

Protocol Validation System for On-Board Communications



PVS Phase 1 Final Presentation

*SpaceWire WG #14, 22nd February 2010
Noordwijk, Netherlands*



PVS Rationale



- Current **evolution of satellite on-board communications**, require the development & experimentation with new dedicated communication protocols and services (SpW, SOIS, etc.)
- **New generation of validation tools is required** to support advanced protocol development, test, integration & validation



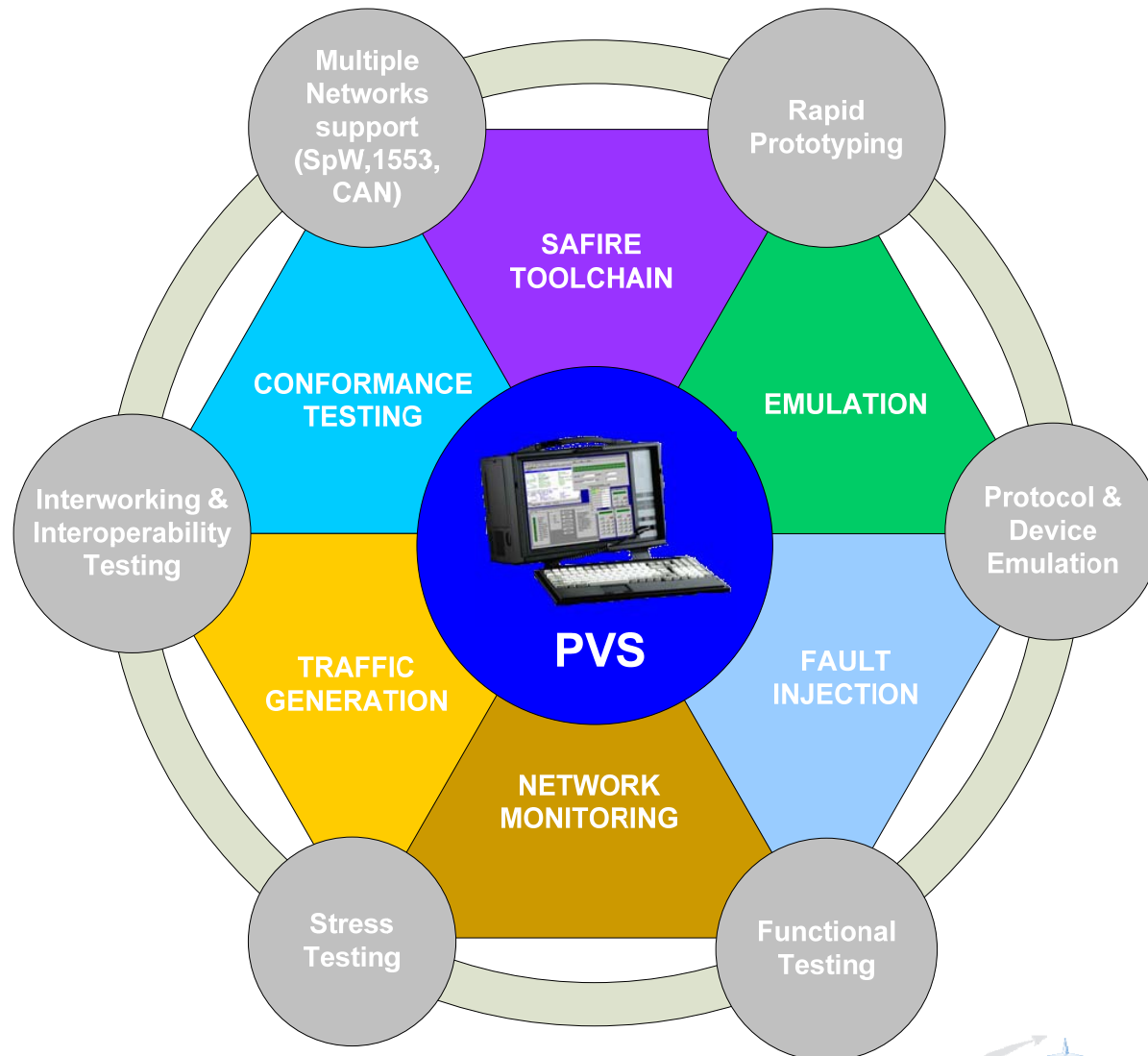
- A **protocol validation tool** with more than 20 years of experience in the **telecommunication sector**. & with hundreds of installations worldwide
- Has been widely used for testing various telecommunication networks (ISDN, V5, SS7, IN, GSM, UMTS, VoIP, custom)



The basic motivation is to provide an open, generic and fully integrated protocol validation system (PVS) for satellite on-board communications supporting multiple physical interfaces (SpW, MIL-STD-1553) and functionalities (emulation, validation, interworking testing, monitoring).



PVS at a glance



PVS foreseen features

- **DEVICE EMULATION:** economic & portable replacement of a network element in the testbed
- **PROTOCOL EMULATION:** experimentation with various protocol features (parameterization of protocol variables, exclusion/inclusion of protocol optional functions, combination of multiple protocols)
- **CONFORMANCE TESTING:** execution of tests to ensure that a device (System Under Test) is operating in compliance with the applicable ECSS and CCSDS standards.
- **FAULT-INJECTION:** injection of errors at various protocol layers to validate the response of the devices/networks in erroneous conditions
- **TRAFFIC GENERATION:** generation of traffic for validation of higher layer protocols or bulk traffic injection at lower layers for performance evaluation and network dimensioning
- **NETWORK MONITORING:** network monitoring, through direct physical traffic acquisition (network statistics, error detection, troubleshooting)

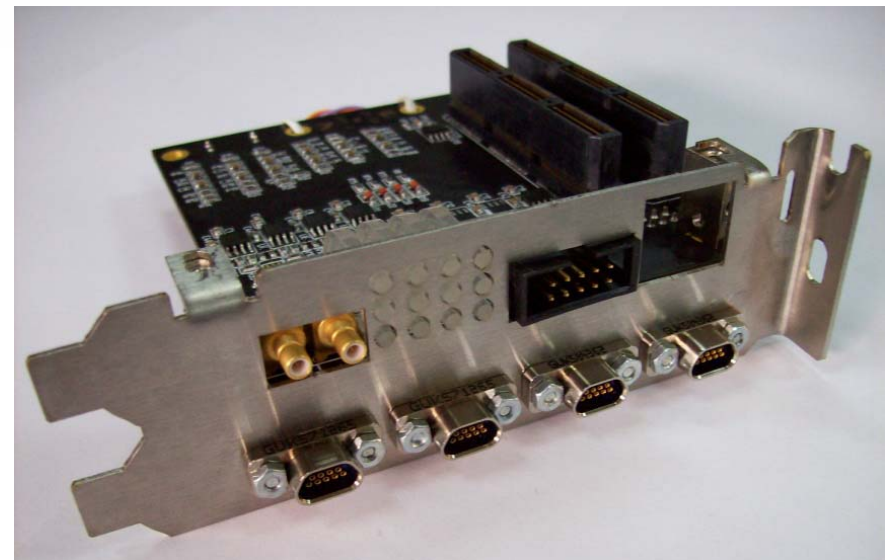
Current contract technical objectives

- *PVS Phase 1 (Feb 2009 – Jan 2010):*
 - Requirements capturing & analysis, based on requirements by ESA and EADS Astrium, and top level partitioning
 - Technology review on related technologies, tools and protocols
 - Identification of SpW-T features to validate
 - Realisation of a PVS proof-of-concept prototype for SpW networks
 - Evaluation and demonstration of the PVS with SpW-T and GAMMA protocols
 - Development plan definition for the full PVS

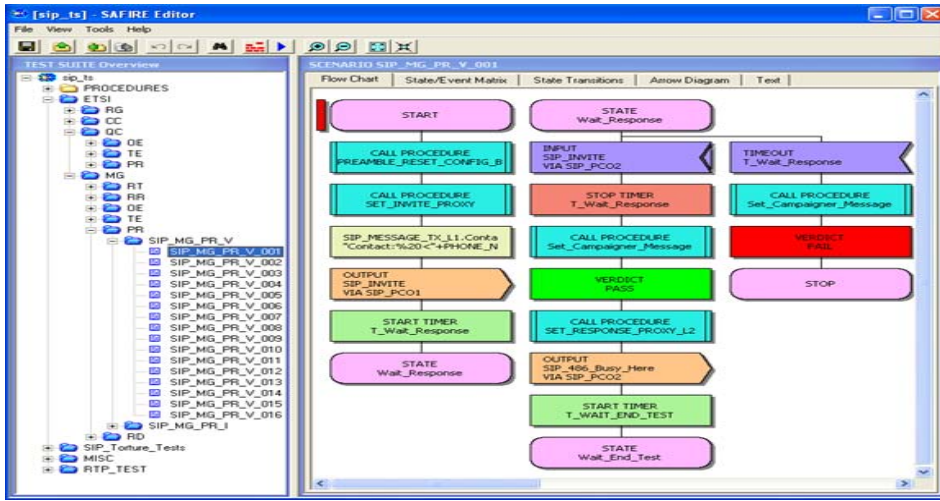
Results: Hardware platform



- 4 SpW ports
- FPGA protection
- Fine (KHz) Tx clock granularity
- trigger I/F
- > 300 Mbps SpW Line Rate



Results: Integration with SAFIRE graphical tool chain



The screenshot shows the SAFIRE Campaigner interface. It displays a 'Log List' with the following entries:

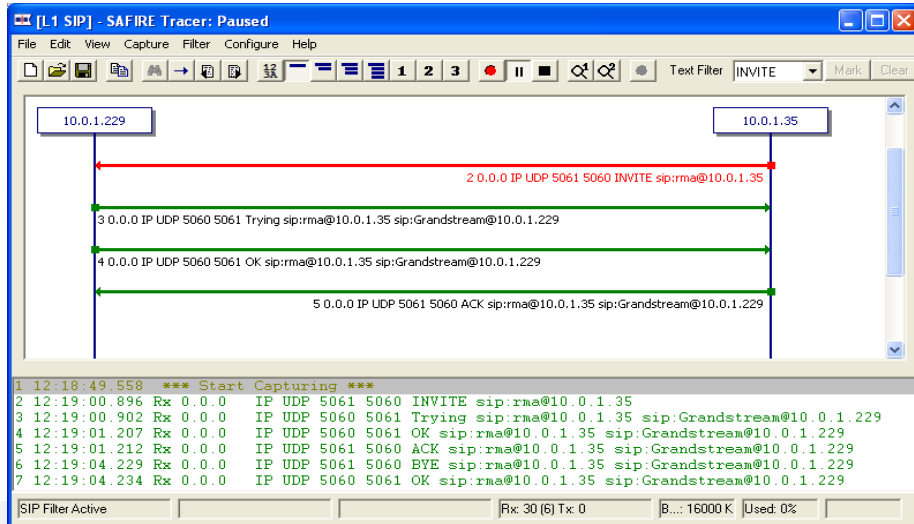
```

13:50:49.866: Start campaign
13:50:49.866: Not using test suite configuration data
13:50:50.069: Load Act_Master
    
```

Summary statistics: Current: 4, Total: 4, Pass: 4, Fail: 0, Inconc: 0, Stopped: 0.

The 'Result List' table shows the following data:

Test case	Group	Verdict	Reason	Start Time	Stop Time	Duration
Act_Master		PASS		13:50:50.069	13:50:50.083	13 ms
Act_Slave		PASS		13:50:50.388	13:50:50.399	10 ms
Deact_Slave		PASS		13:50:50.695	13:50:50.708	12 ms
Data_M2S		PASS		13:50:51.013	13:50:51.022	9 ms



The screenshot shows the SAFIRE Tracer interface displaying a CDR SIP call record. The call data is as follows:

```

Call ID 2
Calling rma
Called rma
Release Cause Normal Call Clearing
Time Info
Start 04.10.2007 12:19:00
Stop 04.10.2007 12:19:00
Total 3.338
Setup 3.333
Post Dial Delay 0.311
Connected 0.005
Tear Down 0.000
Extra Info
SIP
Call ID 176e3a8d-cdf9-1810-8209-0013e8803
From URI sip:rma@10.0.1.35
To URI sip:Grandstream@10.0.1.229
Media Payload Types audio
Silence Suppression 'G.711 uLaw'
Stream Info
Calling To Called Payload 'G.711 uLaw'
Called To Calling Payload '472E37313120754C6177'H
SWIQT Jitter 992
Lost Packets 4076
Burst Lost Packets 0
Out Of Sequence Packets 0
Called To Calling Payload 'G.711 uLaw'
SWIQT Jitter 994
Lost Packets 213
Burst Lost Packets 0
Out Of Sequence Packets 0
    
```

Results: Validation of SpW-T and GAMMA protocols

The screenshot shows the SAFIRE SpW-T Protocol Validation System interface. The window title is "SAFIRE SpW-T Protocol Validation System". The menu bar includes "File", "View", "Run", and "Help". The toolbar contains icons for statistics, RX/TX control, and link speed configuration.

The main interface is divided into three sections: General, Receiver, and Transmitter. The General section contains a table with columns: Name, ID, Statistics, Port Mode, and Configure. The Receiver section contains a table with columns: Status, Rx Frames, and Start/Stop. The Transmitter section contains a table with columns: Status, Traffic File, and Tx Frames.

The table below shows the configuration for Station 1, Board 1, SpW-T. The link speed is set to 100 Mbps and there are 2 ports.

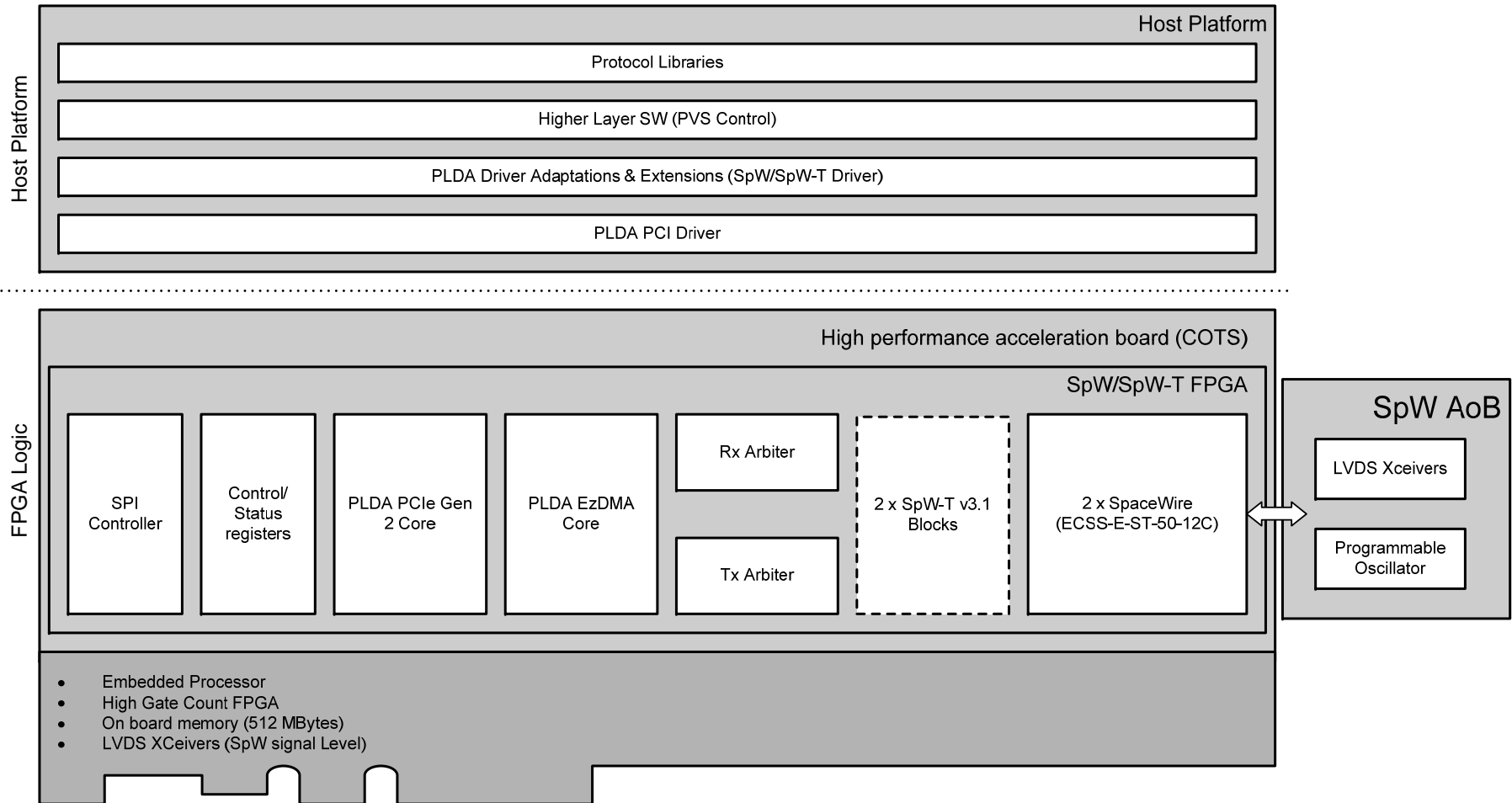
Port	ID	Statistics	Port Mode	Configure	Start/Stop	Status	Rx Frames	Start/Stop	Status	Traffic File	Tx Frames
Port 1	1	<input type="checkbox"/>	Single	P1_Sched	<input type="checkbox"/>	Stopped	0	<input type="checkbox"/>	Stopped	P1256 512 1024	0
Port 2	2	<input type="checkbox"/>	Single	P2_Sched	<input type="checkbox"/>	Stopped	0	<input type="checkbox"/>	Stopped	p2 tx 256 512 10	0

Annotations in the image point to various features:

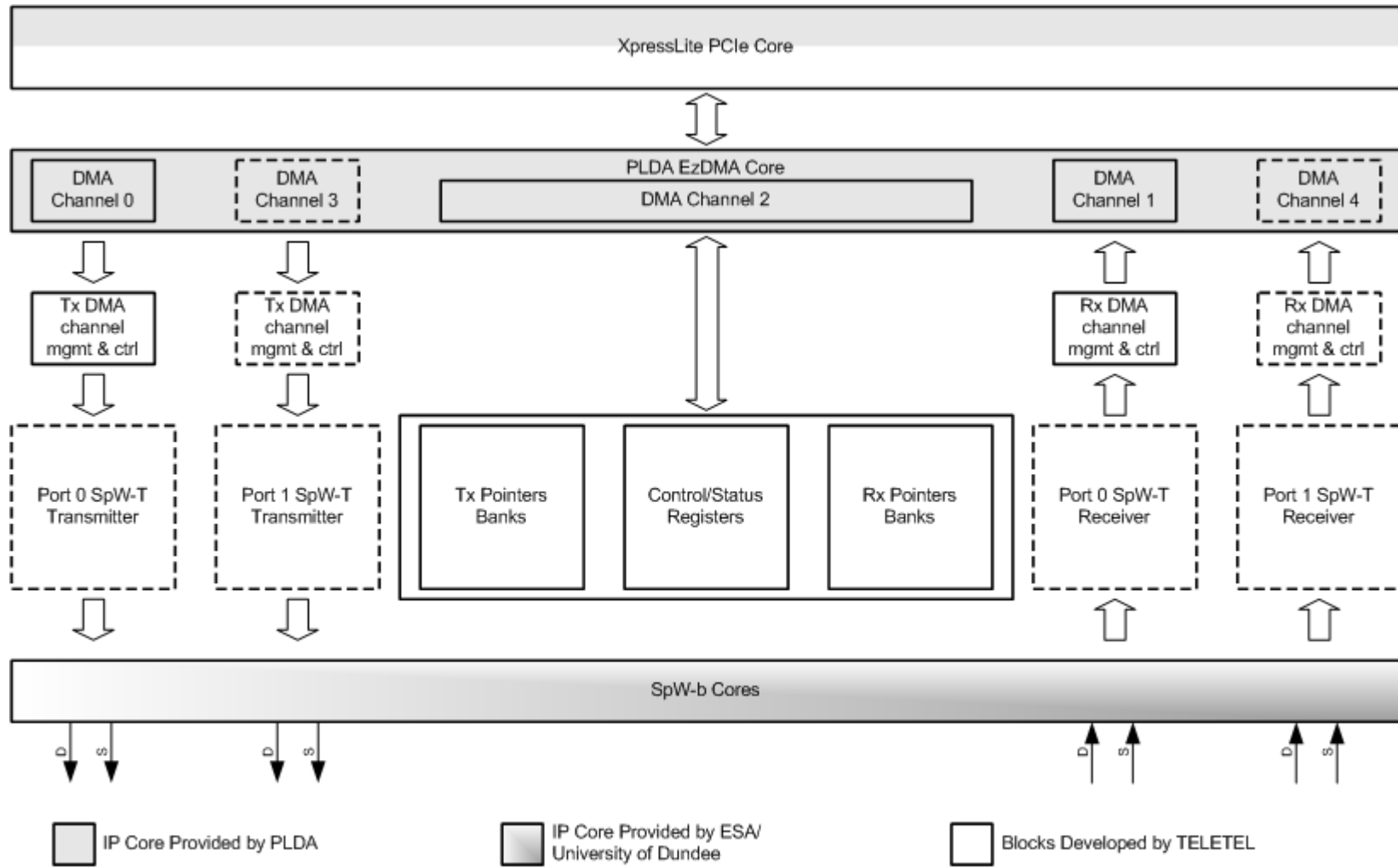
- View the selected statistics (points to the Statistics icon in the toolbar)
- Start/Stop selected RX (points to the RX control icon in the toolbar)
- Start/Stop selected TX (points to the TX control icon in the toolbar)
- Start/Stop selected RX/TX simultaneously (points to the combined RX/TX control icon in the toolbar)
- Link speed configuration (points to the link speed dropdown in the General section)
- Statistics selection (points to the Statistics checkbox in the table)
- Port configuration (points to the Port Mode dropdown in the table)
- RX selection (points to the RX checkbox in the table)
- TX selection (points to the TX checkbox in the table)
- Frame editor (points to the Traffic File field in the table)

Status: Running

PVS Phase 1 System Architecture



SpW/SpW-T FPGA



IP Core Provided by PLDA

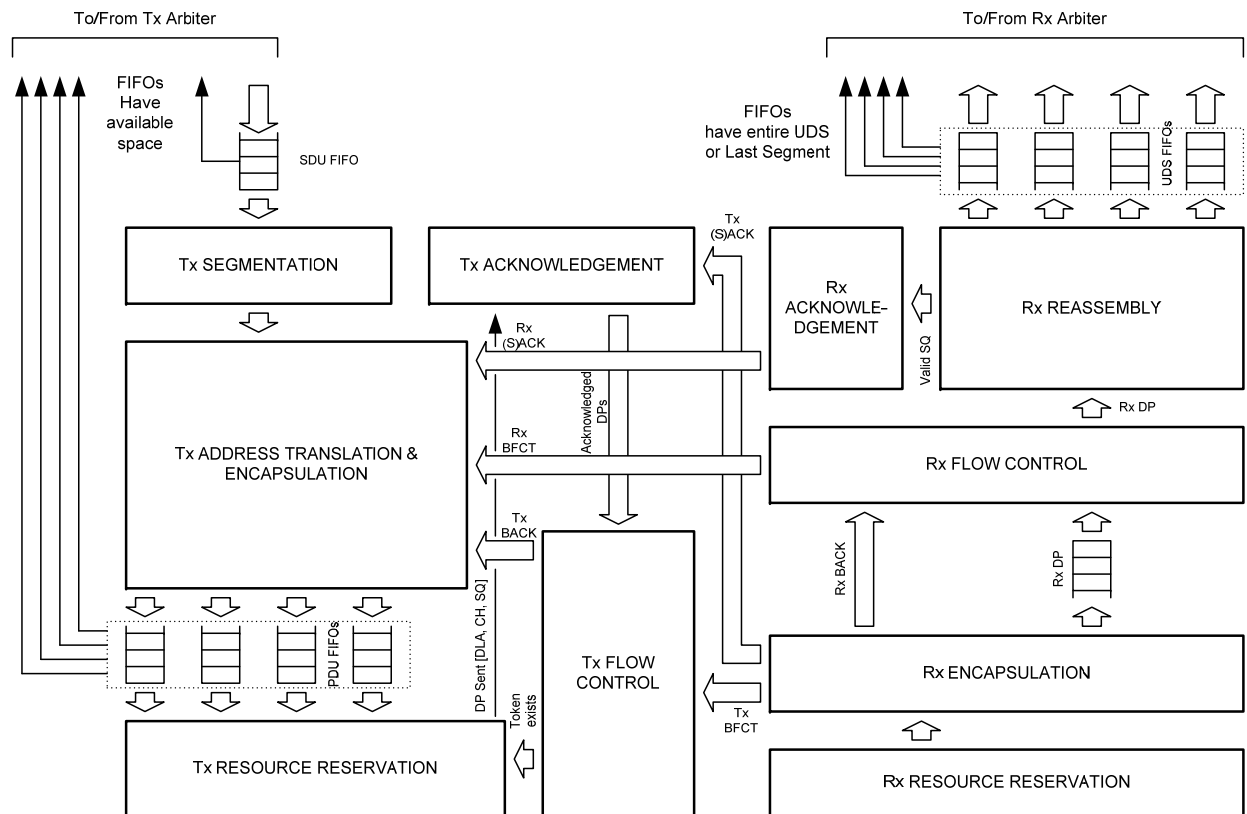
IP Core Provided by ESA/ University of Dundee

Blocks Developed by TELETEL

----- Modules instantiated in SpW-T FPGA only

SpW-T Block

- ✓ Segmentation
- ✓ End to End flow Control
 - ✗ SBFCT support
 - ✗ BFCT Timeout/Retransmission
- ✓ Acknowledgement
- ✓ Address Translation
 - ✗ Path addressing
- ✓ PDU Encapsulation
- ✓ Resource Reservation
 - ✓ Header/data CRC
 - ✓ Sequence Number
 - ✓ Missing ACK



SpW-T Implementation metrics

	Registers	LUTs	Slices	BRAMs
Segmentation	106	197	66	
Tx Encapsulation	671	1182	552	8
Tx Acknowledgement	619	837	35	1
Tx Flow Control	50	124	66	
Resource Reservation	354	325	250	
Rx Encapsulation	400	341	295	1
Rx Acknowledgement	70	141	78	
Rx Flow Control	64	123	60	
Reassembly (Logic)	170	164	96	
Reassembly (Buffers)	588	412	464	4
Tx Statistics	18	86	35	1
Rx Statistics	22	259	116	3
SpW-T Block	3313	4639	2405	18

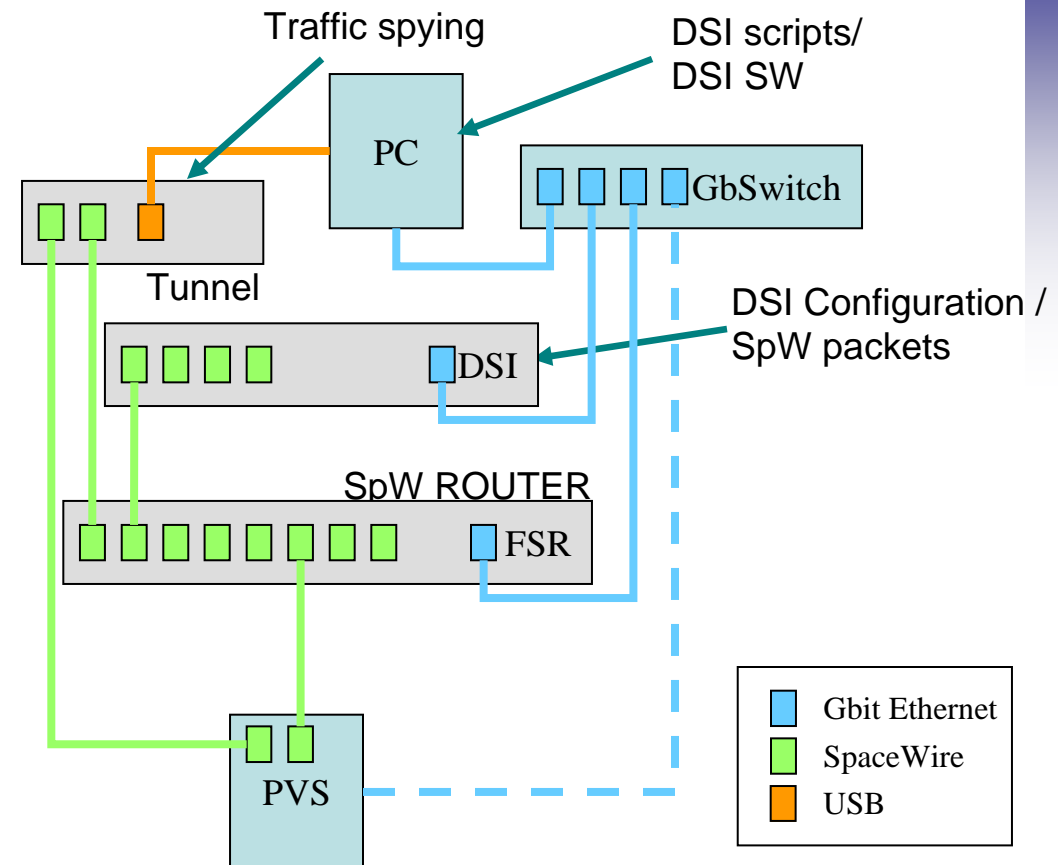
	Registers	LUTs	Slices	BRAMs
SpW-b Core	520	528	383	2

	Registers	LUTs	Slices	BRAMs
Tx DMA Arbiter	621	677	337	
Rx DMA Arbiter	540	964	363	
Pointers Bank	204	147	110	2

	Registers	LUTs	Slices	BRAMs
PVS with 2 x SpW/SpW-T	20786	26942	13360	64

SpW-T Test Bed

- PVS SpW-T tested against SpW-T SW implementation on Linux 2.6 using 4-Links DSI
- 4Links FSR router
- Monitoring through Star-Dundee IP Tunnel
- Remote integration tests through internet
- Same tests executed in remote & local configurations
- Endurance testing executed on Scheduled mode with transfers on more than 12 hours (65 GB logfile)



SpW-T Test Results

Description	Error injected	Error detected	Verdict
Nominal asynchronous/Scheduled communication	-	-	PASS
Asynchronous/Scheduled communication with error	SQ	YES	PASS
	Length	YES	PASS
	HDR CRC	YES	PASS
	Data CRC	YES	PASS
Asynchronous communication with missing ACK	ACK inhibit	YES	PASS
Asynchronous communication with invalid ACK	CH	YES	PASS
	SQ	YES	PASS
	CRC	YES	PASS
Asynchronous communication without congestion	-	-	PASS
Asynchronous communication with congestion	-	-	PASS
Scheduled communication without congestion	-	-	PASS
Scheduled communication with congestion	-	-	PASS

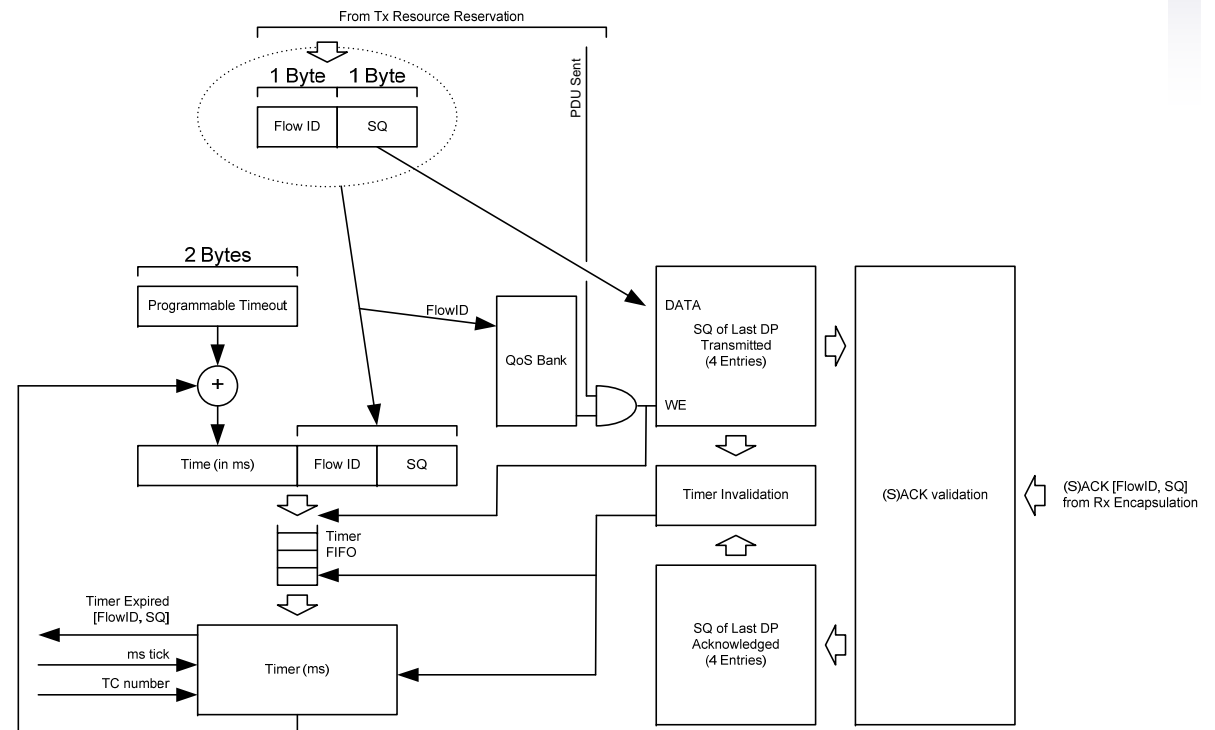
SpW-T V3.1 specification/implementation issues

- Error handling is restricted to data errors
- Timing errors are not addressed (e.g. Time Code loss)
- Action to perform in case of error at the level of application is not defined
- Problem with BFCTs during initialization. If destination sends BFCTs while source is not ready the BFCT is lost. The destination shall retry until BACK is received. How many times?
- The SBFCT time constraint (3 us in the example of the V3.1 spec.) not realistic for SW implementation & requires high speed HW operation (> 100MHz)
- Need to access the Token buffer through the application (e.g. in case of PDU loss the BFCT is consumed and never received from the remote side)
- SQ storage at various functions requires many memory resources
- The need for separate UDS buffers at the receiver increases memory needs even more
- Other minor issues (e.g. values not specified for DP, ACK/SACK, BFCT/SBFCT, BACK)

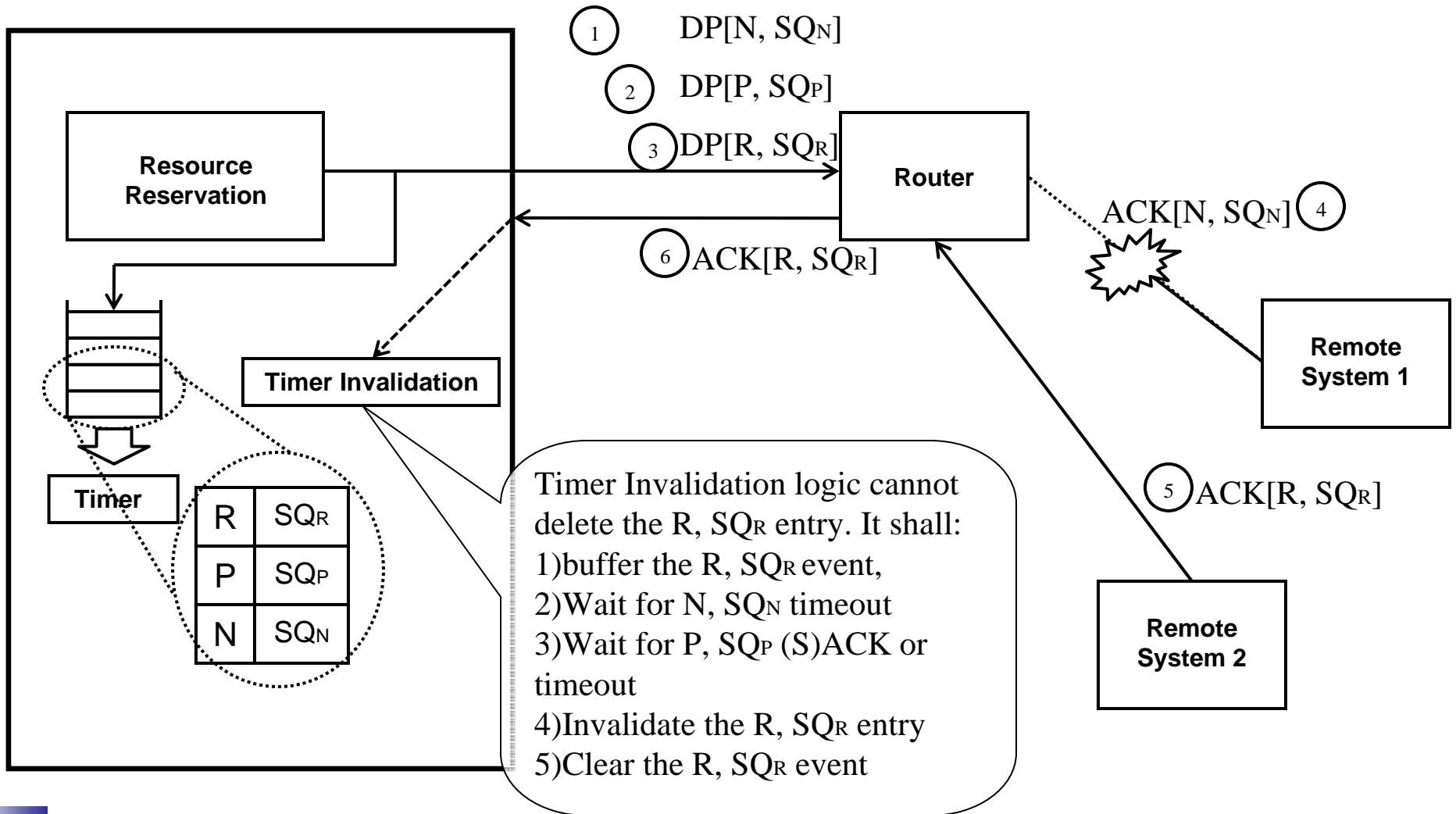
Issues to be considered on the next spec. revision - Acknowledgement (1/2)

For the acknowledgement function the handling of (S)ACKs when multiple SpW-T channels are used, forced us to implement a common Timeout FIFO, which **significantly complicates the design of the “timer invalidation” block in order to:**

- Handle out-of-order ACKs
- Compensate for ACK losses



Issues to be considered on the next spec. revision - Acknowledgement (2/2)



Issues to be considered on the next spec. revision – Channels handling

Problem: The combination of {Destination Address, Channel ID, Source Address} do not form a contiguous address space

- Look up cannot be performed to associate this combination with a certain flow

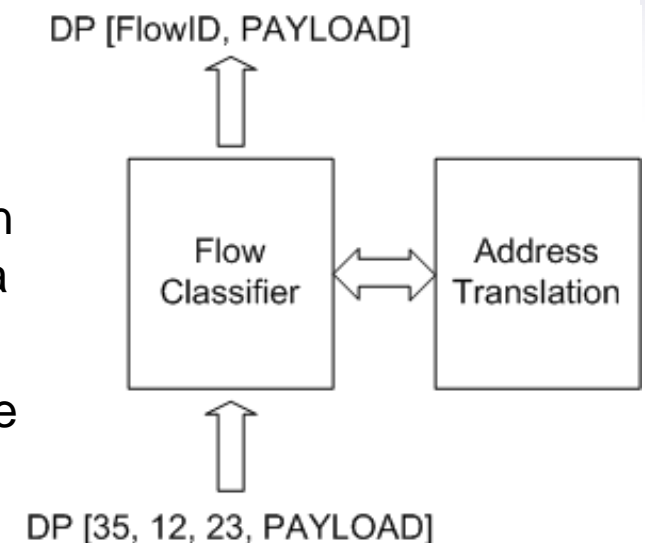
- The search shall be performed by several SpW-T functions (Reassembly, Acknowledgement, Flow Control)

Alternatives: A Classifier block, replacing the {Destination Address, Channel ID, Source Address} combination with a FlowID was developed

- CAM based: Expensive, slows down overall performance

- Linear search: Slow, not scalable

- Binary search: Scalable but more complex



- PVS/DSI is among the first validated SpW-T implementations
- Current specification (v3.1) has several open issues
- Next specification revision shall heavily consider:
 - Implementation issues (!)
 - Error handling at application level



Antonis Tavoularis,
Vangelis Kollias
TELETEL SA
A.Tavoularis@TELETEL.eu
V.Kollias@TELETEL.eu
www.teletel.eu

teletel

Christophe Honvault
EADS Astrium SAS Satellites
christophe.honvault@astrium.eads.net
www.astrium.eads.net

