Flexible Communication Among DDS Publishers and Subscribers

Christian Esposito and Domenico Cotroneo

Università di Napoli
Federico II

Dario Di Crescenzo

Consorzio SESM
Soluzioni Innovative

www.mobilab.unina.it

christian.esposito@unina.it

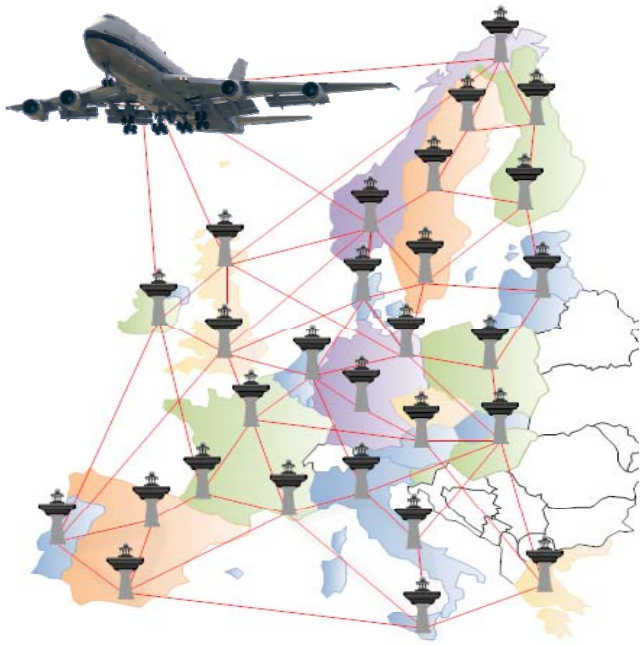
The present work is the result of the collaboration between Mobilab Research Group (Università di Napoli) and Selex-SI under the project "IniziativaSoftware" (http://www.iniziativasoftware.it).

::.. Problem Statement(1/2)

Ultra-Large-Scale (ULS) systems are characterized by a federation of heterogeneous and autonomous systems interconnected by wide-area and unmanaged networks.

Due to geographical extension and political reasons, there is no central control in ULS systems

It is possible to have a non-uniform evolution of the system: modified and not-modified applications have to coexist (multi-versioning).



::.. Problem Statement(2/2)

The PSM over UDP of RTPS specification indicates Common Data Representation (CDR) as Wire Representation for the exchanged messages.

CDR couples the comprehension of the received message to the knowledge of the sent message structure. Therefore, if the publisher changes the message structure, the unchanged subscribers cannot understand the received message.

The achievable communication is not flexible

::.. Objective

1. **Definition of other serialization formats that allow guaranteeing multi-versioning property;**
2. **Realization of a DDS-based prototype that adopts these formats instead of CDR to serialize and deserialize application data;**
3. **Assessment of the performance achievable with these formats;**
4. **Comparison with the case of using CDR.**

::.. Solution (1/2)

To obtain flexibility, CDR has to be replaced with a tree-based serialization formats:

The publisher disseminates on the wire not only the message content but also a representation of the message structure as a tree so to overcome the CDR limits.

XML is widely adopted to facilitate the sharing of structured data across different systems, particularly in a wide area scenario.

Message structure is coded into the opening and closing tags that wrap the data.

::.. Solution (2/2)

XML syntax is redundant or large relative to binary representations (e.g. CDR) of similar data, and this affects application efficiency through higher transmission and serialization costs.

More efficient serialization formats are needed

JSON

YAML

JSON and YAML are two tree-based formats that offer the same message structure coding as XML, but adopt a mechanism of name/value pair that allows saving bytes.

::.. Performance Evaluation I (1/8)

A measurement-based campaign has been conducted using the DDS solution provided by RTI and these parsers:

- **CDR: In-house developed parser;**
- **XML: XERCES (both in DOM and SAX mode);**
- **JSON: JSON-C (http://oss.metaparadigm.com/json-c/)**
- **YAML: LIBYAML (http://pyyaml.org/wiki/LibYAML)**

Scope: evaluate the performance when using tree-based serialization formats and compare with the CDR case.

::.. Performance Evaluation I (2/8)

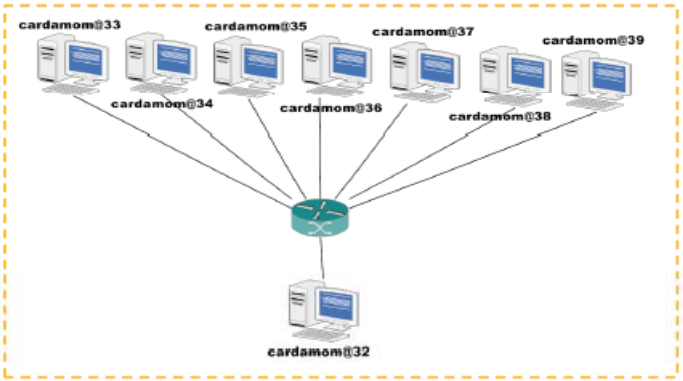
Metrics:

- **Serialization efficiency:**
  η = byte_to_send / serialization_stream_size
- **RTT Latency**

Test Settings:

- **Ping-Pong test between a publisher and a subscriber**
- **The testbed consists of two nodes of a cluster interconnected with a Gigabit Ethernet**
- **The communication is best-effort**
- **The type of the exchanged messages is the one standardized for the communication among ACCs**

::.. Performance Evaluation I (3/8)

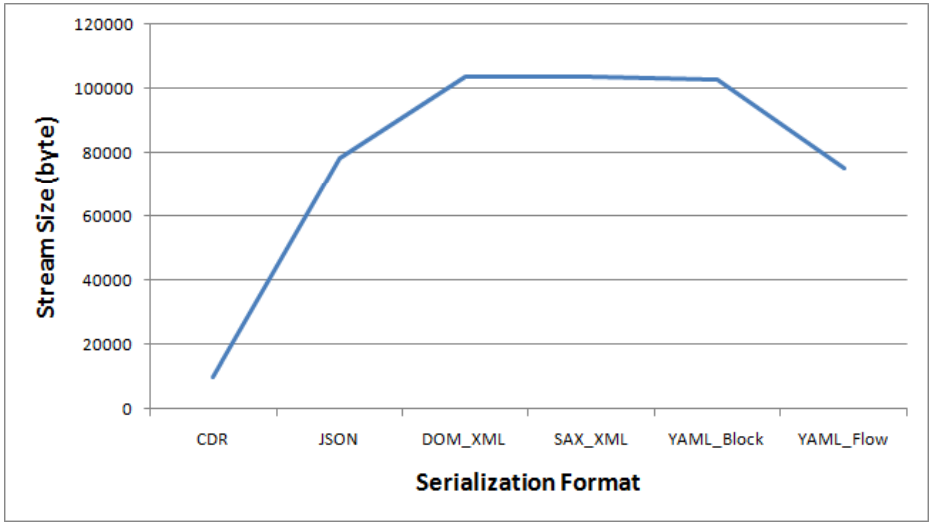


Operative System:	**Red Hat Enterprise Linux 4 (kernel 2.6.9 - smp)**
Compiler:	**GCC 3.4.3**
CPU:	**2 Xeon 2.8 GHz (with Hyperthreading)**
RAM:	**6 Gb**
Disk:	**36 Gb**
Network:	**2 Gigabit Ethernet / 1 Myrinet interface**
Intercon.:	**1 Stackable Switch with matrix > 56 Gbps / 1 Myrinet Switch 32 ports**

:::. Performance Evaluation I (4/8)

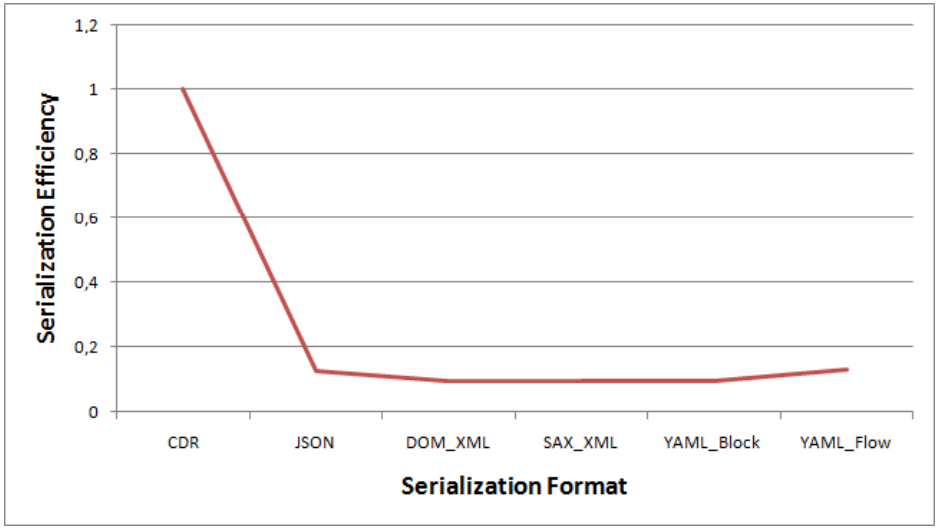
CompactAvenue

::.. Performance Evaluation I (5/8)

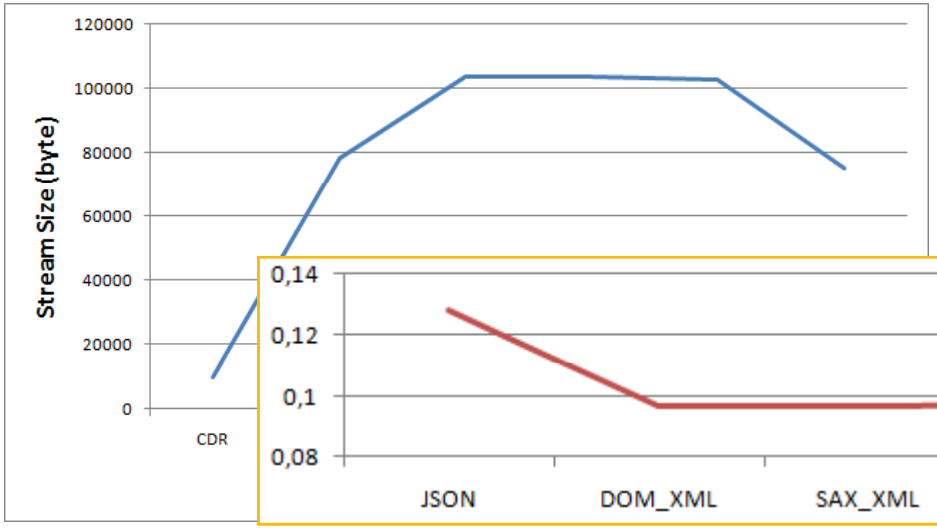


Tree-based formats have higher median stream size than CDR.

CDR is very efficient.



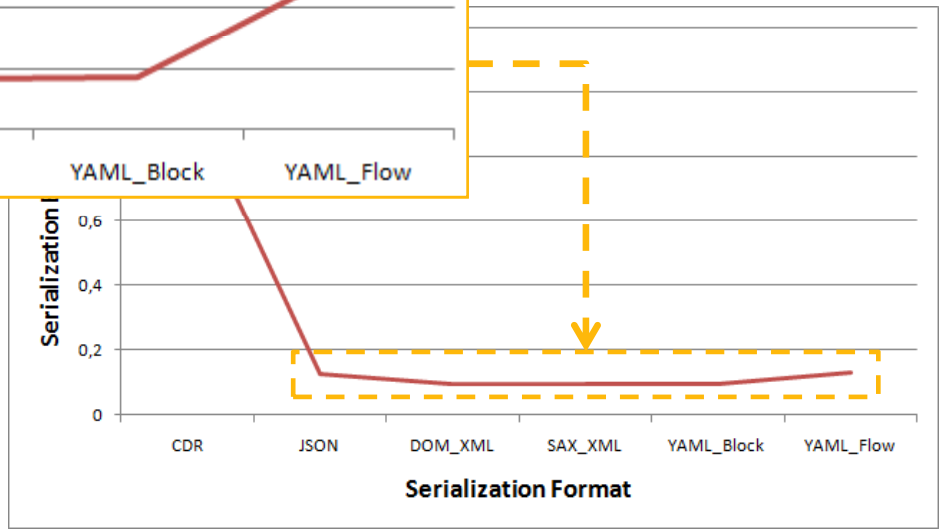
::.. Performance Evaluation I (6/8)



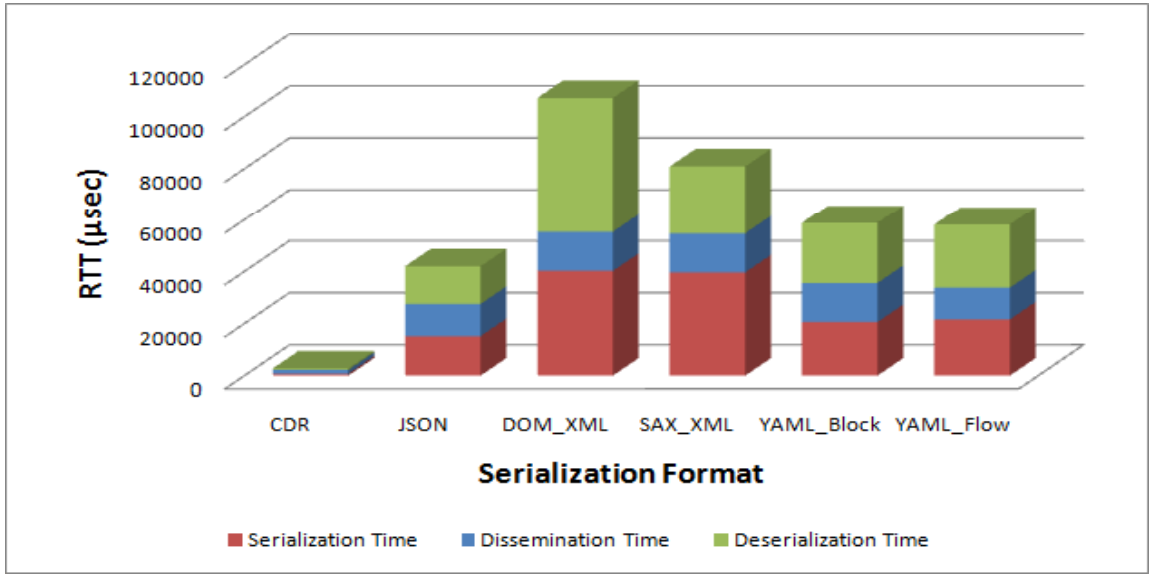
Tree-based formats have higher median stream size than CDR.

CDR is very efficient.

Tree-based formats have a very low efficiency.

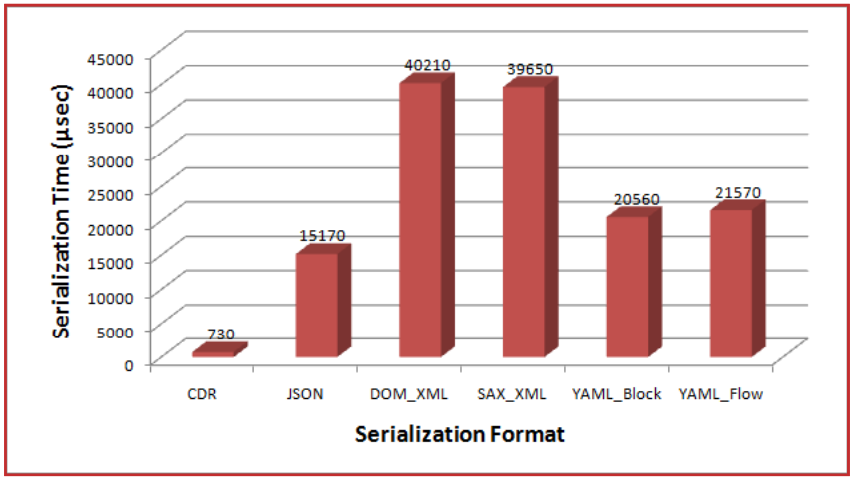


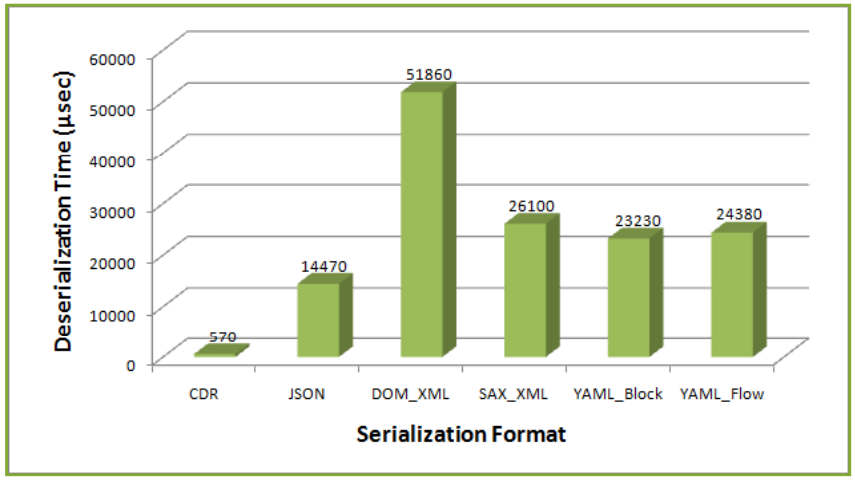
::.. Performance Evaluation I (7/8)



Because less bytes are exchanged over the network, CDR exposes better median performance than tree-based formats. XML have highest latency, but using SAX can reduce the overall performance.

::.. Performance Evaluation I (8/8)

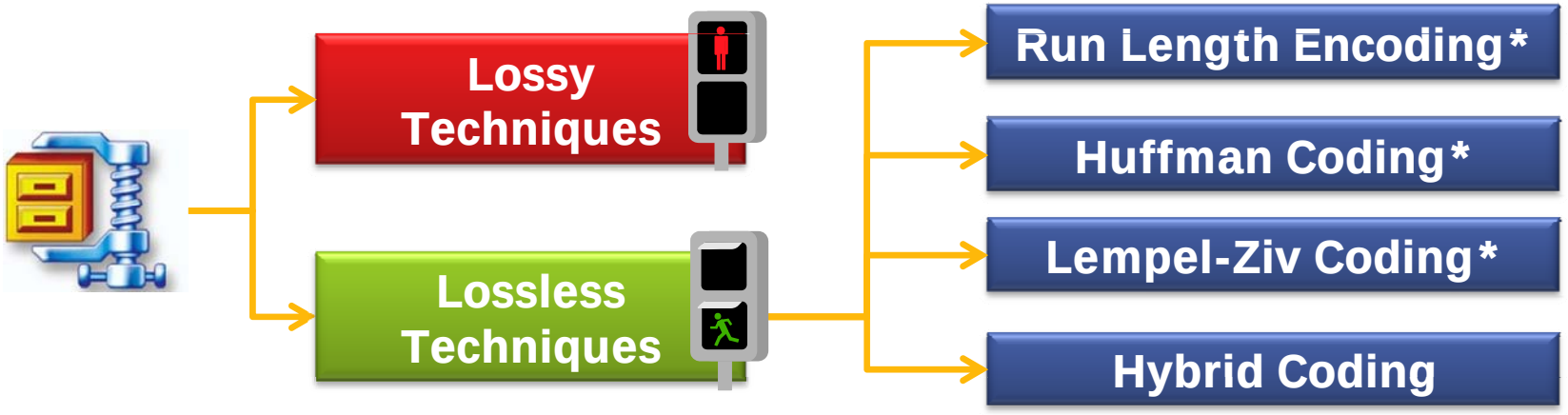




JSON and YAML parsers are more efficient than DOM XML, and SAX XML has performance comparable to YAML when deserializing data.

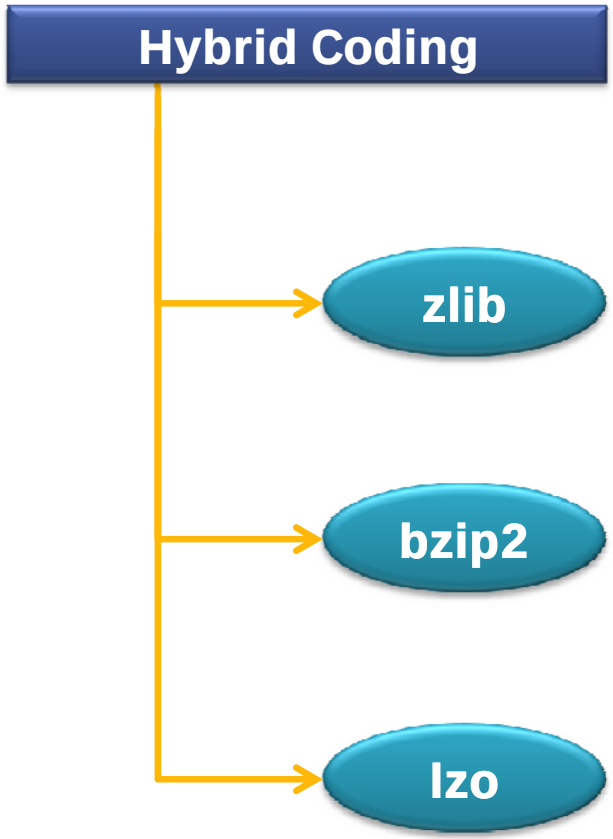
::.. Data Compression (1/2)

The solution to improve the efficiency of the tree-based formats is using a data compression technique, so to reduce the bytes exchanged over the network.



*(http://sourceforge.net/projects/compressions/)

::.. Data Compression (2/2)



The basic compression techniques can be combined in order to achieve better efficiency and performance

Adopts the 'DEPLATE' method, a combination of LZ and Huffmann Coding (http://www.zlib.net/)

Use the Burrows-Wheeler block sorting technique and Huffman coding (http://www.bzip.org/)

Use Lempel-Ziv-Oberhumer algorithm, that is similar to zlib but faster (http://www.oberhumer.com/opensource/lzo/)

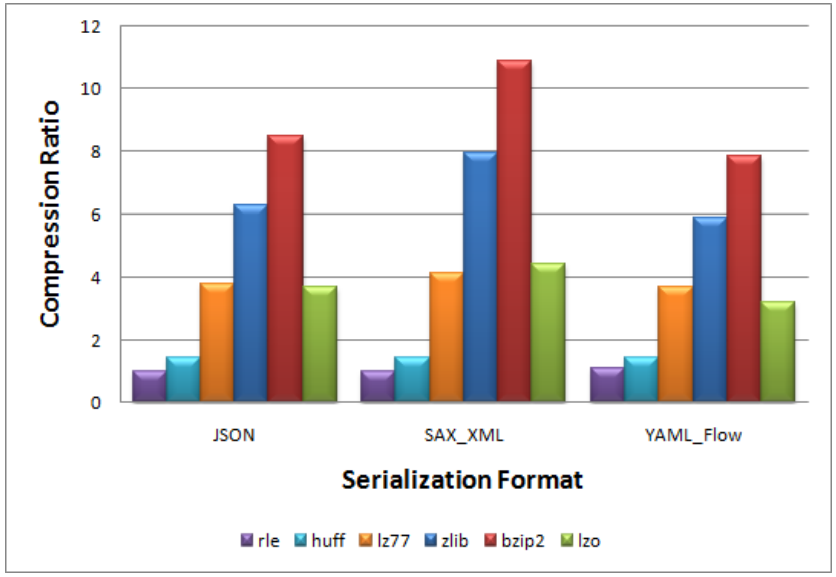
::.. Performance Evaluation II (1/6)

Scope: measure the performance of the DDS product when using data compression and assess the achievable improvement in terms of stream size and latency.

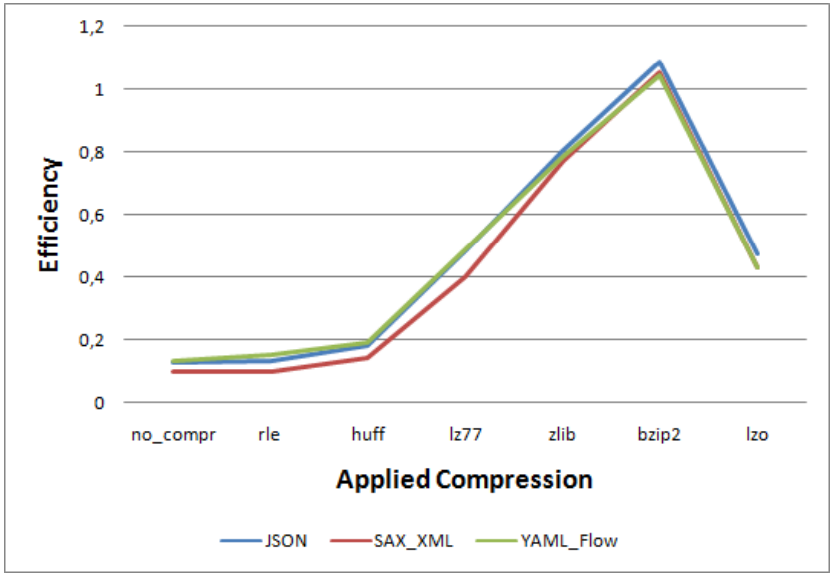
Metrics:

- Compression ratio
  χ = serialization_stream_size / compr_stream
- Compression+Serialization efficiency
  η = byte_to_send / compr_stream
- RTT Latency
- Compression+Serialization overhead:
  ω = RTT_CDR_format / RTT_compr+serial

::.. Performance Evaluation II (2/6)

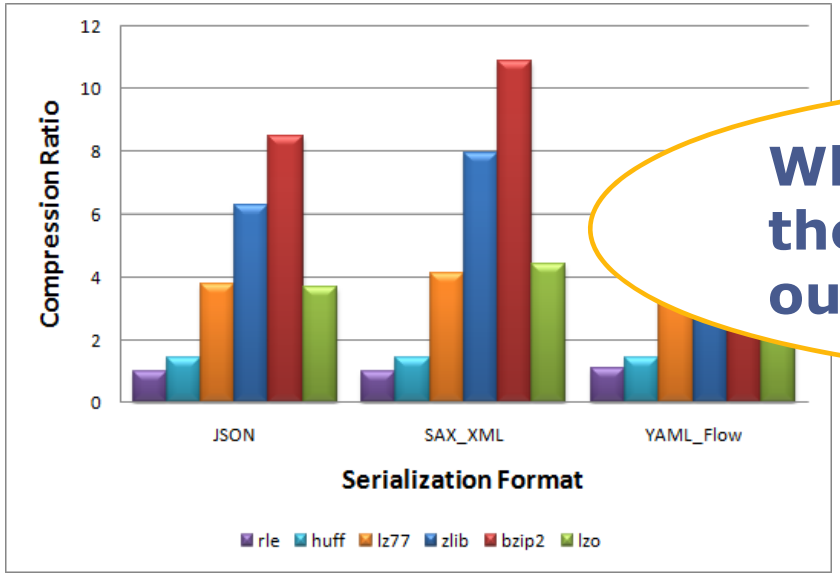


The best compression ratio is achieved by bzip2, while the two basic techniques have poor compression ratio.



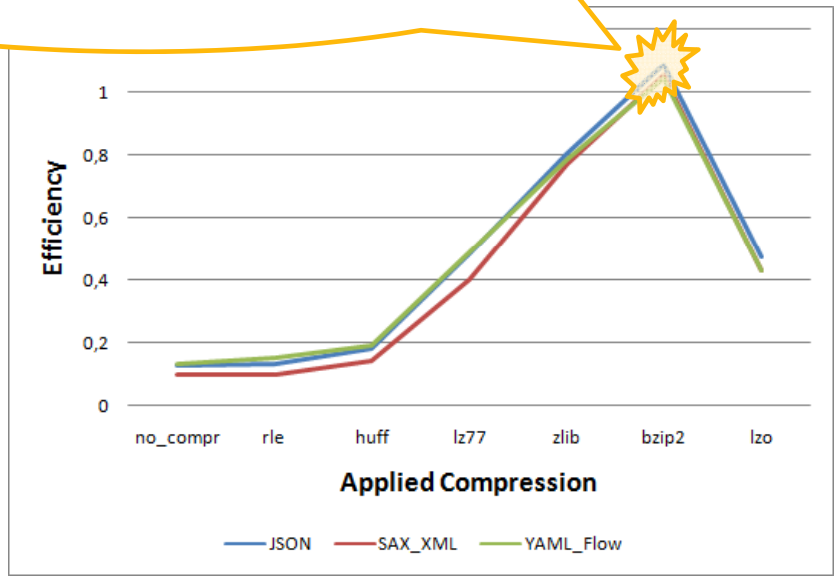
The use of compression technique allows the tree-based formats being more efficient.

::.. Performance Evaluation II (3/6)



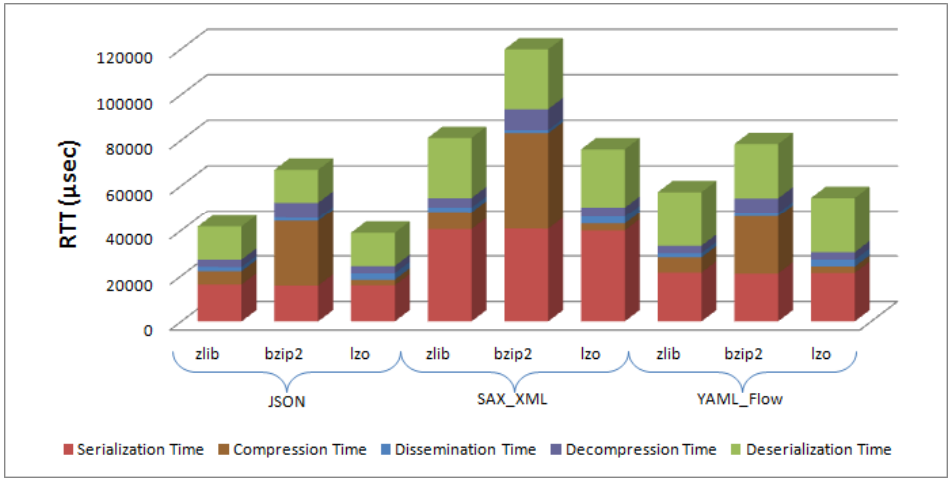
The best compression ratio is … while the … poor

When using bzip2, all the tree-based formats outperform CDR.



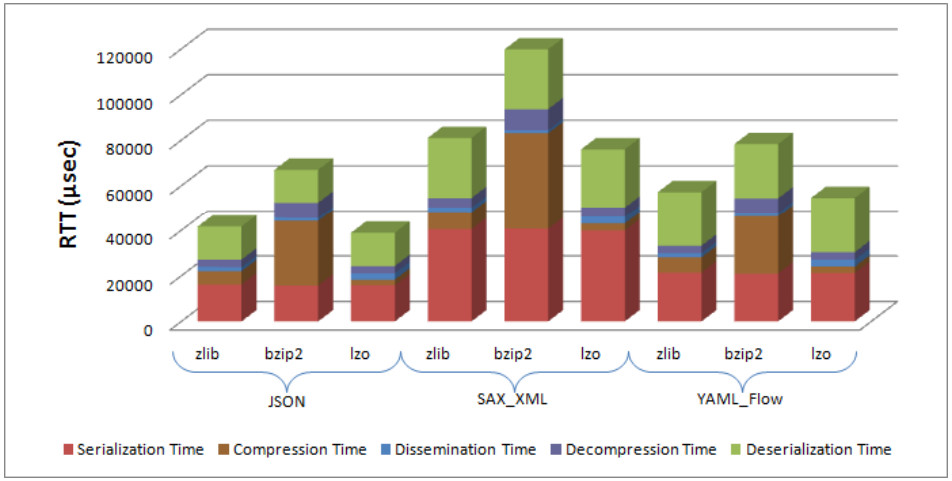
The use of compression technique allows the tree-based formats being more efficient.

::.. Performance Evaluation II (4/6)

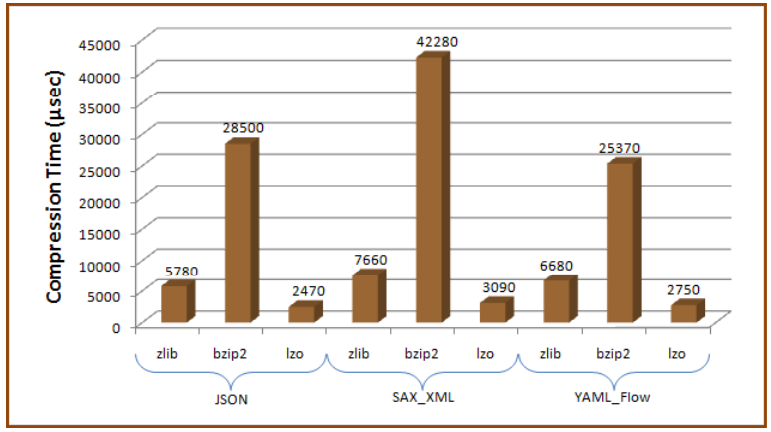


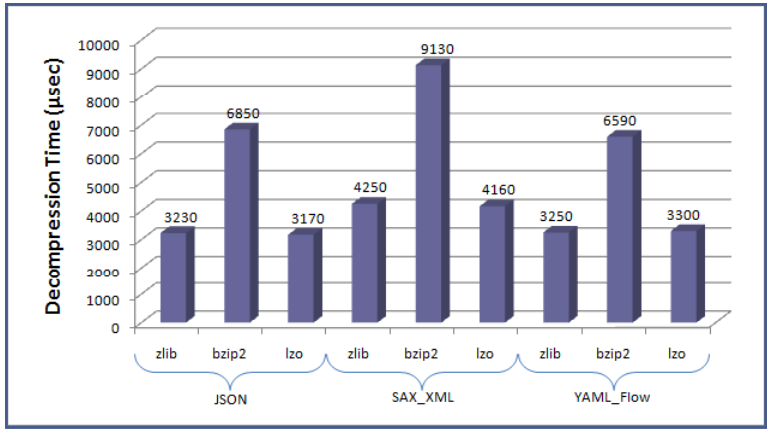
Bzip2 exposes the highest RTT, while lzo the lowest.

::.. Performance Evaluation II (5/6)

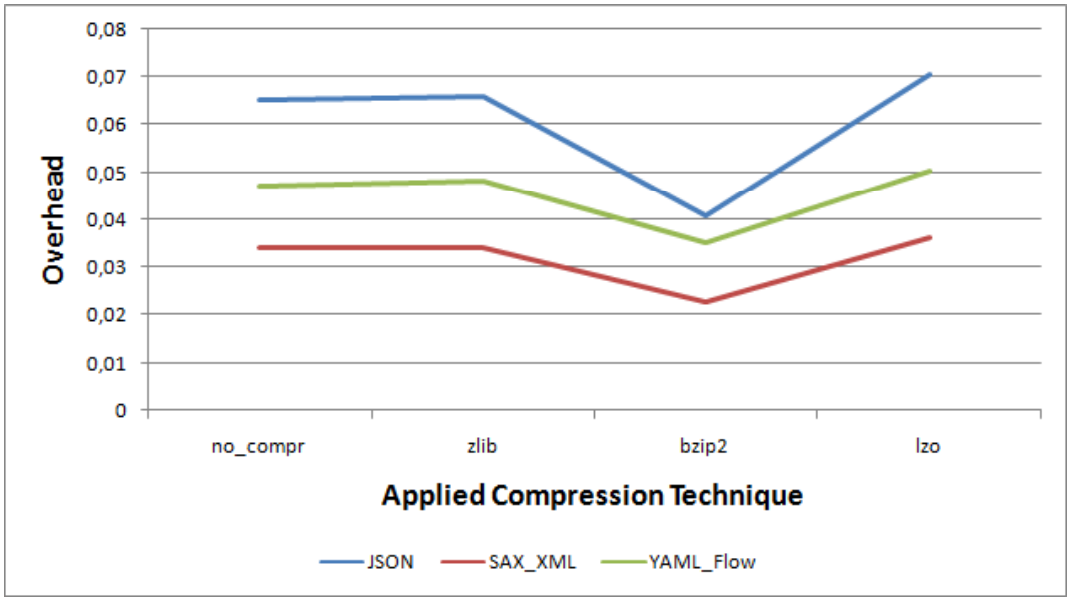


When compressing, Bzip2 is extremely slow, while lzo is very fast.
When decompressing, zlib and lzo have almost the same performance, while bzip2 is slower.





::.. Performance Evaluation II (6/6)



Compression techniques do not noticeably improve the overall RTT.

Lzo exposes better RTT than the case without applying any compression technique.

The overhead to pay for compressing and decompressing tasks almost completely overrides the benefit of having a smaller number of bytes exchanged over the wire.

::.. Performance Evaluation III (1/4)

Scope: evaluate the benefits of data compression in case of communications over wide-area and unmanaged networks

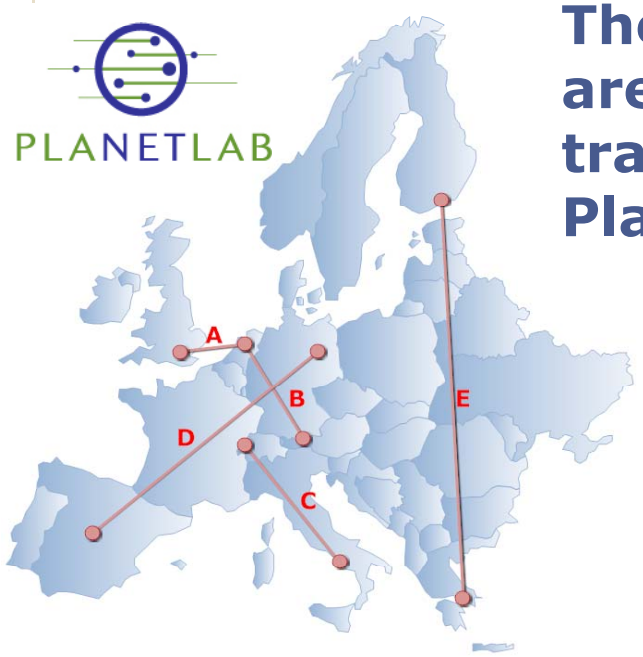
Metrics:

- RTT Latency
- Number of Lost Messages

TestBed: Shunra Network Emulator has been used to obtain a controllable testbed

How are the values of the emulator parameters chosen?

::.. Performance Evaluation III (2/4)



The realistic behaviour of a typical wide-area network is characterized through traffic measuring over several paths on PlanetLab.

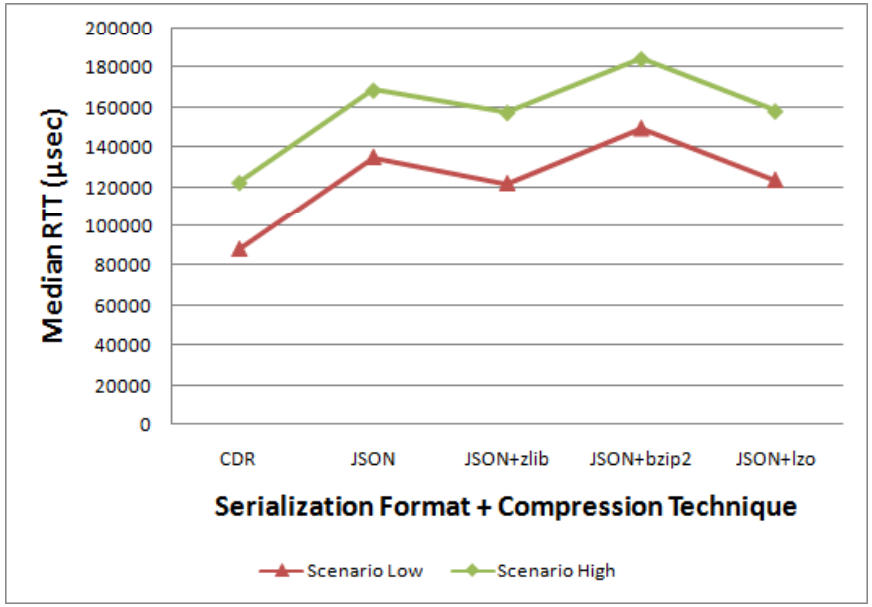
Scenario Low

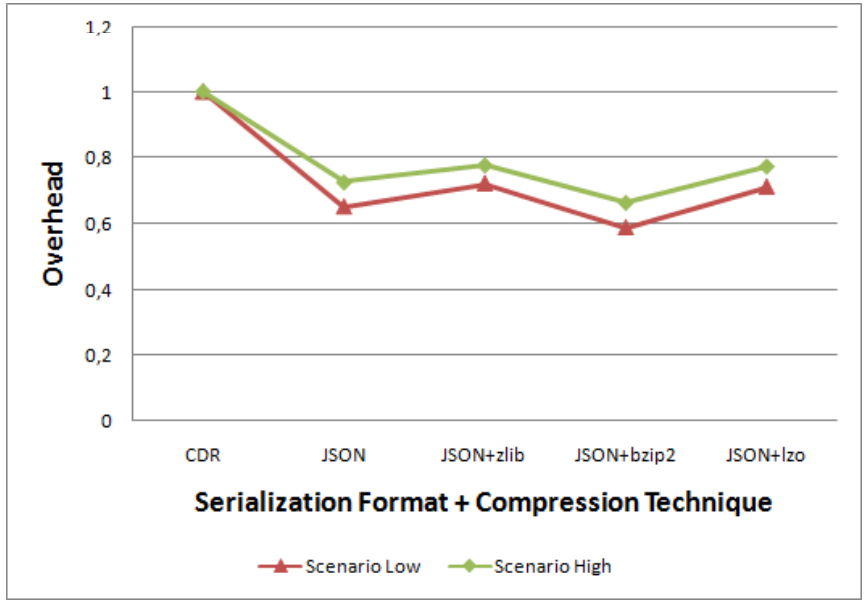
Scenario High

Path ID	Publishing Rate (mess. per sec.)	Delay (ms)		PLR		ABL	
		Median	IQR	Median	IQR	Median	IQR
A	200	8,9	0,87	3,35	1,24	1,59	0,28
	100	8,85	0,78	1,07	0,33	1,56	0,17
	50	8,96	0,53	0,4	0,12	1,07	0,08
B	200	17,95	0,56	5,28	1,78	2,56	1,62
	100	17,91	0,53	1,93	0,95	2,54	0,84
	50	18	0,42	0,74	0,91	1,46	1,45
C	200	27,16	18	5,34	4,86	1,26	0,32
	100	22,3	13,69	1,41	0,94	1,2	0,07
	50	23,48	10,6	0,35	0,22	1,02	0,06
D	200	27,75	1,65	5,78	1,74	1,11	0,05
	100	28,23	1,50	0,77	0,42	1,19	0,1
	50	28,25	1,55	0,2	0,08	1,17	0,26
E	200	43,81	0,78	5,05	0,85	1,44	0,11
	100	43,71	0,82	1,78	0,47	1,45	0,15
	50	43,71	0,63	0,65	0,31	1,14	0,14

Three scenarios are defined from the obtained measures.

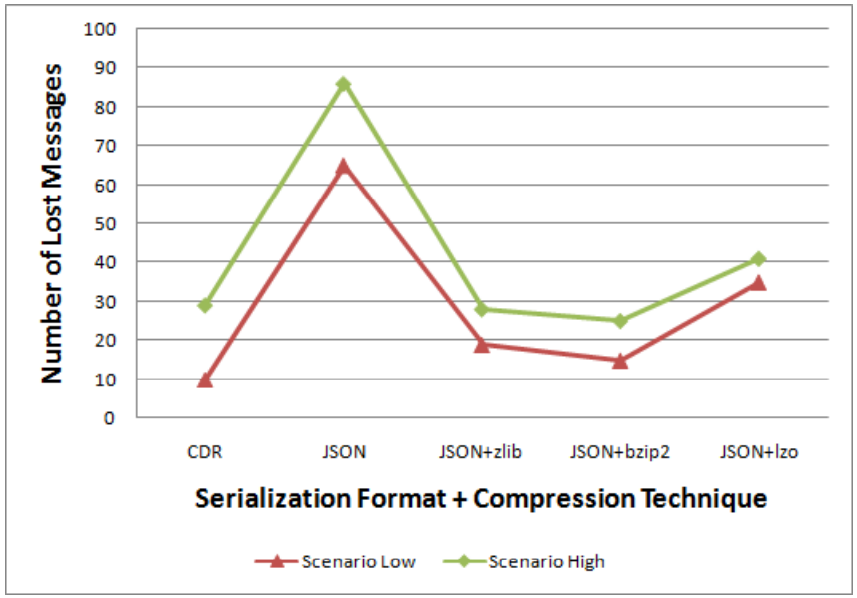
::.. Performance Evaluation III (3/4)





Except bzip2, compression techniques improve the overall achievable RTT. In the wide-area scenario, the gap between CDR and the JSON is not remarkable as in the LAN scenario.

::.. Performance Evaluation III (4/4)



The reduction of the bytes exchanged on the wire causes an improvement in the reliability of the communication.

Zlib is the suitable tradeoff between performance improvement and communication reliability.

::.. Conclusion

Tree-based serialization formats represent a possible solution to guarantee multi-versioning in DDS-complaint solutions, however they imply a very high performance penalty.

Compression techniques can limit this penalty and reduce the number of the lost messages.

Future Work: Implement a different solution, consisting of maintaining CDR as serialization format and somehow making it "self-describing".

Any questions?
Thanks for your attention…

www.mobilab.unina.it