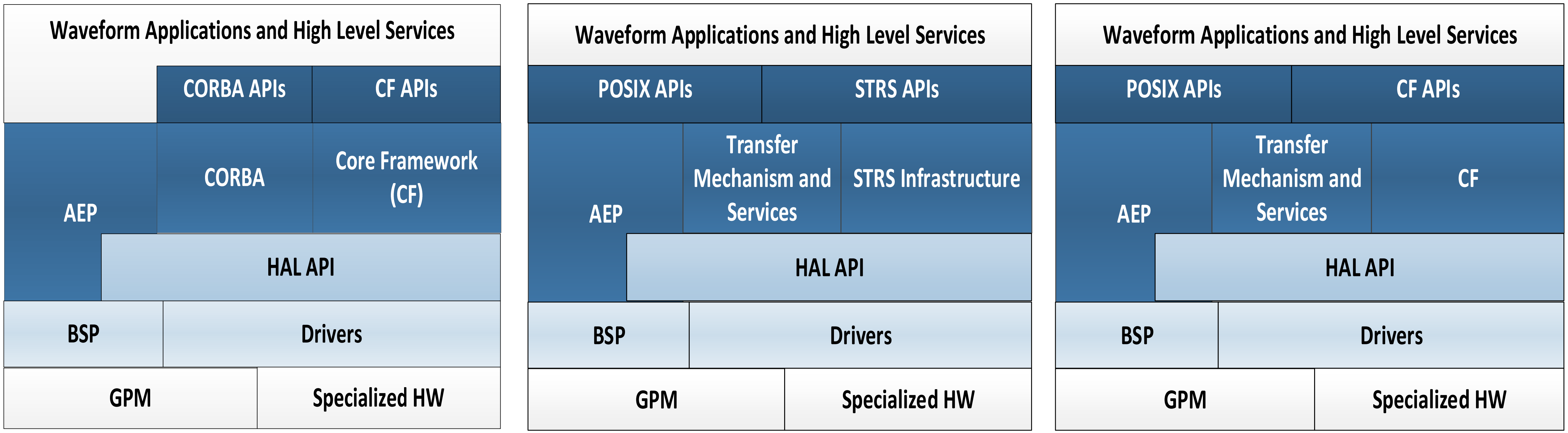
- Reduce the development cycle-time
- Reduce the development cost
- Increase the communication flexibility among space radios and ground stations

Challenges

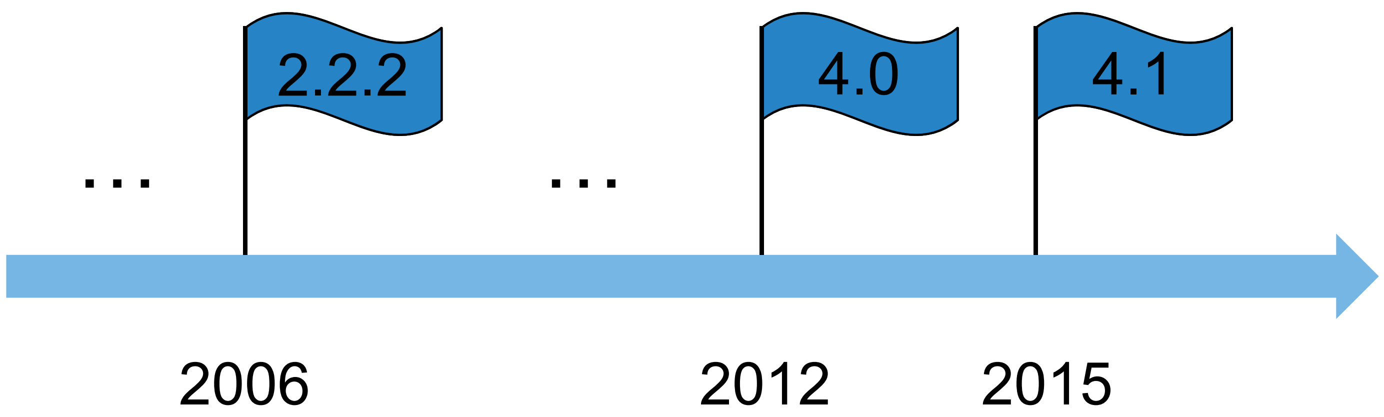
- Strict SWaP constrained requirement
- Low capability of chips for space use
- Reuse of hardware and software

Software Execution Model Comparison

SCA 2.2.2 | STRS | SCA 4.1



SCA Development and Evolution



The year of announcement of several SCA versions

SCA Evolution from 2.2.2 to 4.1

- Adopt “push model” behavior
- Remove dependence on CORBA
- Add static registration behavior
- Provide Units of Functionality (UOF) and SCA Profiles

Layered Framework Comparison

SCA 4.1 layered framework

Core Module

- **File Manager**
  - File Operation /Management
  - File System Operation /Management
  - File Manager Operation
- **Domain Manager**
  - Waveform Install/Delete
  - Waveform Application Manager
  - Waveform Control Manager
  - Domain Configuration Manager
- **Device Service Manager**
  - Device Login/Logout
  - Device Management Control
  - Service Login/Logout
  - File System Management

Basic Module

- Domain Description Parser Module
- Component Connection Management module
- Component Deployment Management Module
- Waveform Reconstitution Module
- Resource Management Module
- ORB Encapsulation Module
- Waveform Development Public Library

STRS layered framework

Core Module

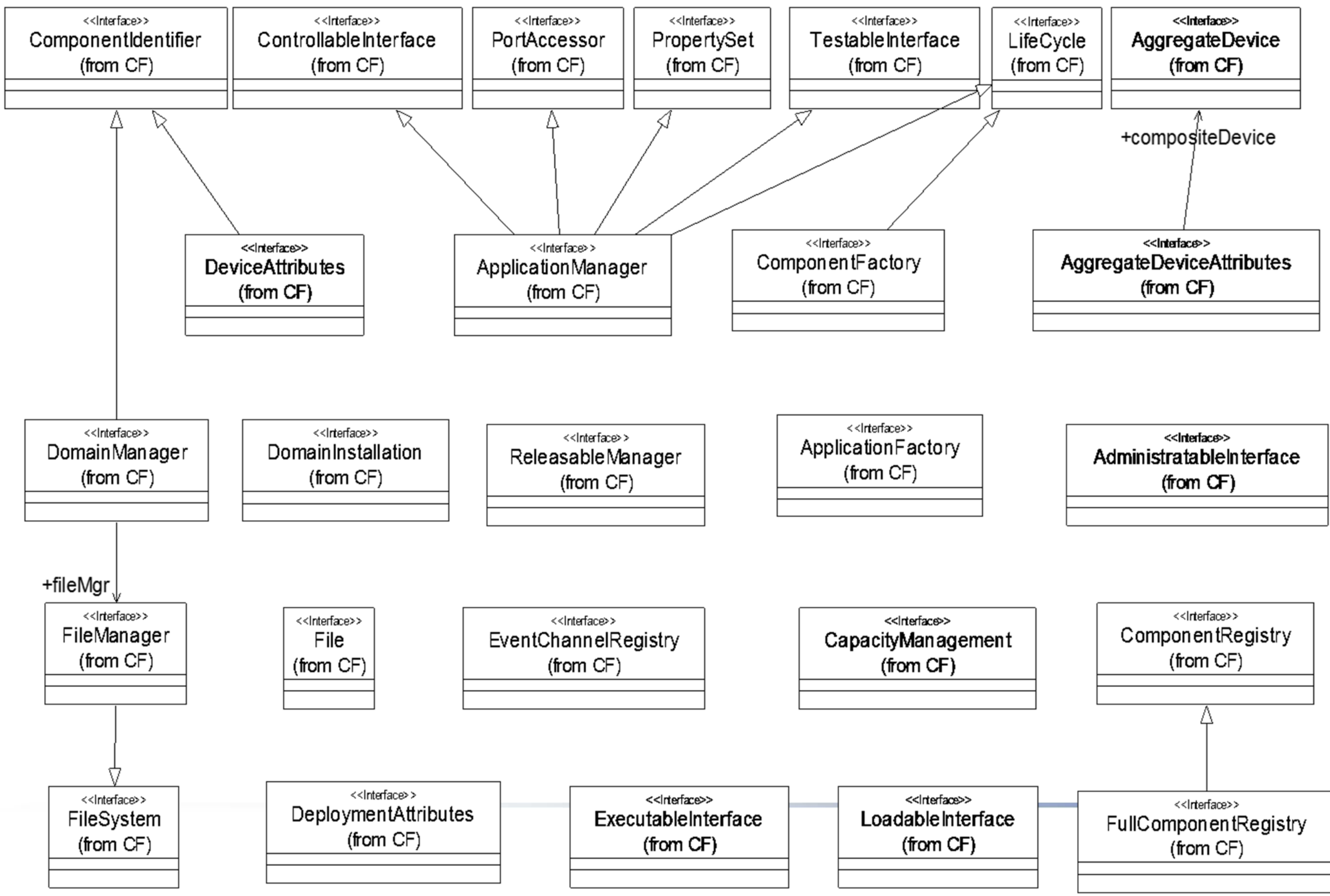
- **File Manager**
  - File Maintenance Management
  - File Operation Interface
- **Domain Manager**
  - Waveform Install/Delete
  - Waveform Application Manager
  - Waveform Control Manager
- **Device Service Manager**
  - Device Login/Logout
  - Device Management Control
  - Service Login/Logout

Basic Module

- Configuration Parser Module
- Component Connection Management module
- Component Deployment Module
- Command Control Management Module
- Debug Information Log Module
- Message-oriented Middleware
- Waveform Development Public Library

Core Framework Interfaces Comparison

SCA 4.1 core framework interfaces



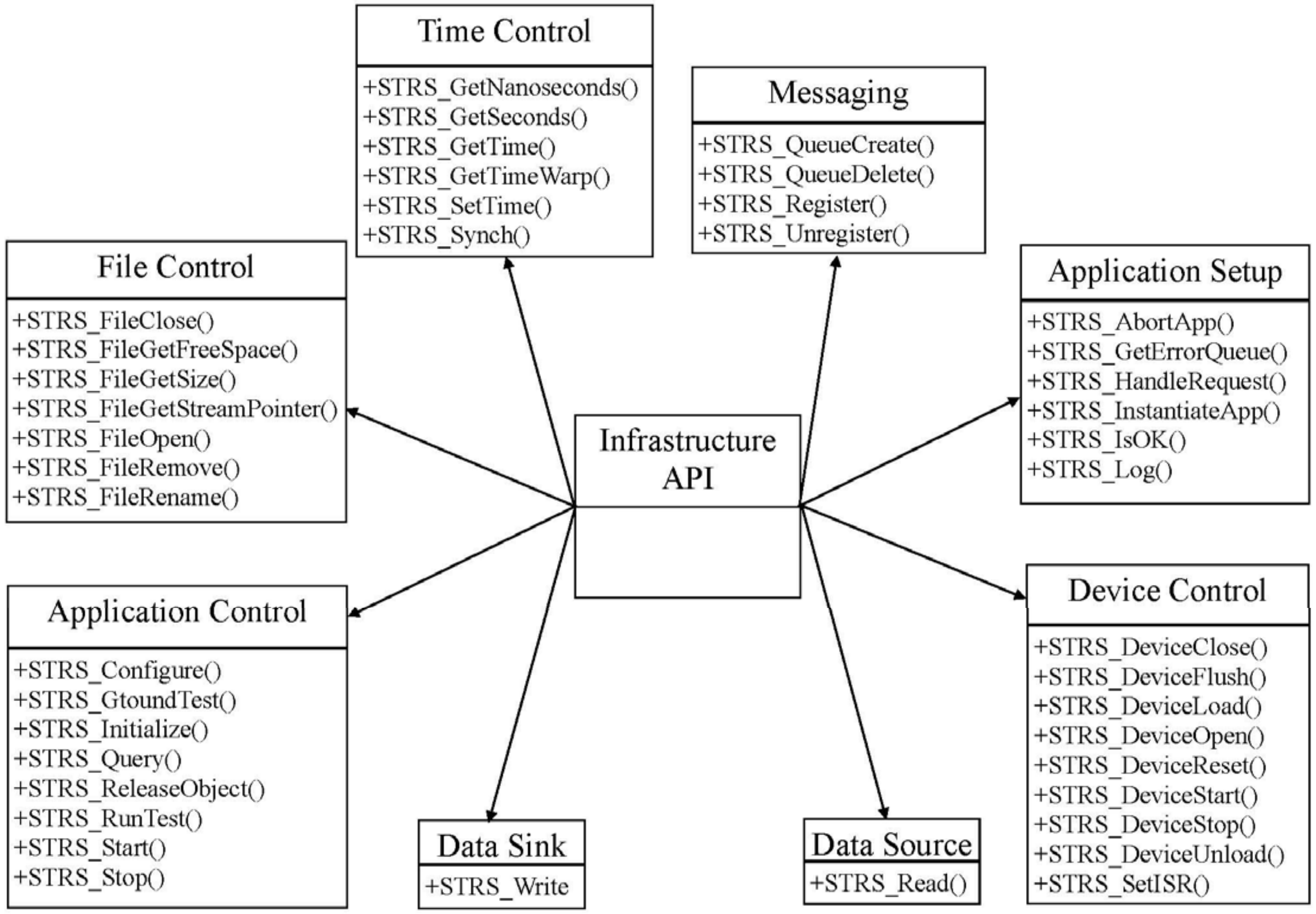
SCA 4.1 core framework written in C++ has complex inheriting relationship

Infrastructure API

- Basic Application Interfaces
  - ComponentIdentifier
  - LifeCycle, etc.
- Basic Device Interfaces
  - -AggregateDevice
  - -CapacityManagement, etc.
- Framework Control Interfaces
  - -ApplicationManager
  - -DeploymentAttributes, etc.
- Framework Service Interfaces
  - -ComponentFactory
  - -FileManager, etc.

Core Framework Interfaces Comparison

STRS core framework interfaces

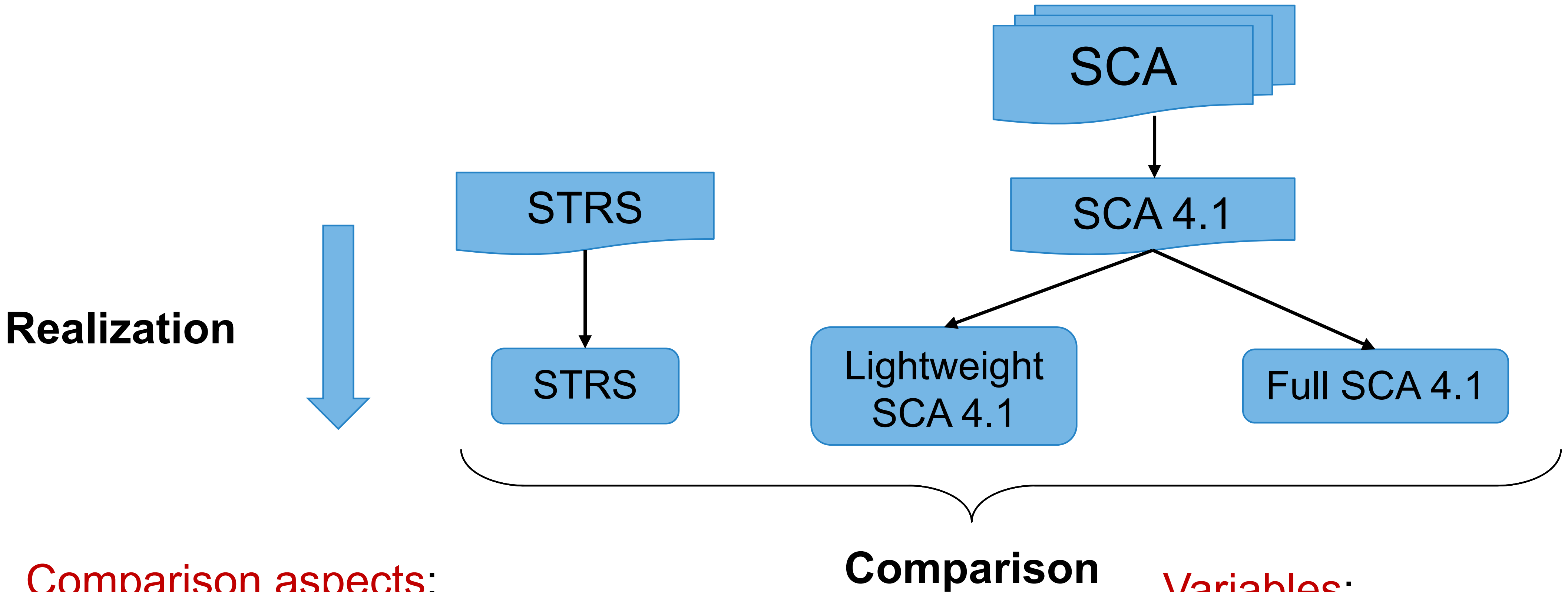


STRS core framework written in C simplifies the implementation

Infrastructure API

- Time Control
- File Control
- Application Control
- Messaging
- Application Setup
- Device Control
- Data Sink
- Data Source

Comparison Aspects and Variables



Comparison aspects:

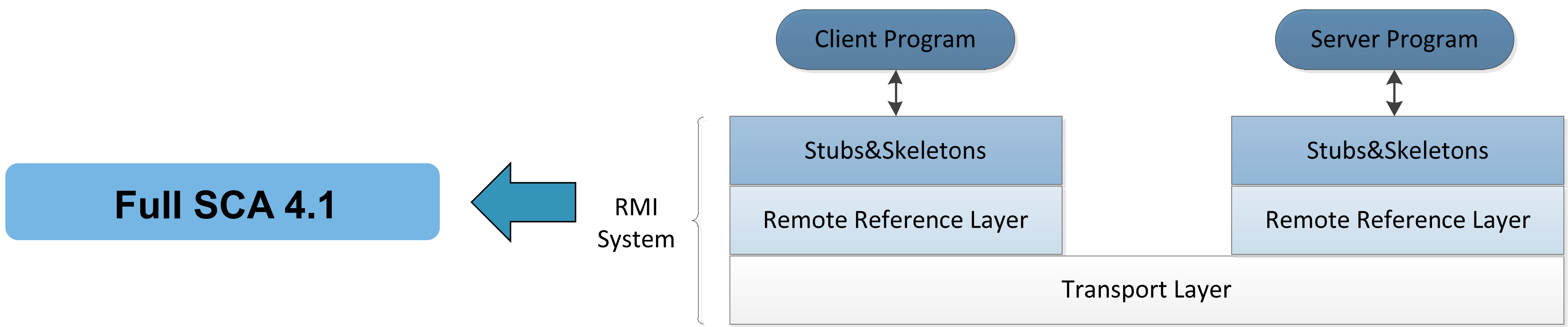
- the static memory occupation
- inter-components communication delay
- waveform deployment delay
- waveform switching delay

Variables:

- the packet size
- the total amount of packets
- the number of components

Agency Mechanism

- The stub and skeleton combine to form the RMI frame protocol.
- The remote reference layer is adopted to find the communication object.
- The transport layer provides the interconnection of client and server based on the TCP/IP protocol.



RMI: Remote Method Invocation

Testbed Introduction

Testbed: ZLSDR-1000

Main chip: ZYNQ 7030

- a dual-core of ARM Cortex-A9 (clock speed 667MHz)
- a FPGA of Kintex-7 (logic cells 125K, DSP Slices 400)

DDR memory size: 1GB

Operating system: Linux 3.17

General SDR platform (ZLSDR-1000)

Calculation of Communication Delay

The inter-components communication time of **a single link** T = T2 - T1

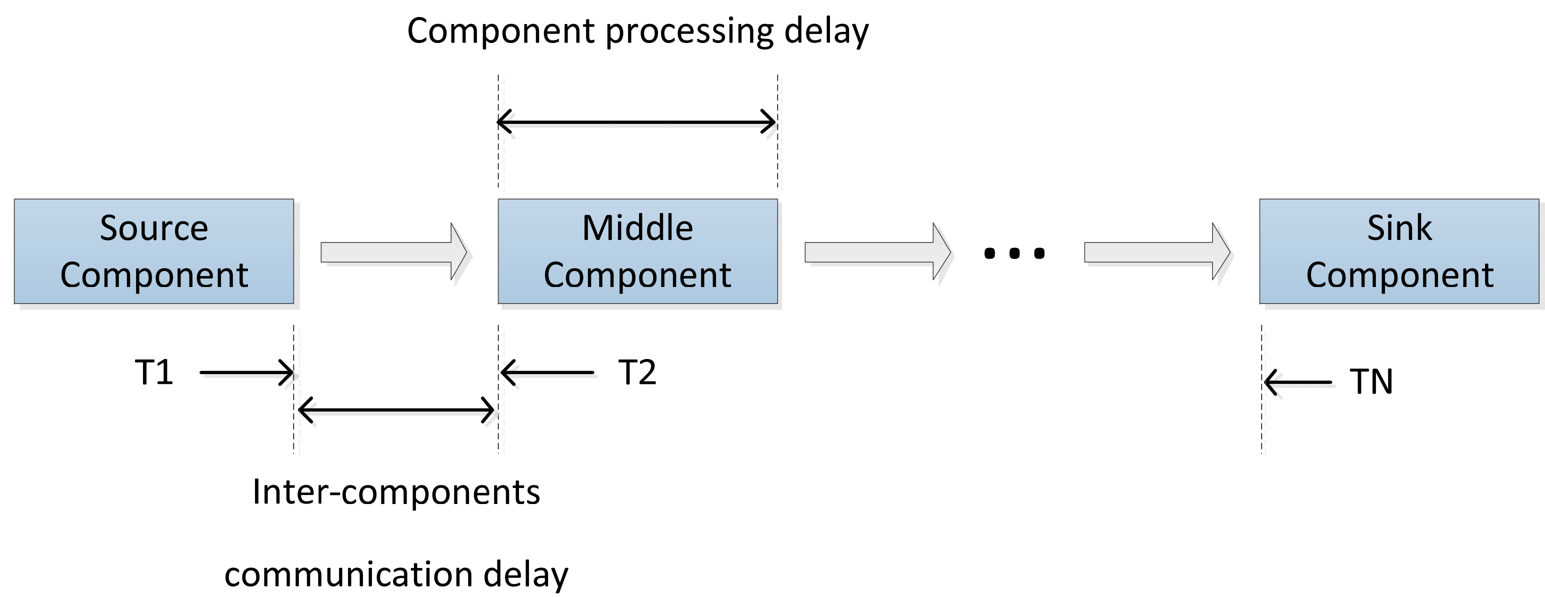
component processing time

↓

zero

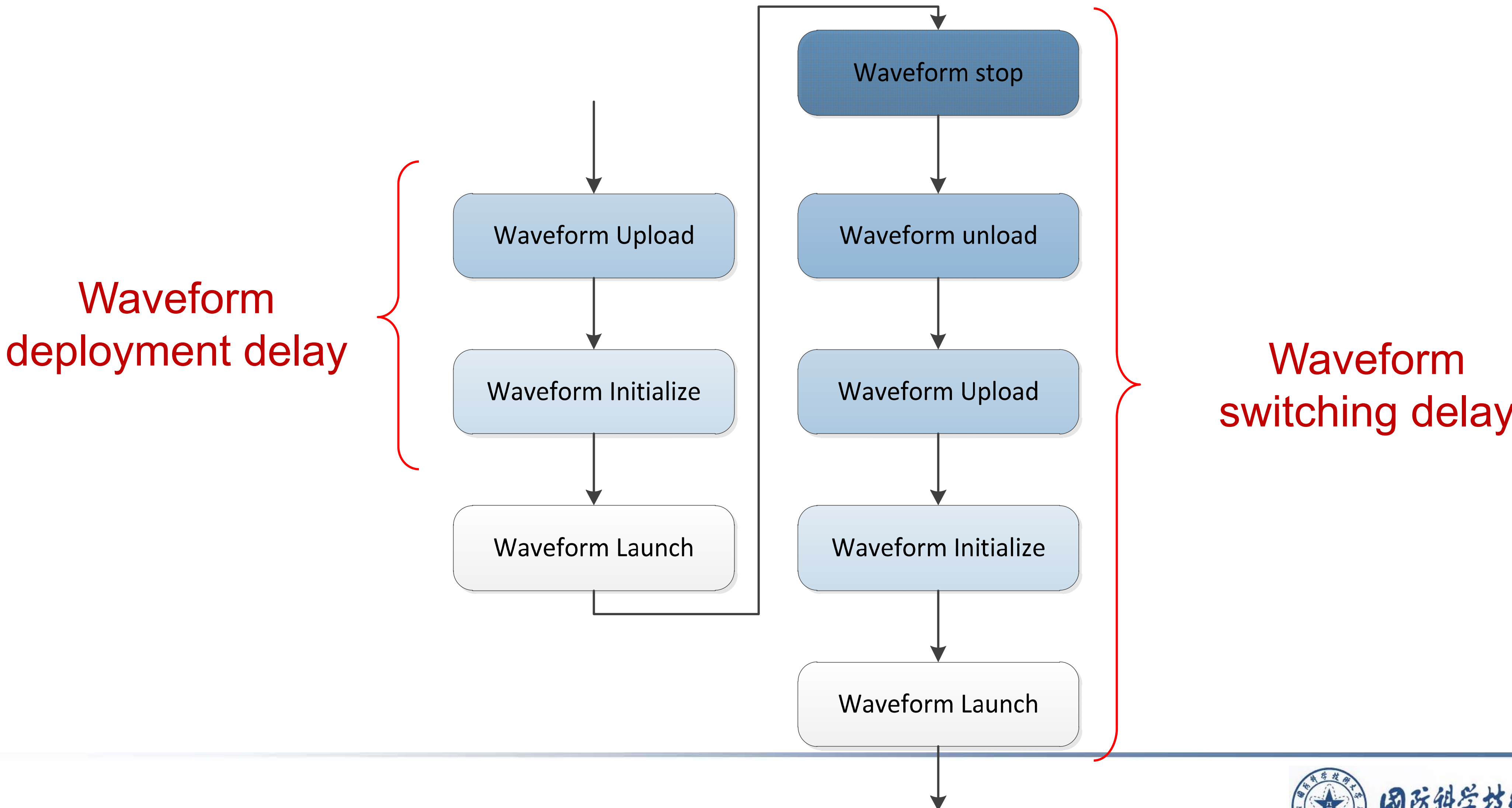
The **amount time of all links** in inter-components communication T = TN - T1

Inter-components communication delay



Waveform Deployment & Switching Delay

Waveform deployment delay and waveform switching delay



Experiment Results

Experiment parameters

Parameter	Value
Number of components	3, 4, …, 11
Packet size (bytes)	128, 256, 512, 1024, 2048, 4096
Amount of packets (10^6)	1, 2, …, 10

Experiment Results

Static memory occupation comparison

Implementation	Static memory occupation (MB)
STRS	4.77
Lightweight SCA 4.1	27.82
Full SCA 4.1	77.20

Experiment Results

Inter-components communication delay comparison with different packet size

Package size (bytes)	STRS (us)	Lightweight SCA (us)	Full SCA 4.1 (us)
128	17.30	17.20	189.00
256	17.30	18.40	189.00
512	17.30	19.00	196.00
1024	17.30	20.80	200.00
2048	17.30	21.00	216.00
4096	17.50	21.00	240.00

Experiment Results

Total time consumption comparison with different amount of packets

Amount of packets	STRS (s)	Lightweight SCA 4.1 (s)	Full SCA 4.1 (s)
100,000	17.50	21.00	240.00
200,000	40.10	42.20	476.00
300,000	66.70	63.60	715.00
400,000	81.60	85.00	946.00
500,000	101.90	105.40	1190.00
600,000	119.20	125.40	1441.00
700,000	143.10	147.80	1669.00
800,000	164.90	162.20	1905.00
900,000	184.10	191.00	2142.00
1,000,000	201.70	207.40	2399.00

Experiment Results

Inter-components communication time comparison with different numbers of components

Number of components	STRS (us)	Lightweight SCA 4.1 (us)	Full SCA 4.1 (us)
3	15.50	23.00	175.00
4	16.00	19.67	186.67
5	16.75	20.50	197.50
6	16.00	19.80	208.00
7	16.50	21.33	211.67
8	16.43	20.14	218.57
9	17.00	20.75	227.50
10	17.22	20.22	233.33
11	17.50	21.00	240.00

Experiment Results

Waveform deployment delay comparison with different numbers of components

Number of components	STRS (ms)	Lightweight SCA 4.1 (ms)	Full SCA 4.1 (ms)
3	12	46.7	843.3
4	13.3	51	1078
5	14.7	55	1318
6	16	58.7	1588
7	17.7	62	1841.8
8	18.7	65.3	2130
9	20	69.7	2385
10	20.5	73	2641.6
11	21	79.6	2903.7

Experiment Results

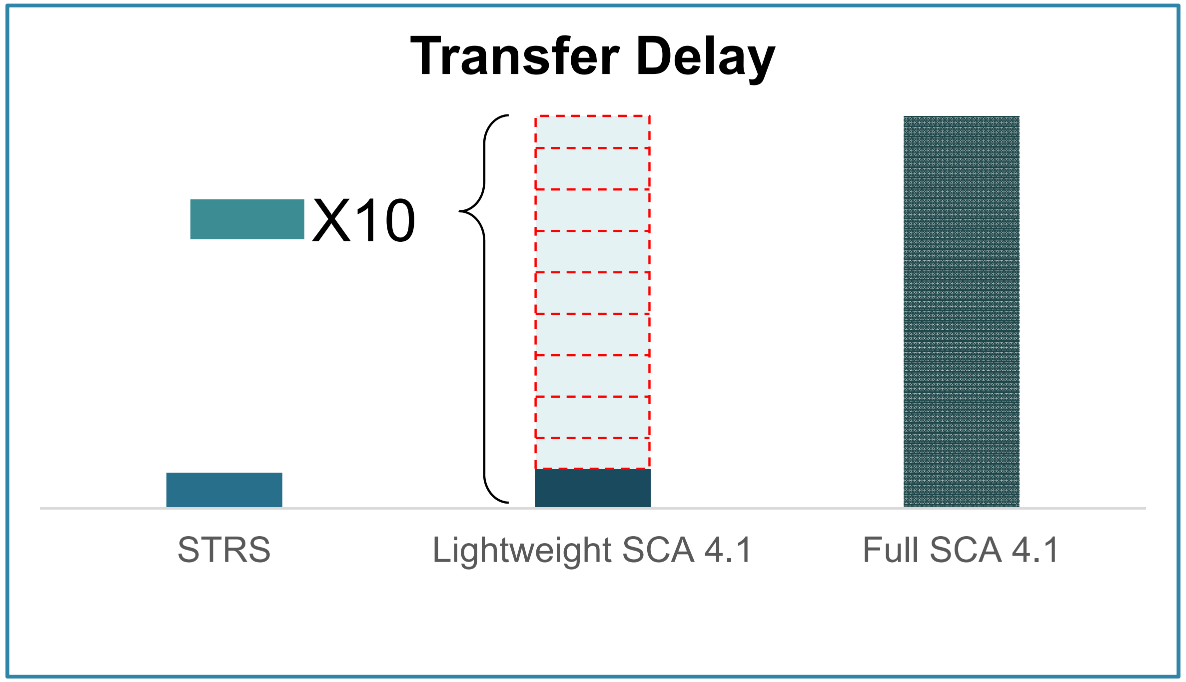
Waveform switching delay comparison with different numbers of components

Number of components	STRS (ms)	Lightweight SCA 4.1 (ms)	Full SCA 4.1 (ms)
3	13	51.6	880.6
4	14.6	56	1124
5	15.7	60	1376.3
6	17.7	64.4	1661.6
7	19.4	67.7	1930.4
8	20.4	71.3	2234
9	21.6	76.7	2502
10	22.2	79	2769.6
11	23	85.9	3035.1

Experiment Analysis

Lightweight SCA 4.1 and STRS have very close transfer efficiency.

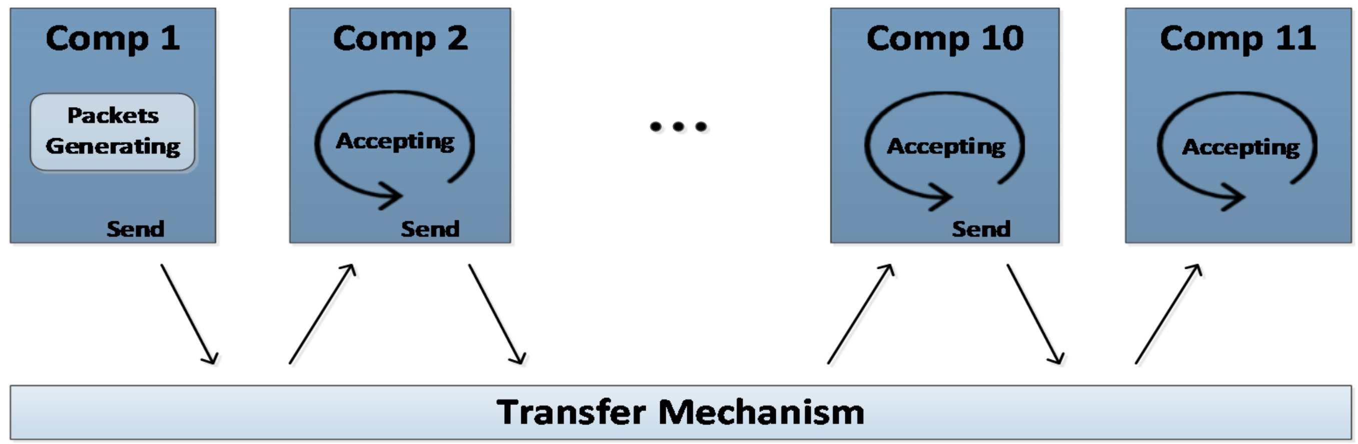
Their transfer delay is almost 1/10 of full SCA 4.1.



Experiment Analysis

Reasons of low efficiency in full SCA 4.1:

- the creation and communication between processes
- the adoption of object request broker (ORB) mode



Experiment Analysis

STRS Advantages:

- high efficiency
- the ability to suit the resource-limited lightweight platform

SCA 4.1 Advantages:

- high flexibility
- providing different SCA Profiles
- reducing the expenditure with thread communication

STRS Disadvantages:

- low portability and interoperability
- difficulty for the application developer

SCA 4.1 Disadvantages:

- high waveform deployment delay
- high waveform switching delay
- large static memory occupation

Conclusion

Recommendation:

- Lightweight SCA 4.1 is worth considering for space radios owing to its high efficiency.
- Users should pay attention to waveform deployment, switching delay and static memory occupation.

Future work:

- Carrying out experiments in platform with limited computing capability and memory size.
- Evaluate the power consumption of different architectures.

Thank you!

Q&A