Evaluation of SNMP-like Protocol to Manage a NoC Emulation Platform

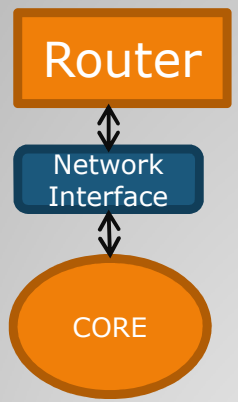
Otavio Alcântara and Virginie Fresse
Hubert Curien Laboratory UMR CNRS 5516
Jean Monnet University – University of Lyon, PRES Lyon
Saint-Etienne, France

Frédéric Rousseau
Univ. Grenoble Alpes, TIMA Lab., F-38031 Grenoble
CNRS,TIMA Lab., F-38031 Grenoble
Grenoble, France

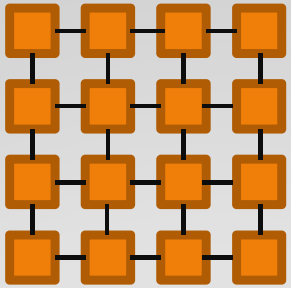
Agenda

- Overview
- Problem Formulation
- Proposition
- Emulation Platform
- Experiments
- Conclusions

- Networks-on-Chip
  - Promising communication technology for the modern many-cores SoC
    - It is based on computer network concepts



NoC Node



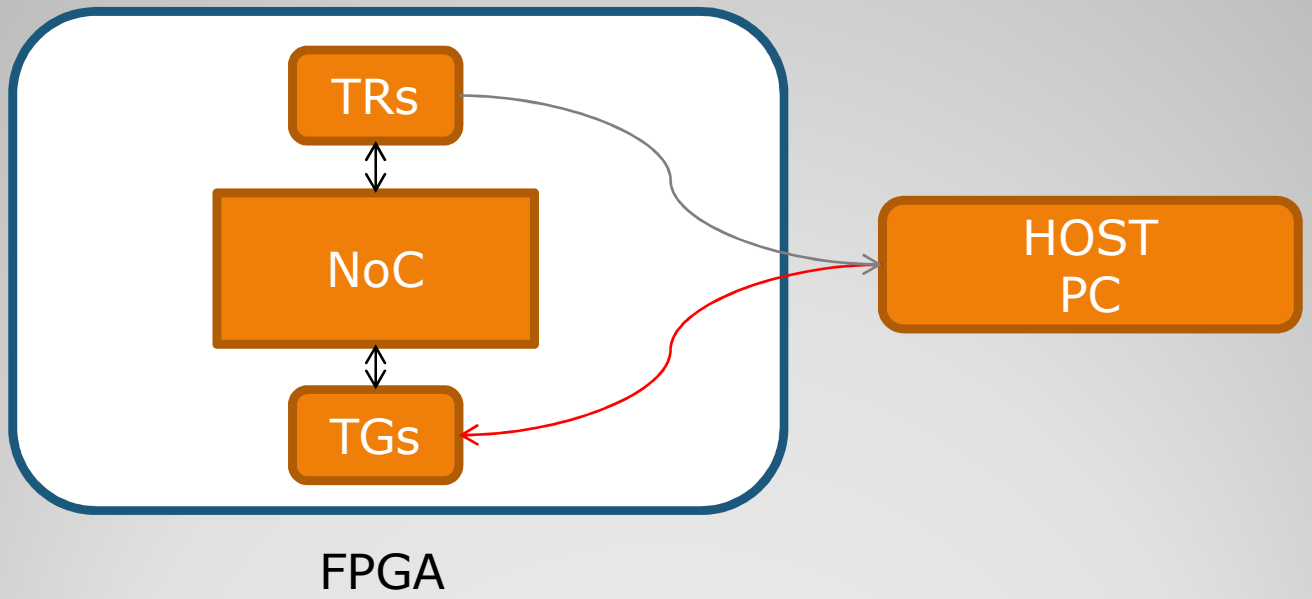
4x4 Mesh NoC

Large Design Space
- Topology
- Routing
- Flow control
- Task mapping
- Quality of Services
- Congestion control

    - NoC specialized tools are essential to SoC design

Overview

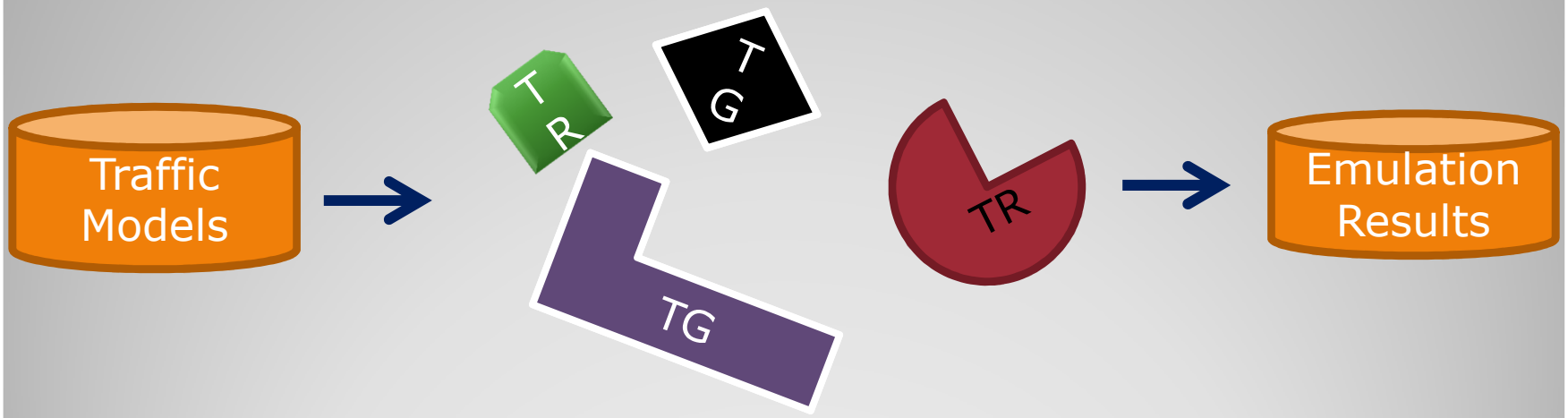
- NoC FPGA-based Emulation
  - Efficient technique for design space exploration and benchmarking
    - It reduces the design time as well as it provides high accuracy



Traffic Generators (TGs) –generate traffic based on applications or synthetic models
Traffic Receptors (TRs) –retrieve the packets and compute performance metrics.

Overview

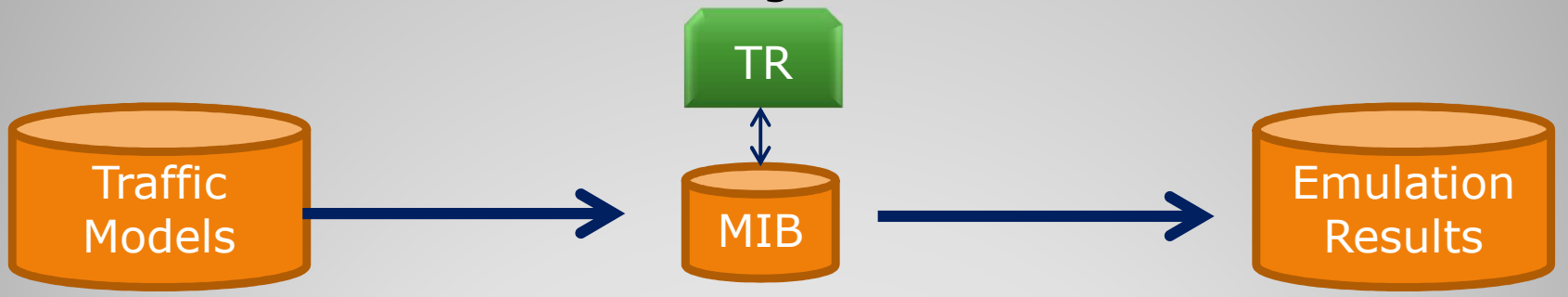
- Integrating emulation components from different manufactures



- A lack a standard protocol targeted at providing common interface to manage NoC emulator components.
- This protocol providing a common interface from the perspective of the management software running at the host PC

Problem Formulation

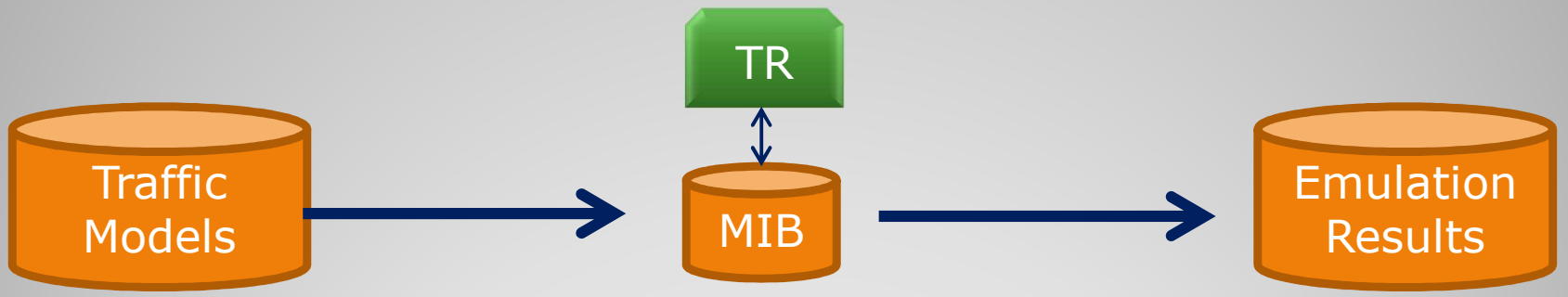
- SNMP
  - Simple Network Management Protocol
    - Used in network management systems to monitor network-attached devices
    - It is evaluated to manage an FPGA-based NoC Emulator



- The emulation components are managed through a set of standard operations executed in a Management Information Base (MIB)
- Operations on the MIB change the platform behavior it is possible to configure and execute several evaluation scenarios without FPGA re-synthesis

Proposition

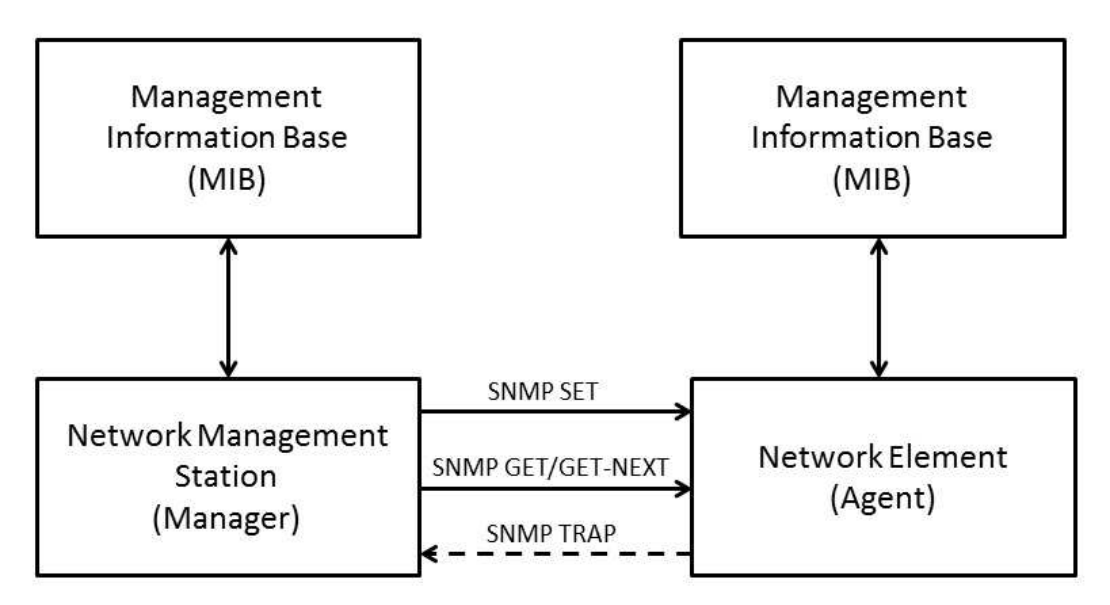
- SNMP
  - A standard protocol eases the integration of emulation components created by different designers



- Others computer network management protocols could be solutions for integrating the emulation platform (SMBus or NC-SI)
  - Although, theses protocols lack flexibility to manage a wide range of devices unlike SNMP that is a general purpose network management protocol

Proposition

- Application level TCP/IP protocol

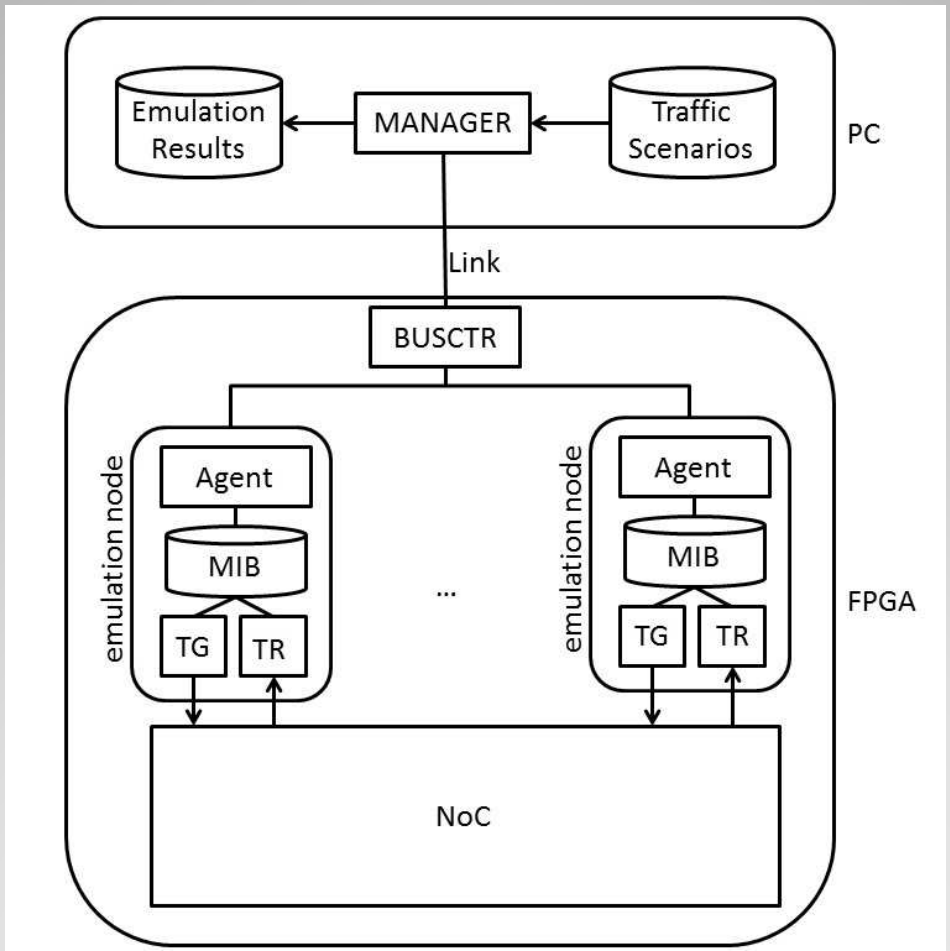


  - A request-response communication model allows the managers to manage the agents
    - The MIB description is standardized by the SNMP RFCs
    - In order to manage a device, it is necessary only to know its MIB structure

- Overview
- Problem Formulation
- Proposition
- **Emulation Platform**
- Experiments
- Conclusions

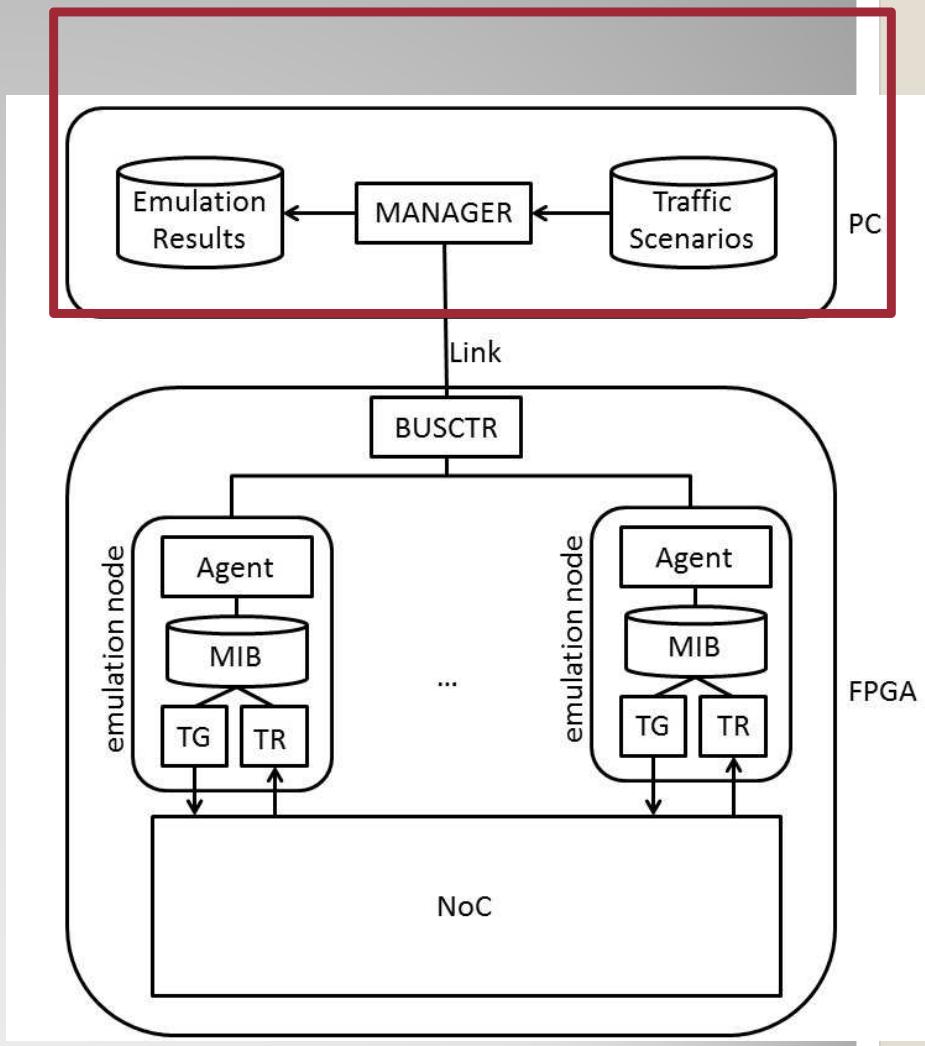
Agenda

- Architecture Overview



Emulation Platform

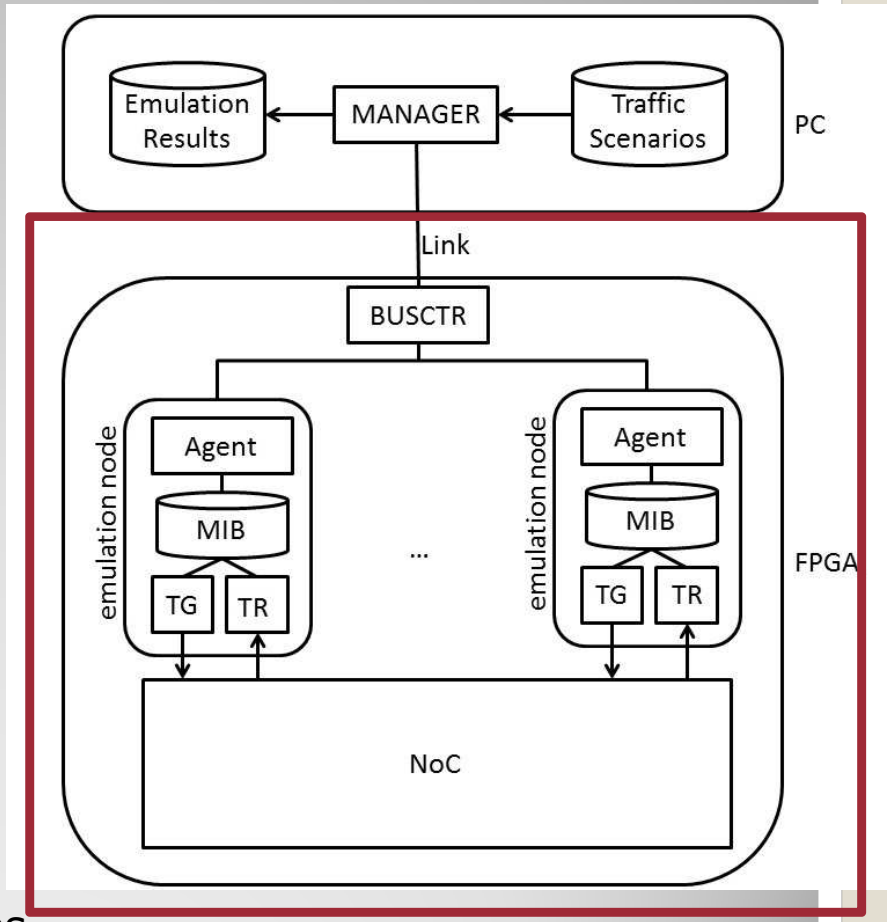
- Architecture Overview
  - The software components:
    - The manager communicates with the emulation platform
    - A management library eases the integration of the emulation with third party systems
    - The communication protocol is inspired on SNMP but it is not targeted for a TCP/IP
    - A set of test cases is found on the Traffic Scenarios
    - The last component saves the performance results



Emulation Platform

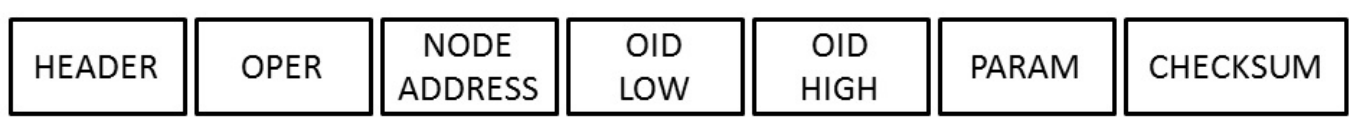
Architecture Overview

- The hardware components:
  - Emulation node
    - The agent is responsible for decoding and executing management commands
    - MIB is an addressable register bank, which contains all the control and status registers needed to manage a traffic scenario
    - TG and TR are the traffic related components
      - They are independent of the SNMP
  - BUSCTR
    - It controls the internal communication bus, which interconnects all emulation nodes



Emulation Platform

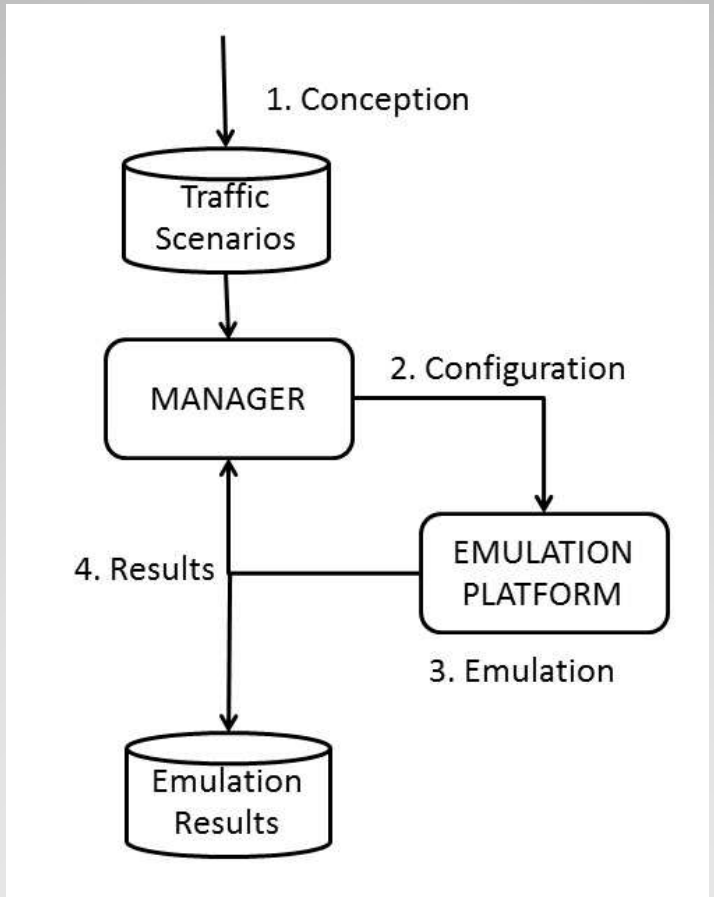
- Protocol implementation
  - It is implemented three SNMP operations
    - GET, GET RESPONSE and SET



- Traffic Models
  - SNMP model can be easily adapted to many traffic models
    - In this work, micro-benchmark and task graph traffic models are used.

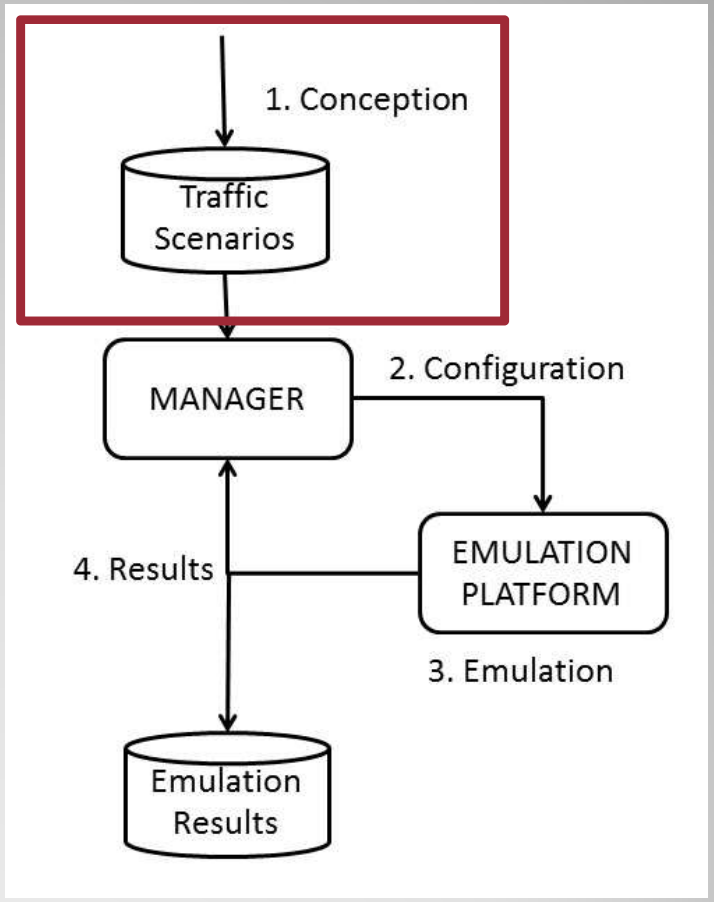
Emulation Platform

- Execution Flow



Emulation Platform

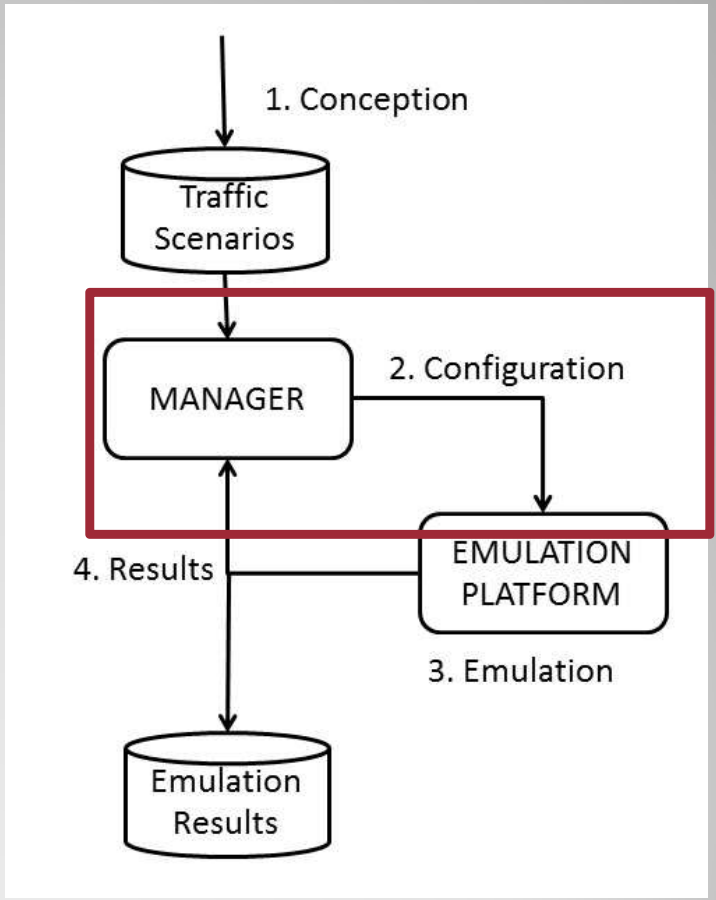
- Execution Flow
  - For each emulation node, it is defined the traffic scenario parameters
    - Number of destinations, injected charge, size of packets,
  - A set of traffic scenarios is created in this stage



Emulation Platform

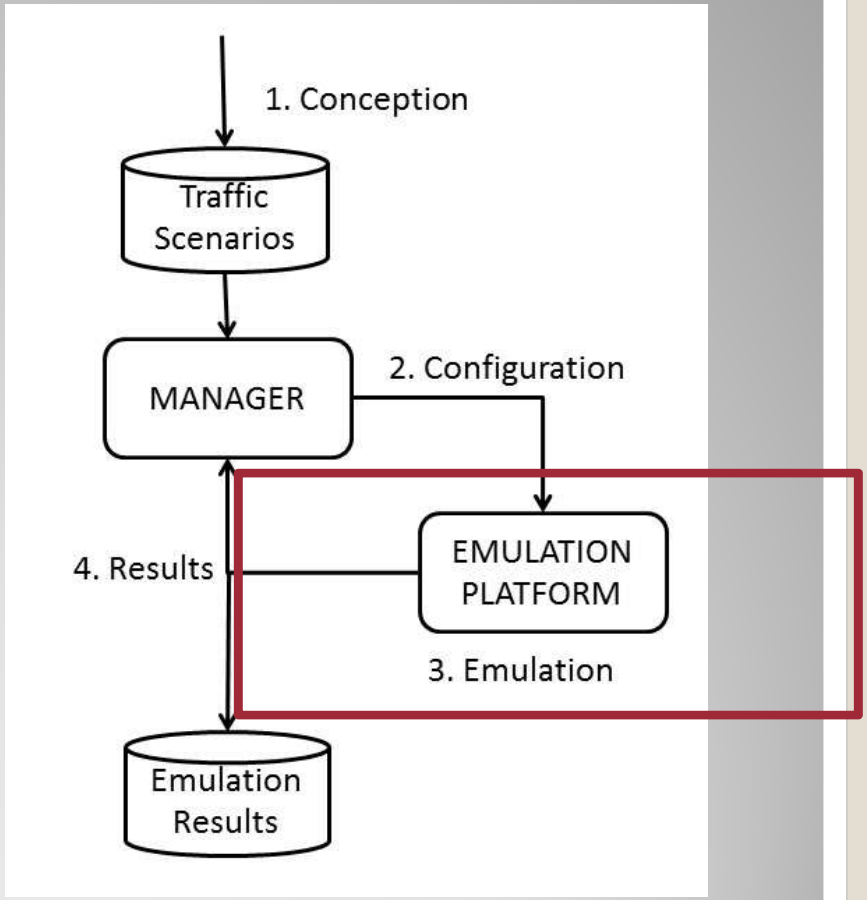
Emulation Platform

- Execution Flow
  - The manager library translates the traffic scenario description on a set of operations required to configure the platform
  - The manager sends SET packets to configure the traffic scenario on each MIB

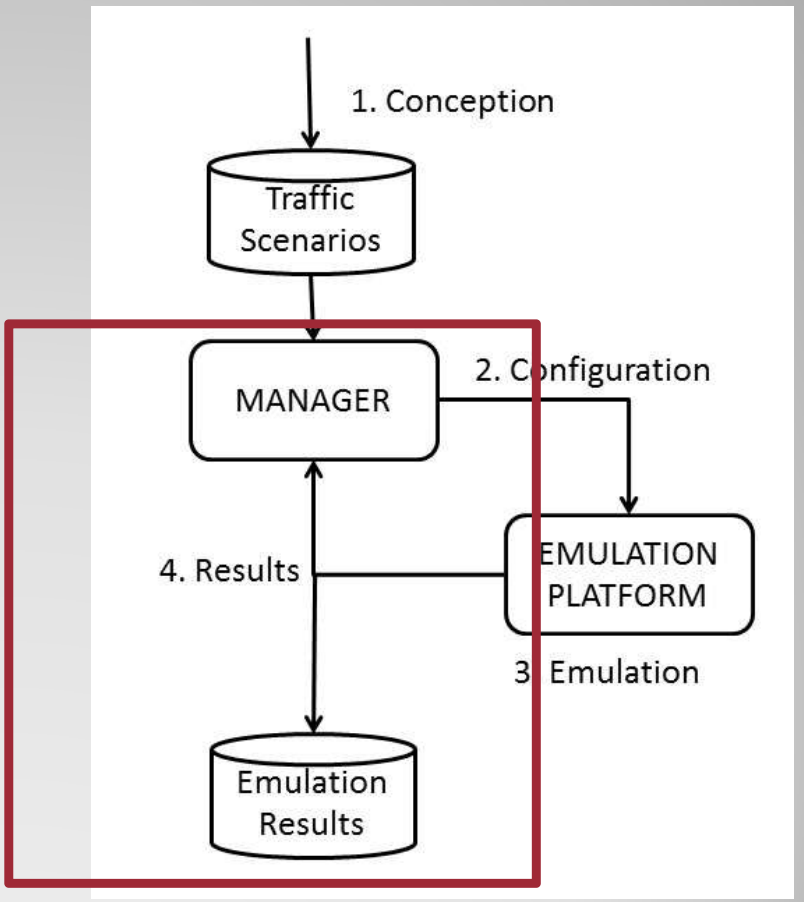


Emulation Platform

- Execution Flow
  - The manager sends a GO packet to start the emulation
  - The emulation lasts as long as necessary to carry out all the packet exchanges
  - At the end, the emulation platform sends an EMU_END packet



- Execution Flow
  - The manager sends GET packets to retrieve the performance results
  - The results are saved on a data base to be analyzed
  - The flow can continue to execute the next traffic scenario



Emulation Platform

- Overview
- Problem Formulation
- Proposition
- Emulation Platform
- **Experiments**
- Conclusions

Agenda

- Experimental setup
  - ML605 Virtex 6 evaluation board
  - Hermes NoC
    - XY routing, handshake flow control, 16-flit buffer
  - NoC and emulation components are synthesized using ISE 14.3/XST

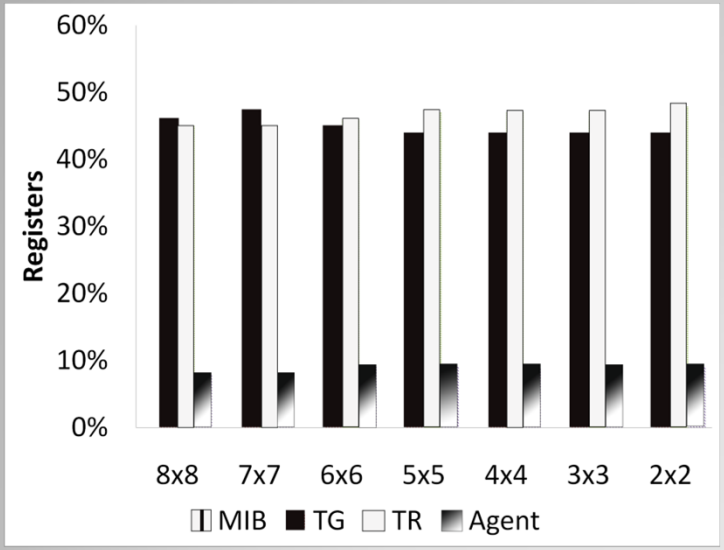
Experiments

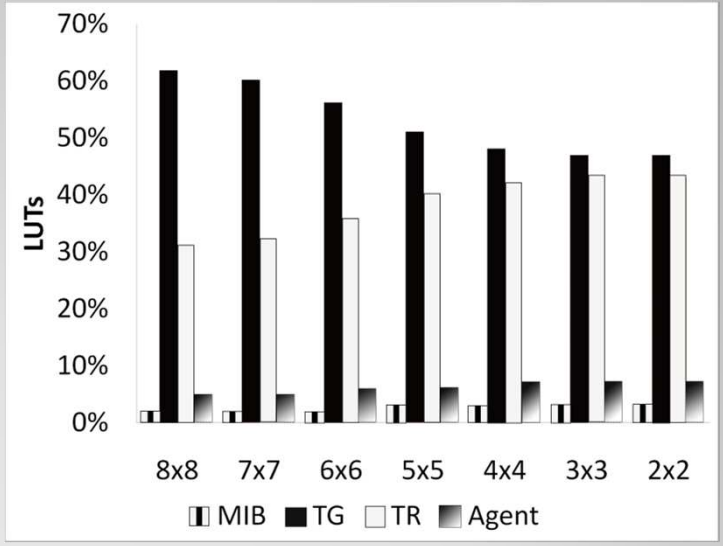
SNMP Operations Evaluation

- Each app is executed ten times on a 7x7 NoC
- The SNMP measurements are relative only to the execution of SNMP dedicated circuits
  - SNMP represents between 1.7% to 0.004% of the time spent to carry out a traffic scenario
  - There are a correlation between the number of tasks and the time spent on SNMP operations
  - The number of SETs is 36%-58% bigger than the number of GETs

Application	SNMP[ms]	Traffic [ms]	GETs	SETs	Tasks
MPEG-4	0.15	430.1	104	232	12
VOPD	0.08	1768.1	52	142	12
Multispectral	0.11	895.2	80	169	14
TGFF 0	0.04	10.7	36	65	9
TGFF 1	0.11	6.3	92	156	14
TGFF 2	0.06	7.0	52	189	7

- Resources Analysis
  - Micro-benchmarking traffic

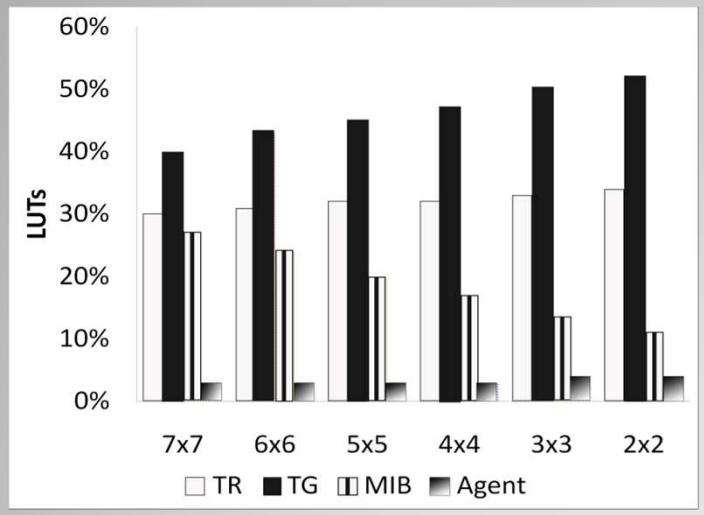


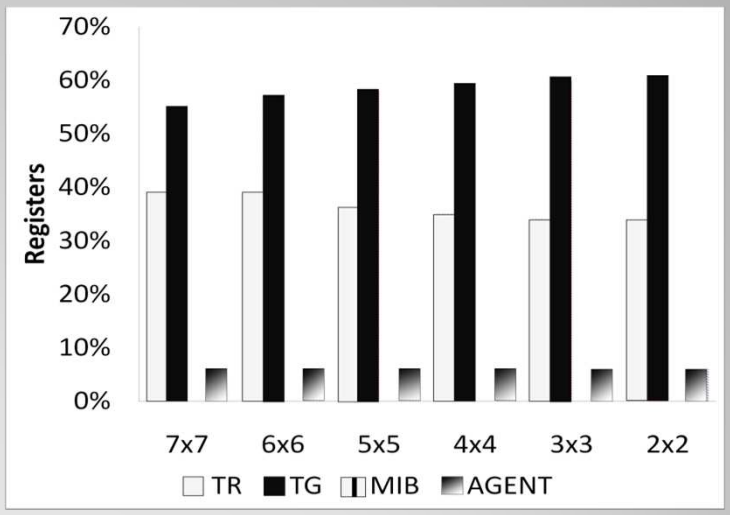


The analysis of an emulation node shows that the SNMP uses only 8% of registers and 7% of LUTs to implement one emulation node.

Experiments

- Resources Analysis
  - Task graph traffic





The analysis of an emulation node shows that the SNMP uses 6% of registers and 15%-30% of LUTs only to implement one emulation node.

Experiments

- SNMP operations are independent of NoC size
  - They represent a very small fraction (0.004% - 1.7%) of the time spent on the execution of a traffic scenario
- SNMP components are lightweight
  - They represent a small fraction of the overall platform
    - TR and TG represent a resources bottleneck
  - It is possible to implement a 8x8 NoC (micro-benchmark) and 7x7 NoC (task graph) on a ML605

Experiments

Conclusions

- We evaluate the SNMP protocol concepts to manage an FPGA-based NoC emulation platform
  - SNMP provides an interoperability model for emulation components based on the MIB
  - The experiments highlight that a light version of SNMP is very efficient for a light resources overhead

THANK YOU

Otavio Alcântara and Virginie Fresse
Hubert Curien Laboratory UMR CNRS 5516
Jean Monnet University – University of Lyon, PRES Lyon
Saint-Etienne, France

Frédéric Rousseau
Univ. Grenoble Alpes, TIMA Lab., F-38031 Grenoble
CNRS,TIMA Lab., F-38031 Grenoble
Grenoble, France