Centralized Traffic Monitoring for online-resizable Clusters in Networks-on-Chip

Philipp Gorski, Dirk Timmermann
Institute of Applied Microelectronics and Computer Engineering
University of Rostock, Germany
{philipp.gorski2, dirk.timmermann}@uni-rostock.de

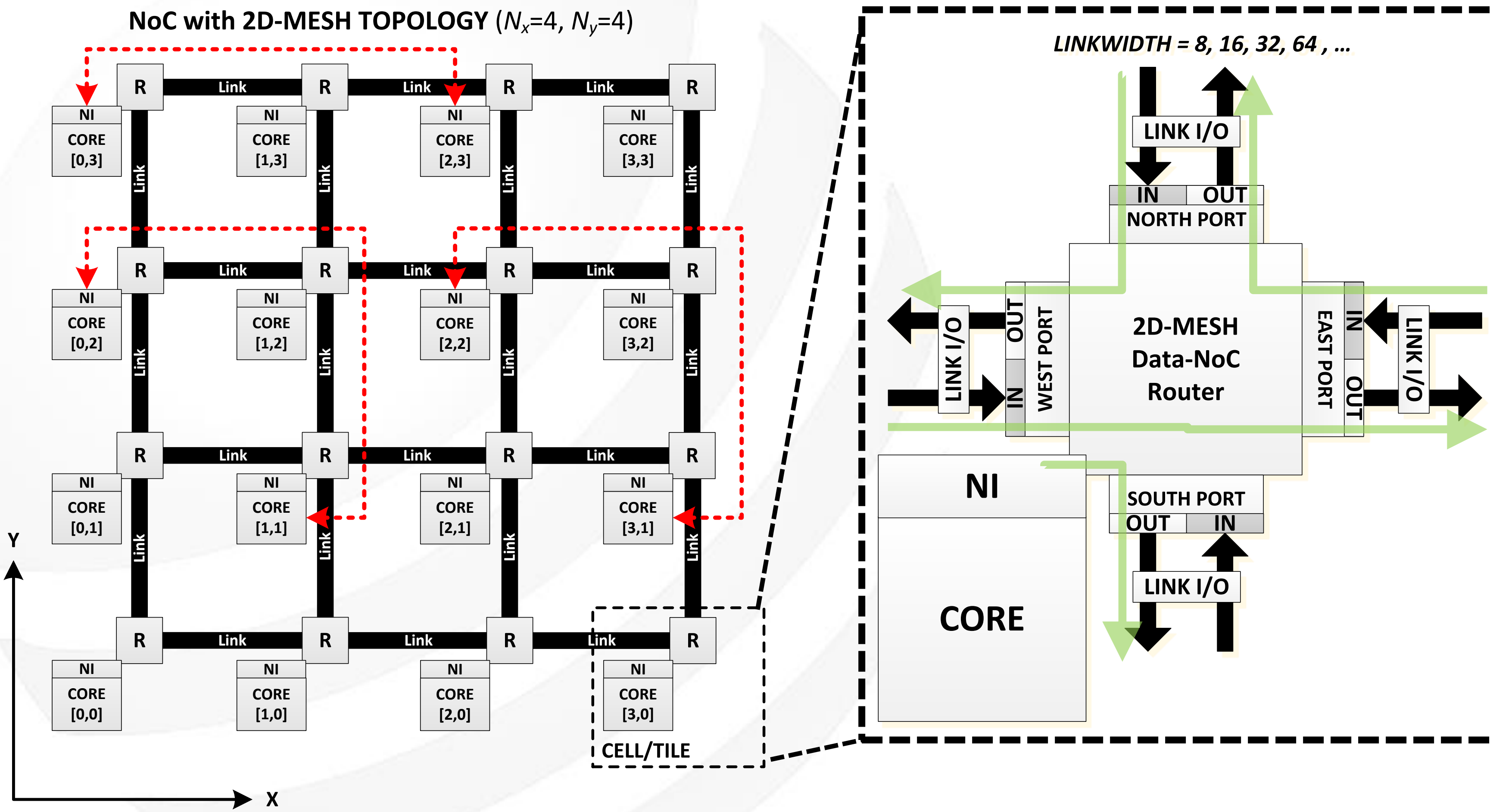
Outline

- **Introduction**
  - Networks-on-Chip (NoC)
  - Why Traffic Monitoring?
- **Traffic Monitoring**
  - Basic Concept
  - Dual NoC Infrastructure
  - Hardware/Software-based Clustering
  - Sensing and Flow
  - Experimental Results
- **Outlook & Future Work**

Introduction – Networks-on-Chip

- **Networks-on-Chip (NoC)**
  - Packet-based and globally asynchronous communication on-chip
  - Replacement of bus-based interconnections → Scalability & Parallelism
- **Basic elements of the NoC:**
  - **Ipcore** = Computational resource that communicates via the NoC
  - **Network-Interface (NI)** = Connection of Ipcore and NoC for reception/transmission of packets
  - **Router (R)** = Switching units that lead packets through the NoC from source to destination Ipcore
  - **Link** = Bidirectional point-to-point connections between Routers
  - **Topology** = Connection Graph of Ipcores, Routers and Links
- **Scalable on-chip communication for Chip Multiprocessor (CMP) Systems**

Introduction – Networks-on-Chip



- **XY-Routing, Wormhole Switching, REQ/ACK Flow Control, Input Buffering and Round Robin Arbitration**

Introduction – Why Traffic Monitoring?

- **Increasing parallelism** in CMP (64, 128, 256, 512, … Ipcores)
- **Communication becomes dominant** for performance and energy consumption
- **Runtime-based management** mechanisms **need** current **traffic information**
  - Application Mapping / Workload Management
  - Adaptive Routing / Traffic Management
  - Application Profiling / Debugging
  - Wear-out and Degradation Management
- **To achieve cooperative interaction of these mechanisms a common information base is needed**

Outline

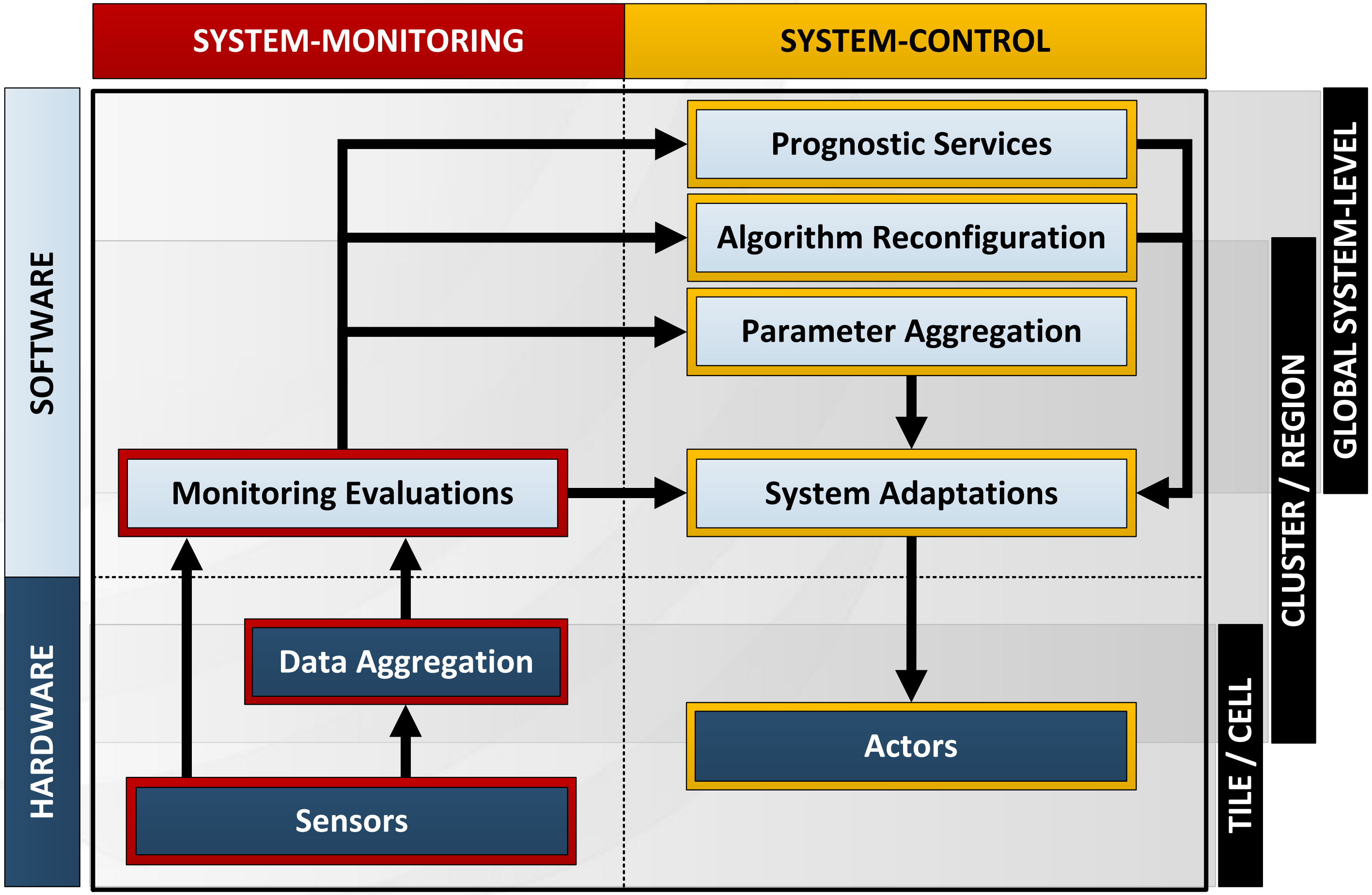
- Introduction
  - Networks-on-Chip (NoC)
  - Why Traffic Monitoring?
- **Traffic Monitoring**
  - Basic Concept
  - Dual NoC Infrastructure
  - Hardware/Software-based Clustering
  - Sensing and Flow
  - Experimental Results
- **Outlook & Future Work**

Traffic Monitoring – Basic Concept

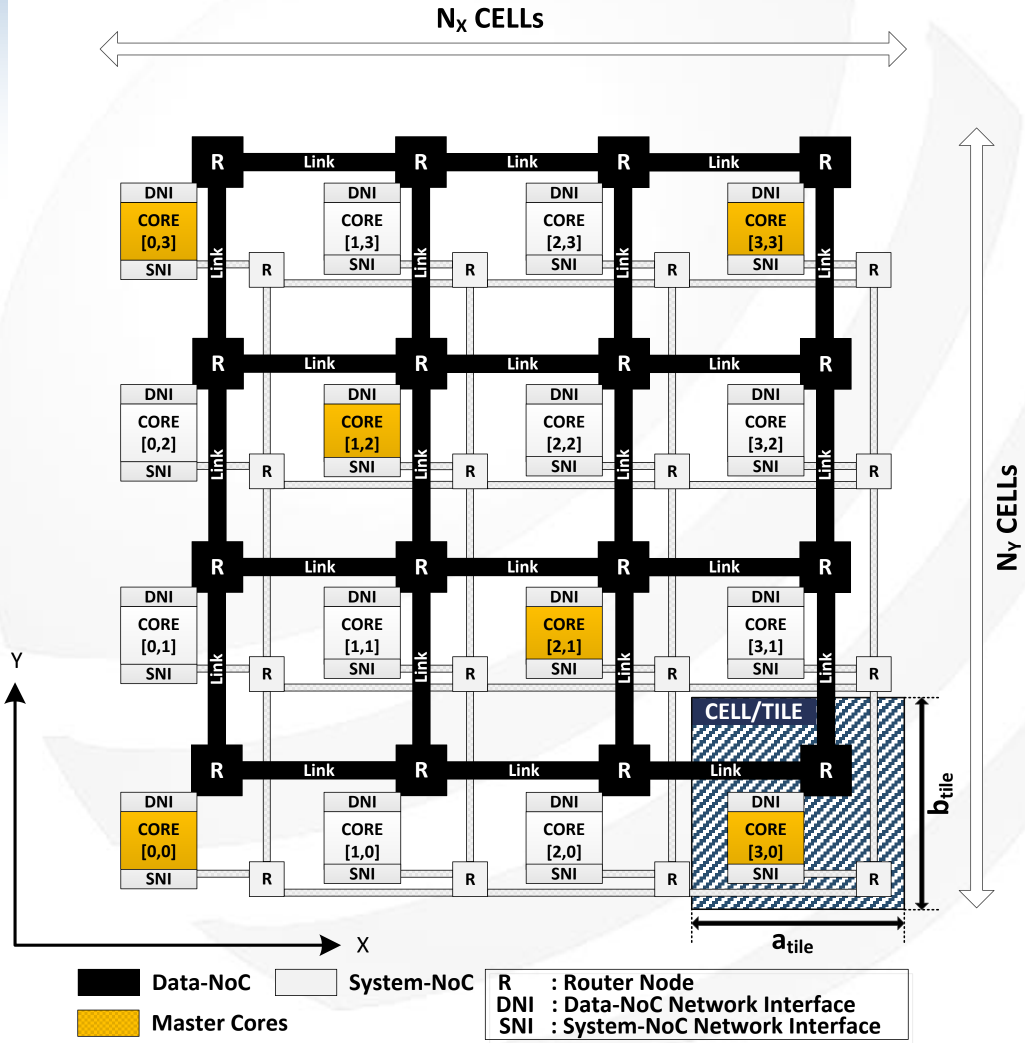
- **Design criteria:**
  - Monitoring inside a Region of Interest → Isolated workload fractions
  - Monitoring with a Resolution of Interest → Timing & Accuracy
  - Reuse of Information/Infrastructure → Mapping, Routing, Profiling
- **Flexible hardware/software solution**
  - Counter-based activity sensing/aggregation in hardware → Simple Events
  - Monitoring data evaluation and management in software
- **Centralized collection of all link- and pathloads inside a defined Region**
  - All **loads scaled to 0-100%** with resolution of 1, 2 or 4% in hardware (ready to use)
- **Adjustable timing for each Region/Cluster**
  - 1000 to 4000000 clock cycles for full data set

Traffic Monitoring – Basic Concept

- **Research scope → More software-defined and cooperative runtime optimization**

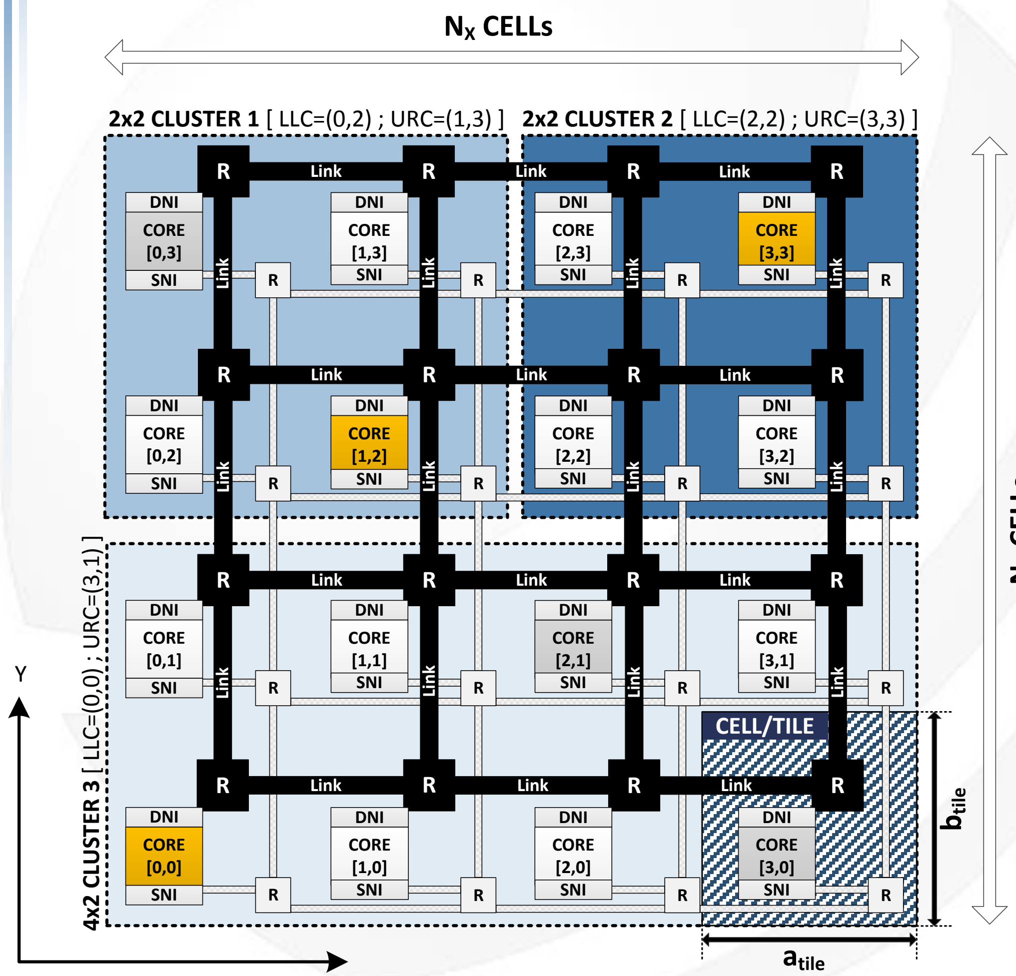


Traffic Monitoring – Dual NoC Infrastructure



- **Data-NoC** for application data
- **System-NoC** for monitoring
- **Minimal** design and runtime **interferences**
- **IP-Cores now has two NIs**
  - Data-NoC Interface (DNI)
  - System-NoC-Interface (SNI)
- **Smallest management unit is the CELL/TILE**
- **Two types of CELLs**
  - Master (CPU)
  - Slave
- **Full reuse!**

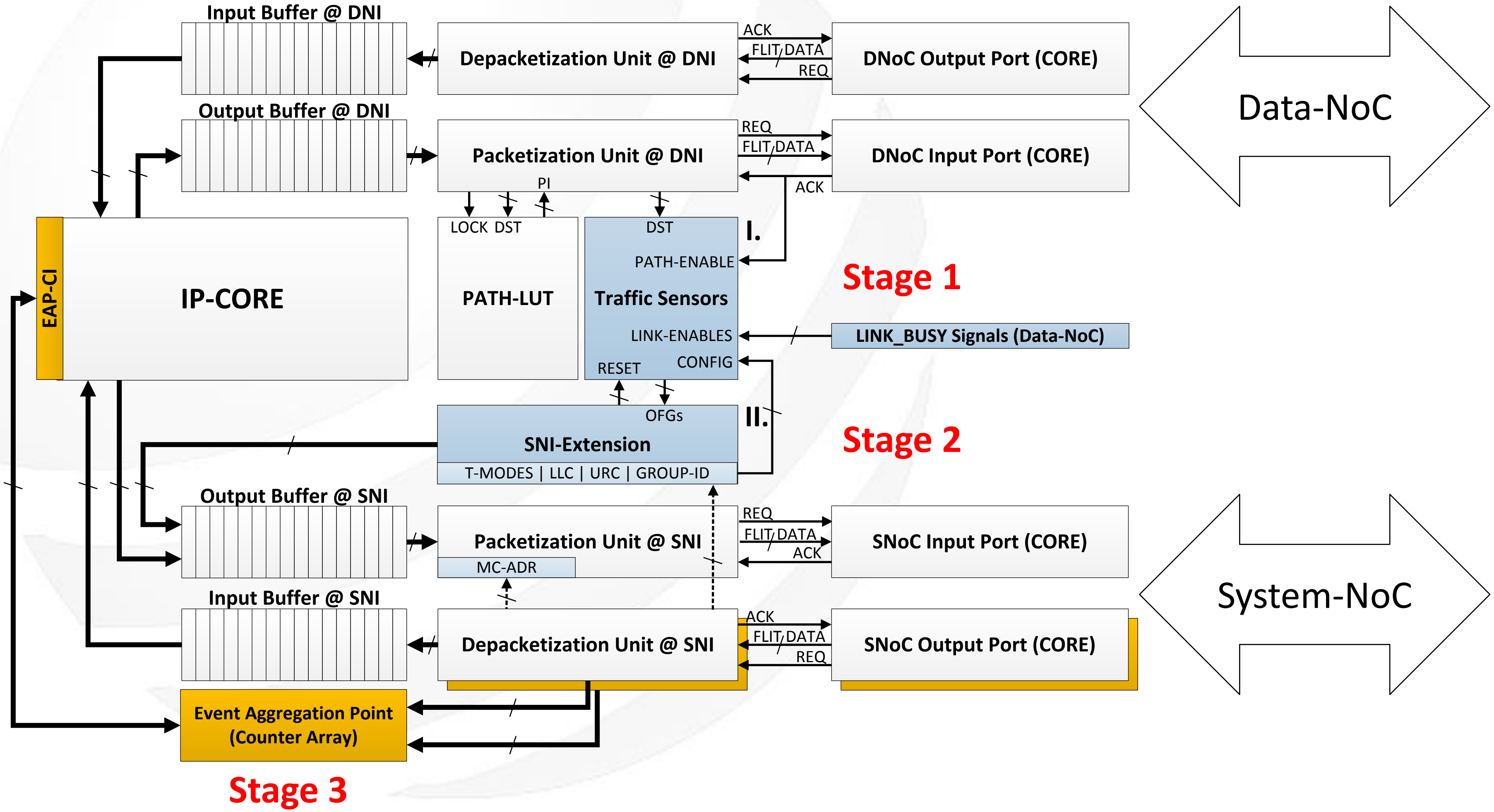
Traffic Monitoring – HW/SW-based Clustering



- **Rectangular shaped group of CELLs**
  - Lower Left Corner (LLC)
  - Upper Right Corner (URC)
  - At least one Master CELL
- **No overlapping** of Clusters
- One Master CELL per Cluster runs monitoring software
- **Each CELL monitors own loads**
  - Cluster paths
  - Router links
- **Hardware extensions at each CELL needed**
- **Maximum size is N_{CLmax} [14, 64]**

Universität Rostock Traditio et Innovatio

Traffic Monitoring – Sensing and Flow



Traffic Monitoring – Sensing and Flow

- **Hierarchical aggregation structure of 3 interacting stages:**
  - **Stage 1 – Traffic Sensors:** Activity sensing of links and paths at CELLs
    - Counting of activity clock cycles as basic events
    - **N_{CLmax} Path Sensors** uses ACK-signal of DNI for destinations inside region → **REAL LOAD!**
    - **5 Link Sensors** uses lock signals of arbiter device → **REAL LOAD + CONGESTION!**
    - Counting of active clock cycles until overflow happens at counter
  - **Stage 2 – SNI-Extension:** Periodic checking and reporting at CELLs
    - Finite state machine periodically checks Traffic Sensors for overflows → **Sensor Check Period**
    - Generates and transmits monitoring packet to Master if overflows happened
  - **Stage 3 – Event Aggregation Point:** Master counts reported overflows
    - Each Traffic Sensor has a corresponding overflow counter at the Event Aggregation Point (EAP)
    - Event Aggregation Point only implemented at Master CELLs
    - Periodic access of counter values via Core Interface (CI) → **Monitoring Cycle**

Traffic Monitoring – Sensing and Flow

- **Stage 1 – Traffic Sensors**

Activity Timing in Clock Cycles

$$t_{i+1} = t_i + t_{clk}$$

$$cnt_j(t_{i+1}) = \begin{cases} cnt_j(t_i)+1 & (s_{ctrl}=1) \wedge cnt_j(t_i) < b_{T-MODE} \\ cnt_j(t_i) & (s_{ctrl}=0) \\ 0 & (s_{ctrl}=1) \wedge cnt_j(t_i) = b_{T-MODE} \end{cases}$$

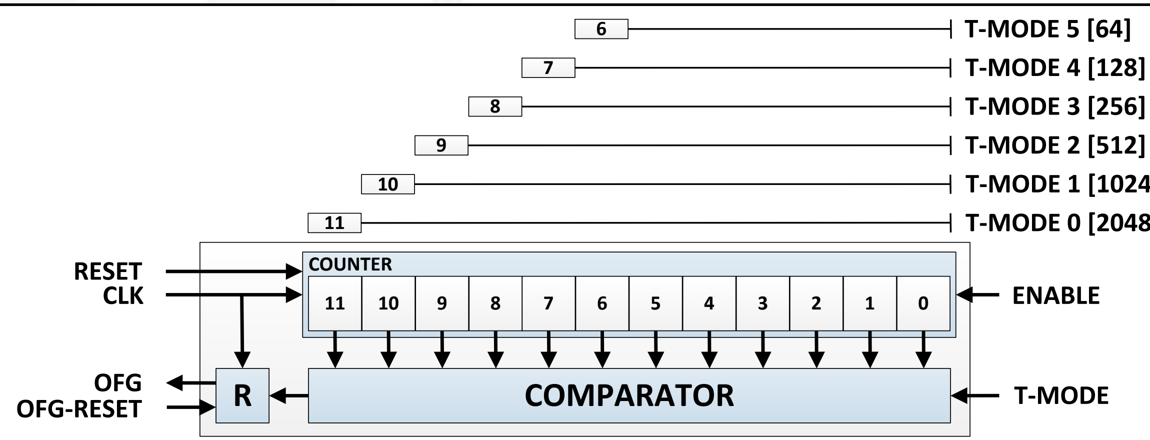
COUNTER

COMPARATOR

$$OFG(t_{i+1}) = \begin{cases} 1 & (s_{ctrl}=1) \wedge cnt(t_i) = b_{T-MODE} \\ 0 & (rst=1) \\ OFG(t_i) & else \end{cases}$$

Implementation:

- 12-bit counter
- 12-bit comparator
- ENABLE = s_{ctrl}
- Six $b_{T\text{-}MODE}$ [64 – 2048]
- OFG = Overflow bit
- $j < (N_{CLmax} + 5)$ per CELL



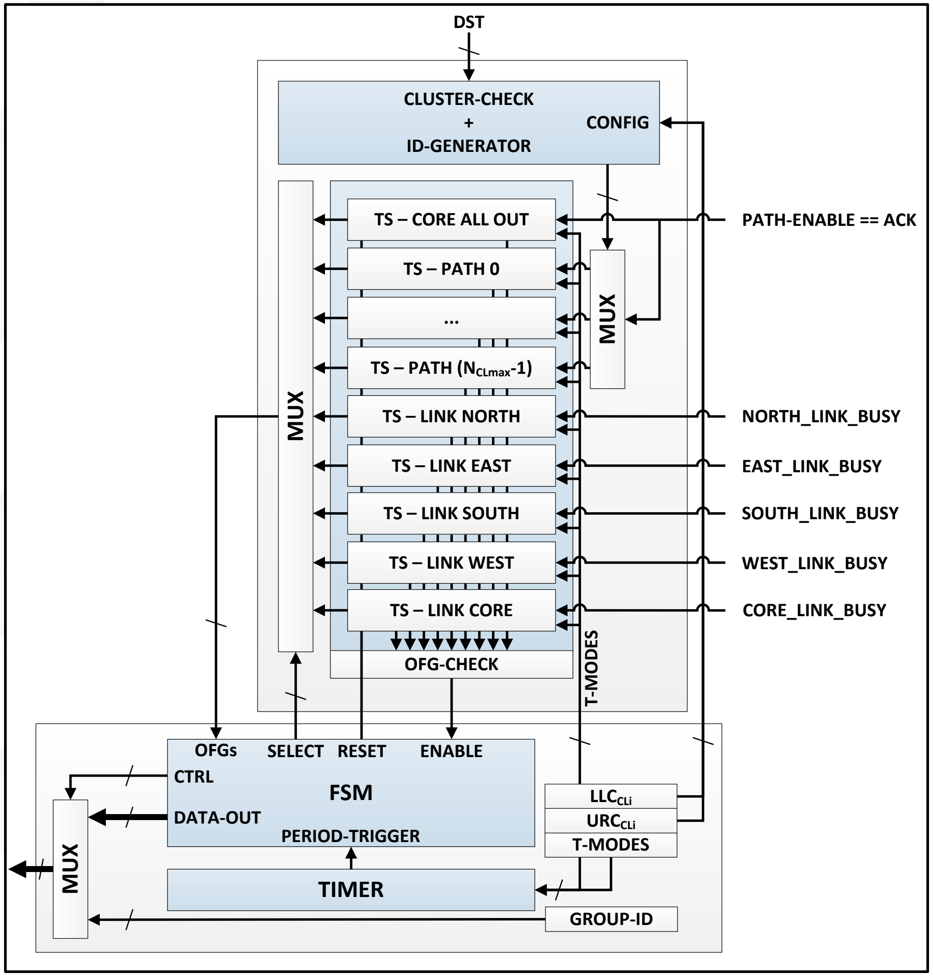
Traffic Monitoring – Sensing and Flow

- **Stage 2 – SNI-Extension**

FSM: Sensor Check Period

$$t_{sp} = b_{T-MODE} \cdot t_{clk}$$

$$check(t_{sp}) = \sum_{j=0}^{N_S-1} OFG_j(t_{sp}) > 0$$



Implementation:

- Check via Xoring all OFG
- Read & Reset all OFG
- Packet composition via FSM
- $b_{T\text{-}MODE}$ set during Cluster Setup
- $b_{T\text{-}MODE}$ equal at all Cluster CELLs

Traffic Monitoring – Sensing and Flow

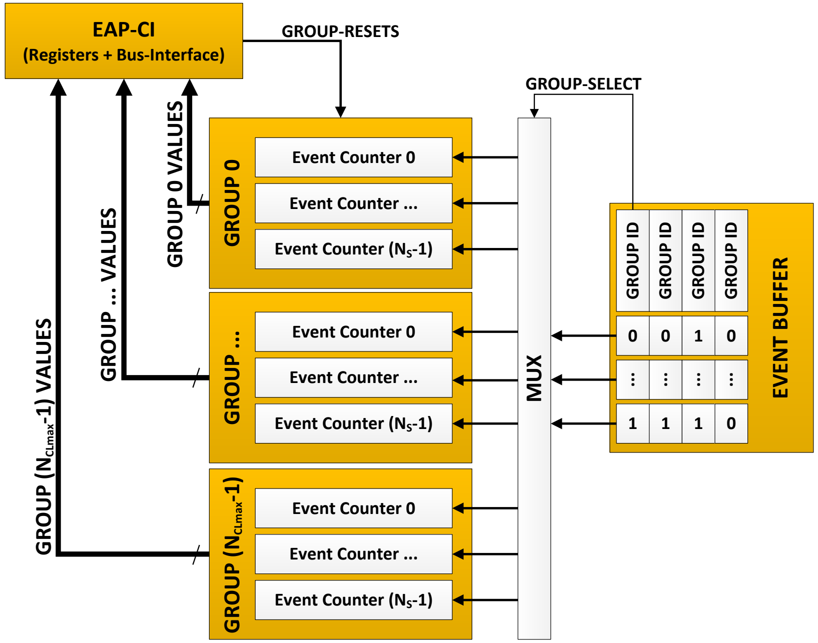
- **Stage 3 – Event Aggregation Point**

$$load_j(t_{sp+1}) = \begin{cases} load_j(t_{sp})+1 & (OFG_j(t_{sp+1})=1) \wedge (rst=0) \\ load_j(t_{sp}) & (OFG_j(t_{sp+1})=0) \wedge (rst=0) \\ 0 & rst=1 \end{cases}$$

Unscaled Loads!

Implementation:

- 7-bit counter GROUPs
- (N_{CLmax} + 5) counter per GROUP
- N_{CLmax} GROUPs (for each CELL)
- N_{CLmax} = [16, 64] analysed
- Event Buffer stores incoming packets
- Vector order = Counter order
- Periodic read and reset by software
- Intermediate read by software
- Direct Access via Core Interface (CI)
- Scaling of loads via Access Timing!



Traffic Monitoring – Sensing and Flow

- **Load Scaling via Access Timing**
  - Stepping k_s={1,2,4} → sload=0:k_s:100 in % of max BW per Link/Path
  - N_{SP} = # of Sensor Periods $t_{sp}=f(t_{clk}, b_{T\text{-}MODE})$ per Monitoring Cycle
  - T_{MC} = Monitoring Cycle

$$N_{SP} = 100 / k_s$$

$$T_{MC} = N_{SP} \cdot t_{sp}$$

$$sload_j = \begin{cases} load_j & k_s = 1 \Rightarrow N_{SP} = 100 \\ load_j << 1 & k_s = 2 \Rightarrow N_{SP} = 50 \\ load_j << 2 & k_s = 4 \Rightarrow N_{SP} = 25 \end{cases}$$

Scaled Loads!

- **min($b_{T\text{-}MODE}$)=f(Clustersize)**
  - Traffic Monitoring → Many-to-One Pattern
  - **Condition: (Injected BW in Cluster < Receivable BW at Master)**
    - **16 CELLs → min($b_{T\text{-}MODE}$) = 128**
    - **64 CELLs → min($b_{T\text{-}MODE}$) = 1024**
  - **If condition met → sload ± e_{max} with $e_{max} \le 2 \cdot k_s$!!!**

Traffic Monitoring – Sensing and Flow

- **Timing of Traffic Monitoring offers two options:**
  - Adjustment of $b_{T\text{-}MODE}$ → At all Cluster CELLs
  - Adjustment of Stepping k_s → At Master CELL only
  - Tradeoff: Effort vs. Accuracy

	Monitoring Cycle T_{MS} [µs] for t_{clk}=1ns		
$b_{T\text{-}MODE}$	k_s=1	k_s=2	k_s=4
64	6.4	3.2	1.6
128	**12.8**	**6.4**	**3.2**
256	25.6	12.8	6.4
512	51.2	25.6	12.8
1024	**102.4**	**51.2**	**25.6**
2048	204.8	102.4	51.2

Traffic Monitoring – Experimental Results

- 45nm hardware synthesis via Synopsys Design Compiler
  - Estimation of hardware overhead
- SystemC-based cycle accurate NoC simulation
  - Measurement of absolute traffic monitoring error (max & avrg)

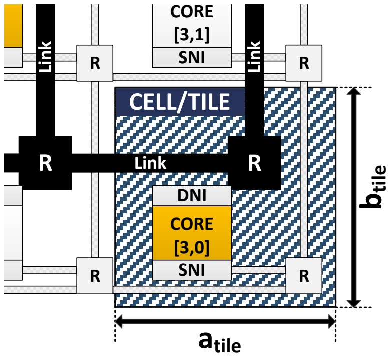
Parameter	Value	
NoC	SNoC	DNoC
Linkwidth w_L	8-, 16-bit	64-bit
Clock Rate t_{clk}	1ns	
Port Buffer Depth	1 flit	5 flit
$b_{T\text{-}MODE}$	64 – 2048 (Simulation: 128, 1024)	
Cluster Size N_{CLmax}	16 and 64	
Cluster Shape	4×4, 8×2 and 16×4, 8×8	
Monitor Position	Lower Left Corner (LLC)	
Technology	45nm (Nangate FreePDK45)	

Traffic Monitoring – Experimental Results

- **Hardware Overhead of the Traffic Monitoring in 45nm**
  - A_{normal} = Area($Router_{SNoC}$, $Links_{SNoC}$, Traffic Sensors, SNI-Extension)
  - A_{master} = Area(A_{normal}, EAP)
  - A_{tile} = 3mm x 3mm (estimate 45nm Intel SCC)

Total Area Overhead per TILE @ 45nm for t_{clk}=1ns, N_X=8 and N_Y=8				
N_{CLmax}	16 CELLs		64 CELLs	
SNoC Linkwidth	8-bit	16-bit	8-bit	16-bit
A_{master}/A_{tile}	0.51%	0.66%	3.11%	3.26%
A_{normal}/A_{tile}	0.29%	0.44%	0.37%	0.52%

- **EAP and Traffic Sensors dominate the hardware overhead**
- **But area estimates remain inside a feasible range!**

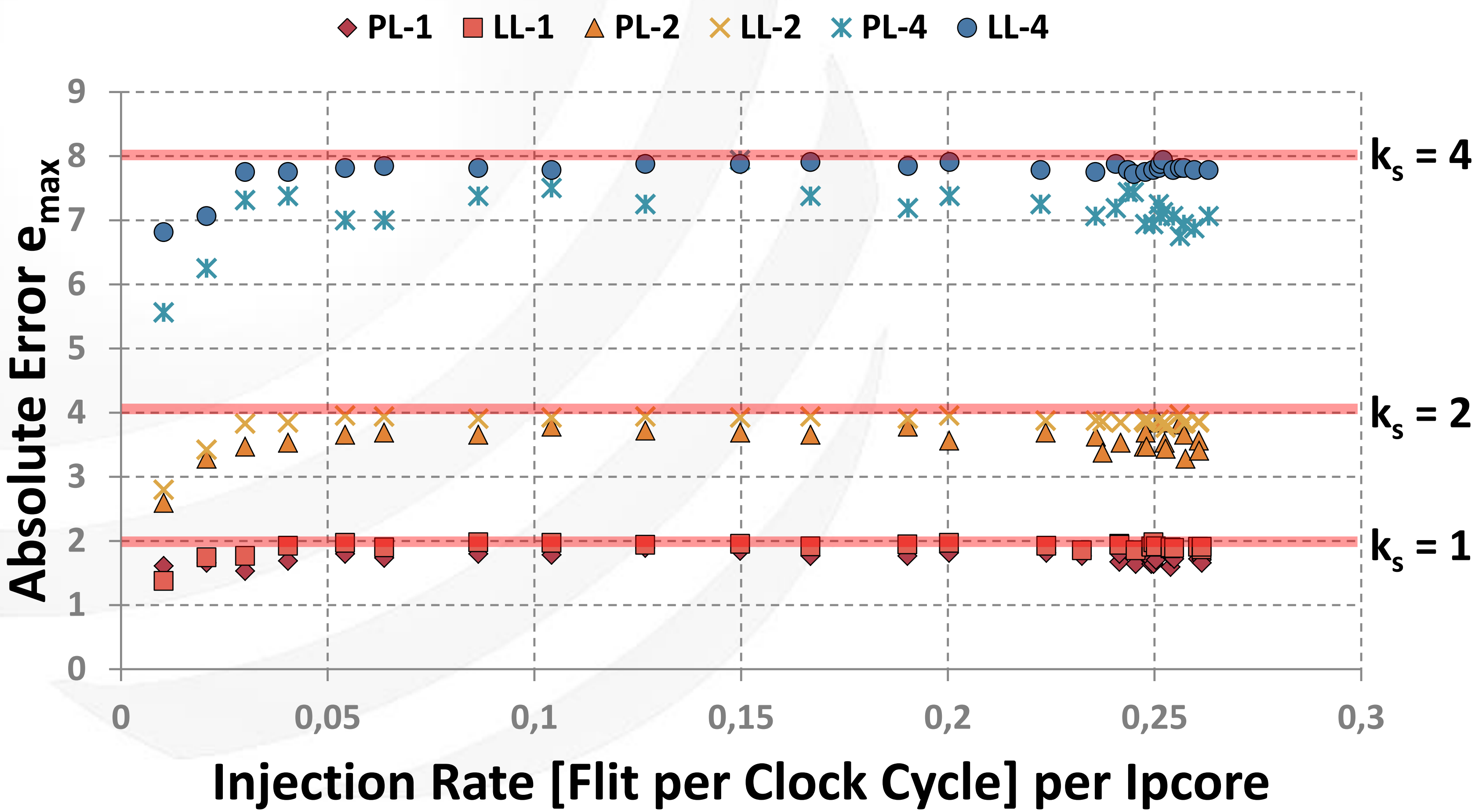


Traffic Monitoring – Experimental Results

- Full system simulation with mixed workloads
  - Random Uniform Traffic Pattern
    - Balanced loads and high overall utilization
    - Packetsize = rand(5,15) Flit
    - Destination = rand(0, $N_x \cdot N_y$ - 1)
  - Random Task Graphs with sequential mapping
    - 7 to 70 Tasks per Graph
    - 2 to 10 Graphs per Workload
    - Packetsize = rand(5,50) Flit
  - Each Workload simulated 10 times with 10 full Monitoring Cycles for each Cluster shape (4x4, 8x2, 8x8, 4x16) and stepping (1, 2, 4)
  - Logging of average and maximum errors for pathloads (PL) and linkloads (LL)

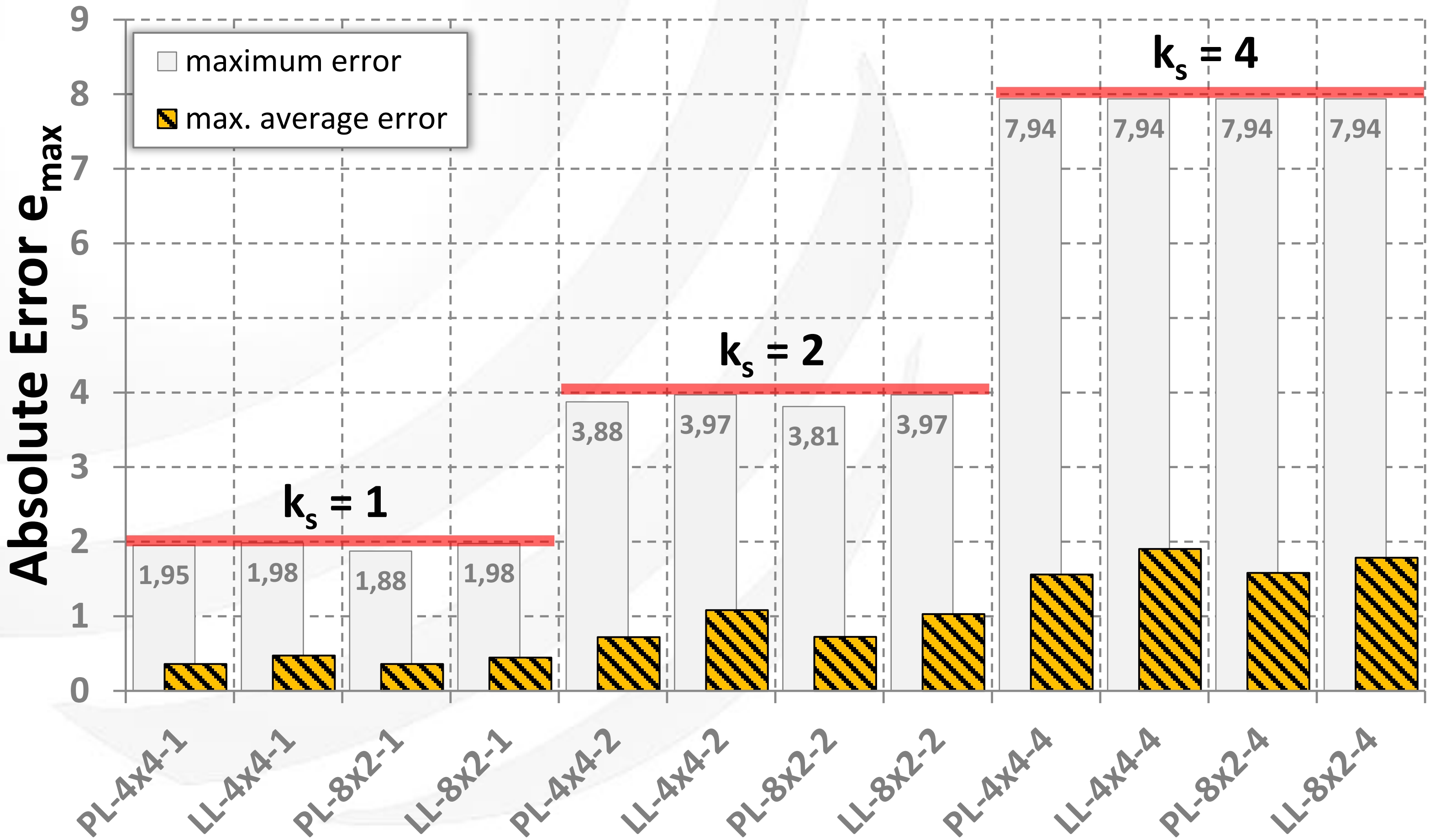
Traffic Monitoring – Experimental Results

- Max. error in 4x4 CELL Cluster @ Random : $\min(b_{T\text{-}MODE}) = 128$



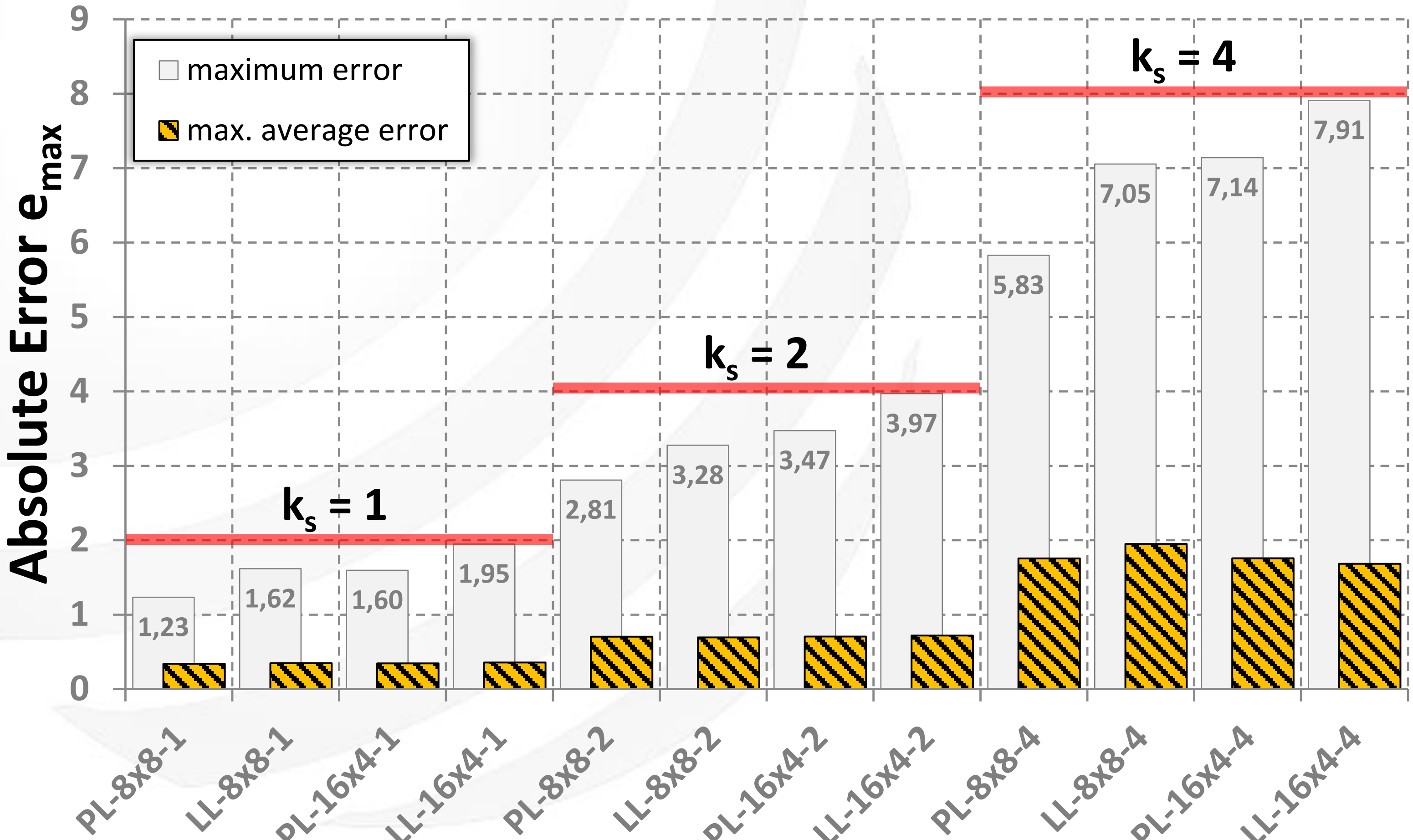
Traffic Monitoring – Experimental Results

- Max. Error all 16 CELL Cluster Scenarios: $\min(b_{T\text{-}MODE}) = 128$



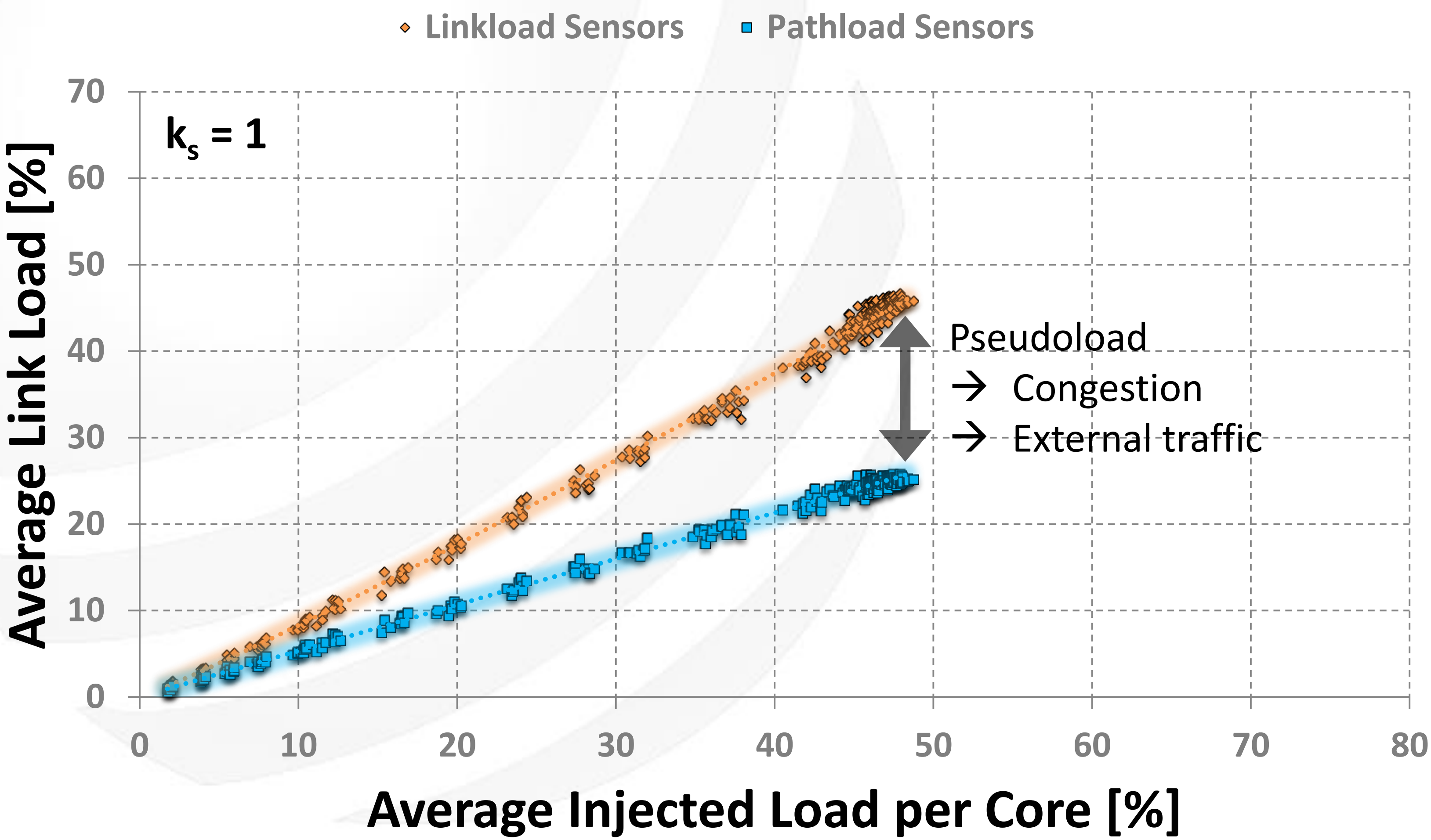
Traffic Monitoring – Experimental Results

- Max. Error all 64 CELL Cluster Scenarios: $\min(b_{T\text{-}MODE}) = 1024$



Traffic Monitoring – Experimental Results

- Pseudoload 4x4 CELL Cluster Scenarios: min($b_{T\text{-}MODE}$) = 128



Outline

- **Introduction**
  - Networks-on-Chip (NoC)
  - Why Traffic Monitoring?
- **Traffic Monitoring**
  - Basic Concept
  - Dual NoC Infrastructure
  - Hardware/Software-based Clustering
  - Sensing and Flow
  - Experimental Results
- **Conclusion & Future Work**

Conclusion & Future Work

- **Flexible HW/SW traffic monitoring proposed**
  - All path- and linkloads inside a resizable region at a single entity
  - Adjustable timing and accuracy
  - Hardware overhead and achievable Monitoring Cycles are feasible
- **Next steps and investigations → Runtime Mechanisms**
  - Workload/Application Profiling (Rent‘s Rule)
    - Communication Distributions/Probabilities
    - Execution Phase Detection
    - Combination with Performance Counter Data
  - Flow-based Traffic Management
    - Path adaptations inside Clusters

THANK YOU!