KALRAY

# Guaranteed Services on the Network-on-Chip of a Manycore Processor

**Duco van Amstel**

**Benoît Dupont de Dinechin**

**RTAS 2014 Industrial Session**

# About Manycore Computing

- Manycore versus Multicore
  - A multicore processor has a 4 – 16 cores that share the memory hierarchy (cache, on-chip, off-chip)
    - Intel Core series, ARM Cortex-A series, Freescale P4080
  - A manycore processor has > 32 cores, implying some memory is close and most memory is far from each core
    - Intel Xeon-Phi, GP-GPU (NVIDIA, AMD), Tilera Tile-GX
- Challenges of Manycore
  - Difficult programming models, hard to reach performance
    - OpenMP or shared memory + vector instructions (Intel Xeon Phi)
    - OpenCL or CUDA + vector instructions (GP-GPU, TI Keystone)
  - Ill-suited to embedded systems
    - Unstable or unpredictable or unverifiable response times
    - High power consumption, high energy per operation

# Kalray MPPA®-256 Processor with CMOS 28nm TSMC

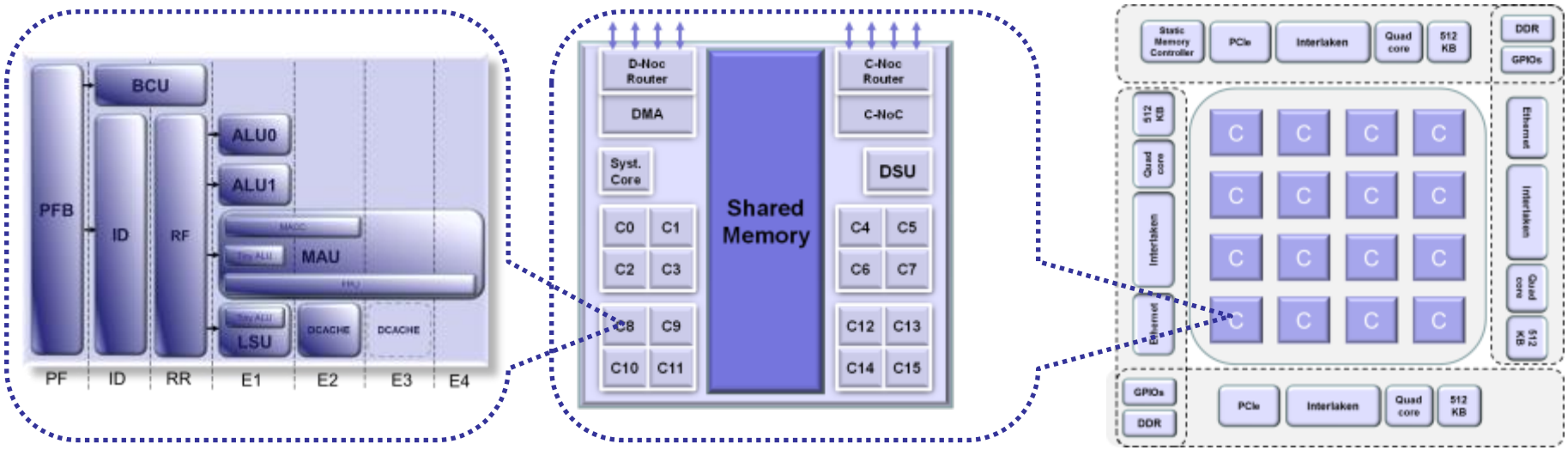
## 256 VLIW processing engine cores + 32 VLIW resource management cores


Shipping since January 2013

- High processing performance
  700 GOPS – 230 GFLOPS SP
- Low power consumption
  5 – 15W at 200 – 400MHz
- High execution predictability
- High-level programming models
- PCI Gen3, Ethernet 10G, NoCX

# MPPA®-256 Processor Hierarchical Architecture

VLIW Core | Compute Cluster | Manycore Processor

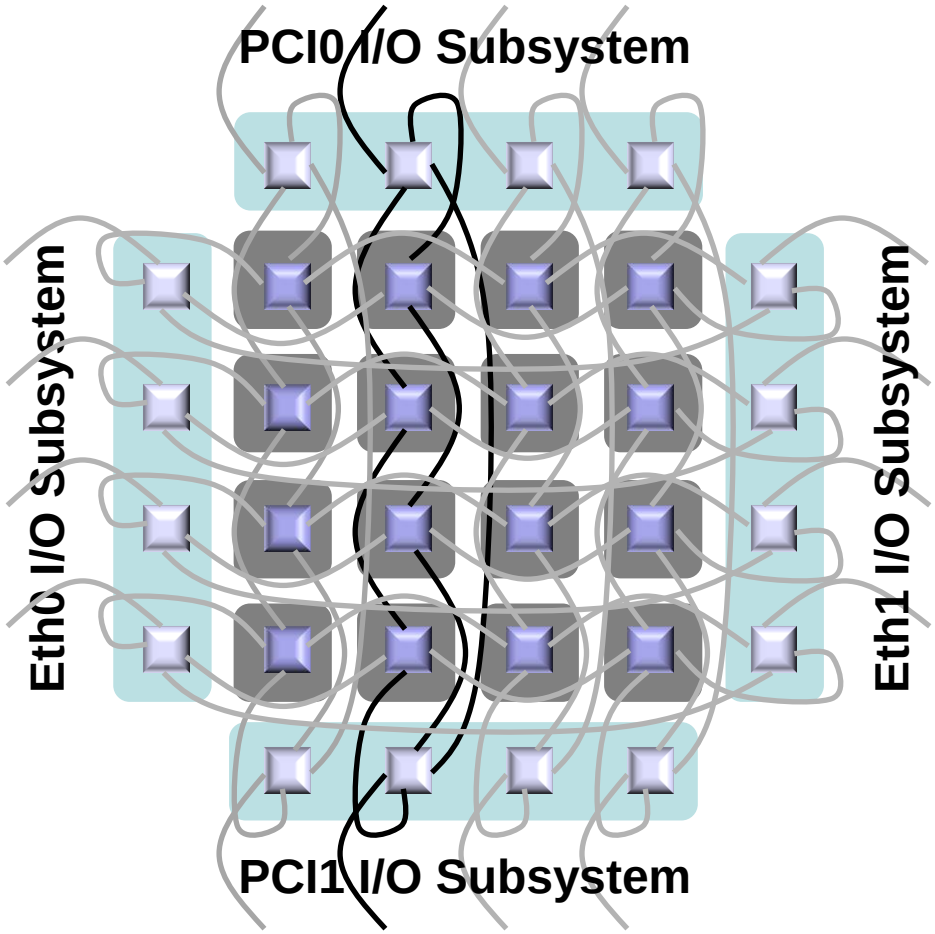



**Instruction Level Parallelism**

**Thread Level Parallelism**

**Process Level Parallelism**

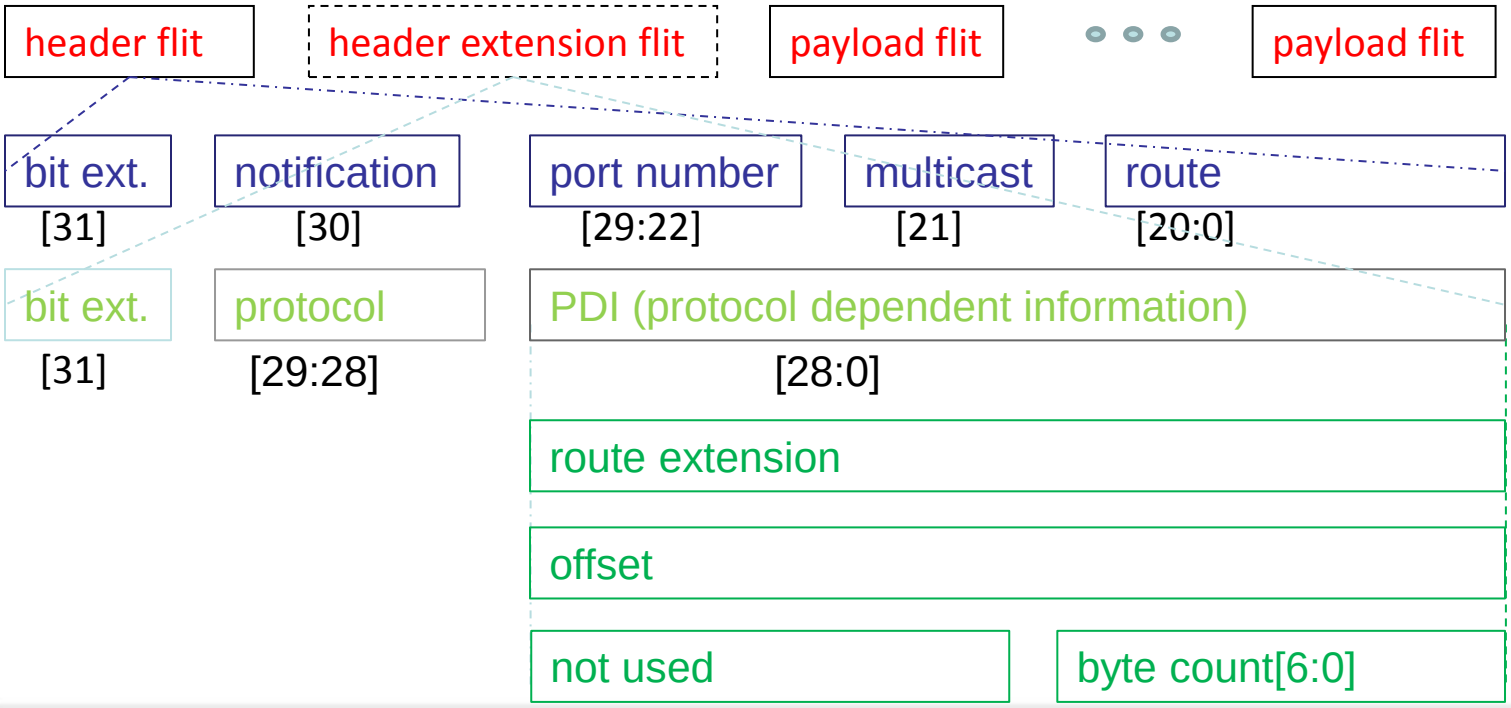
# MPPA®-256 Clustered Memory Architecture



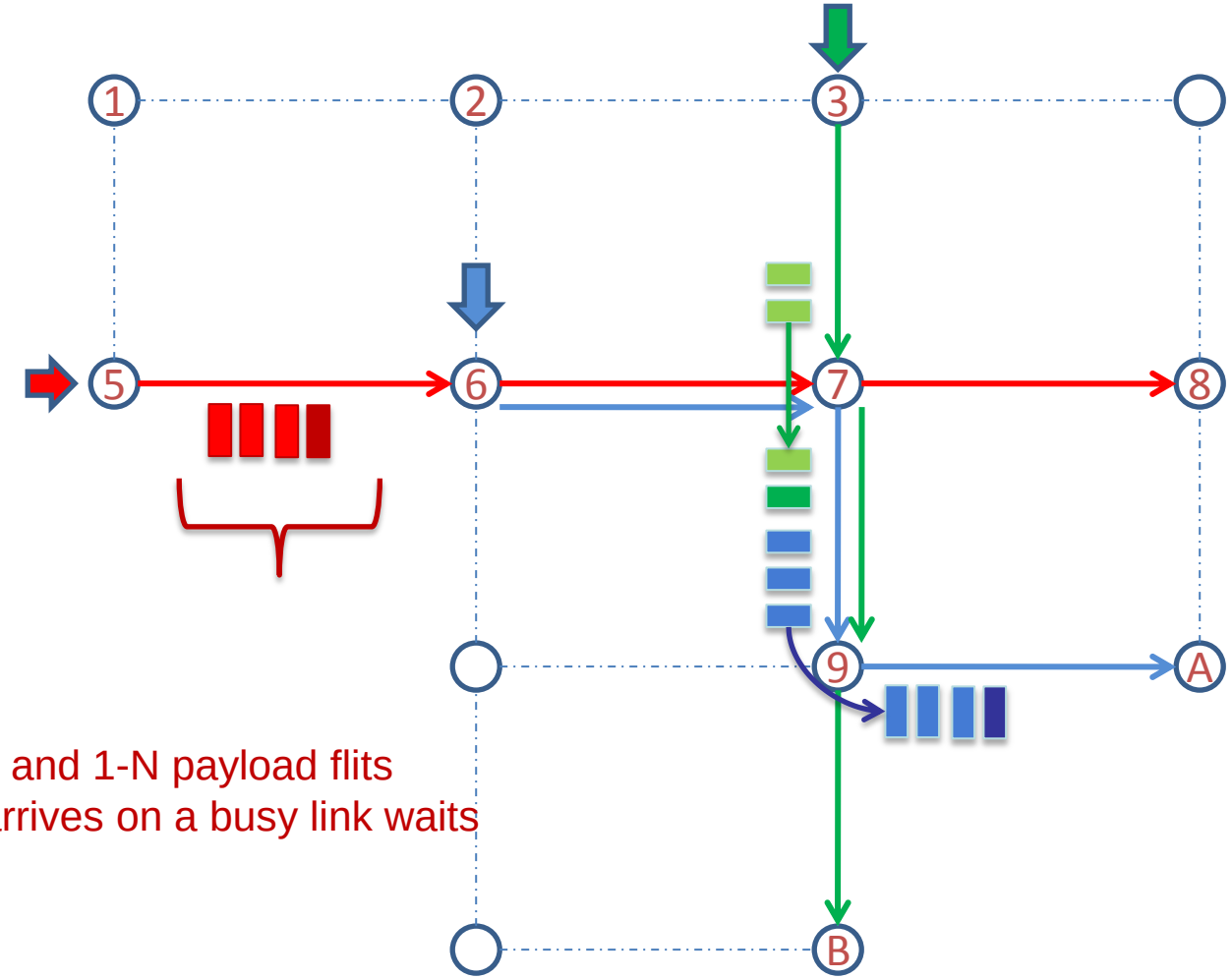

- 20 memory address spaces
  - 16 compute clusters
  - 4 I/O subsystems with direct access to external DDR3 memory
- Dual Network-on-Chip (NoC)
  - Data NoC & Control NoC
  - Full duplex links, 4B/cycle
  - 2D torus topology + extension links
  - Explicit routing at source node
  - Unicast and multicast transfers
  - Oblivious synchronization
- Data NoC guaranteed services
  - Flow regulation at source node

# MPPA® NoC Main Features

- Explicitly routed NoC with wormhole switching
  - Message is a sequence of atomically received packets
  - Each packet has a header and a payload, composed of flits
  - The header flit(s) contains the route, port, and other information



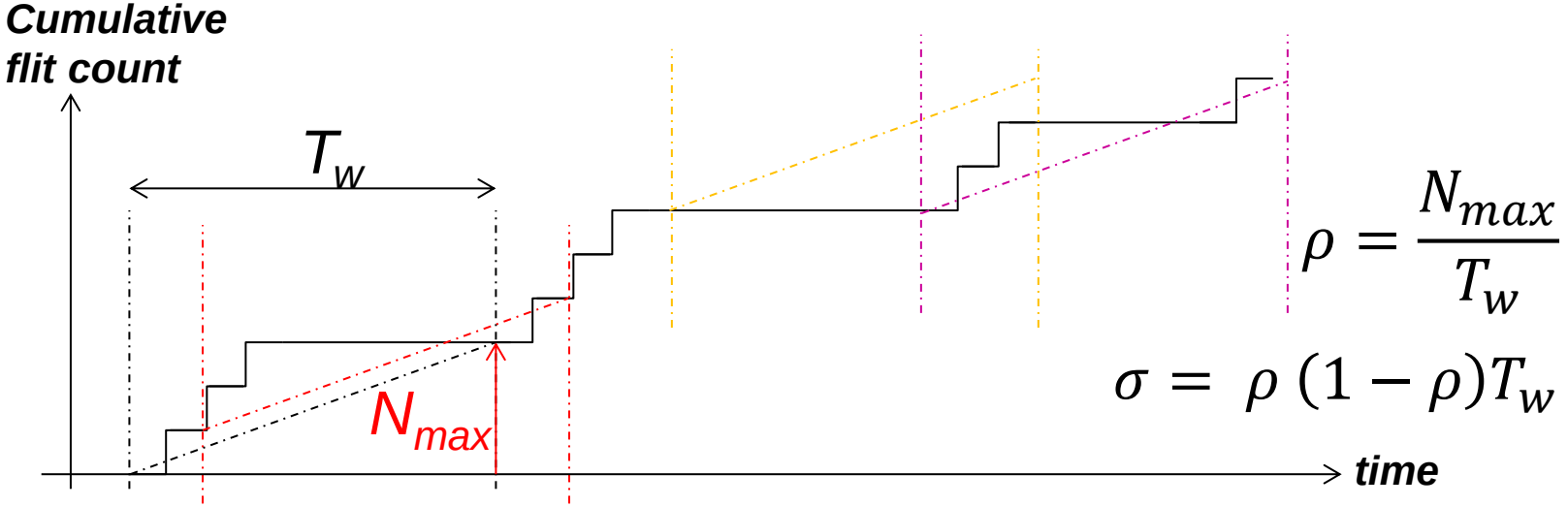
# Wormhole Switching Illustrated




One header flit and 1-N payload flits
A packet that arrives on a busy link waits

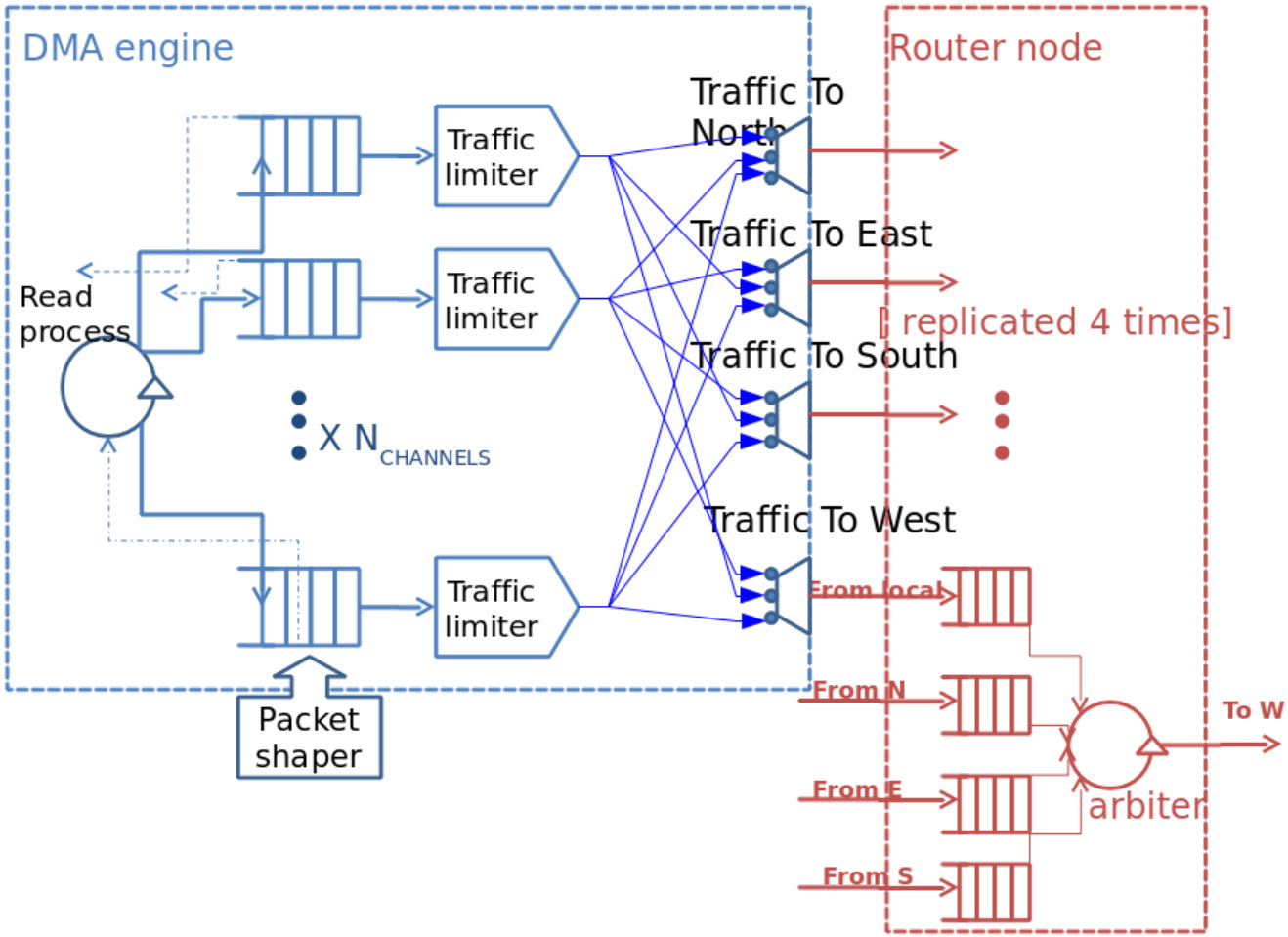
# Foundations of the MPPA® NoC Guaranteed Services

- NoC injection policy that implement a ($\sigma$, $\rho$) regulation
  - For any time interval $\tau$ the number of packets is not greater than $\sigma + \rho\tau$
  - Implemented with a 'sliding window' scheme of parameters $T_w$, $N_{max}$

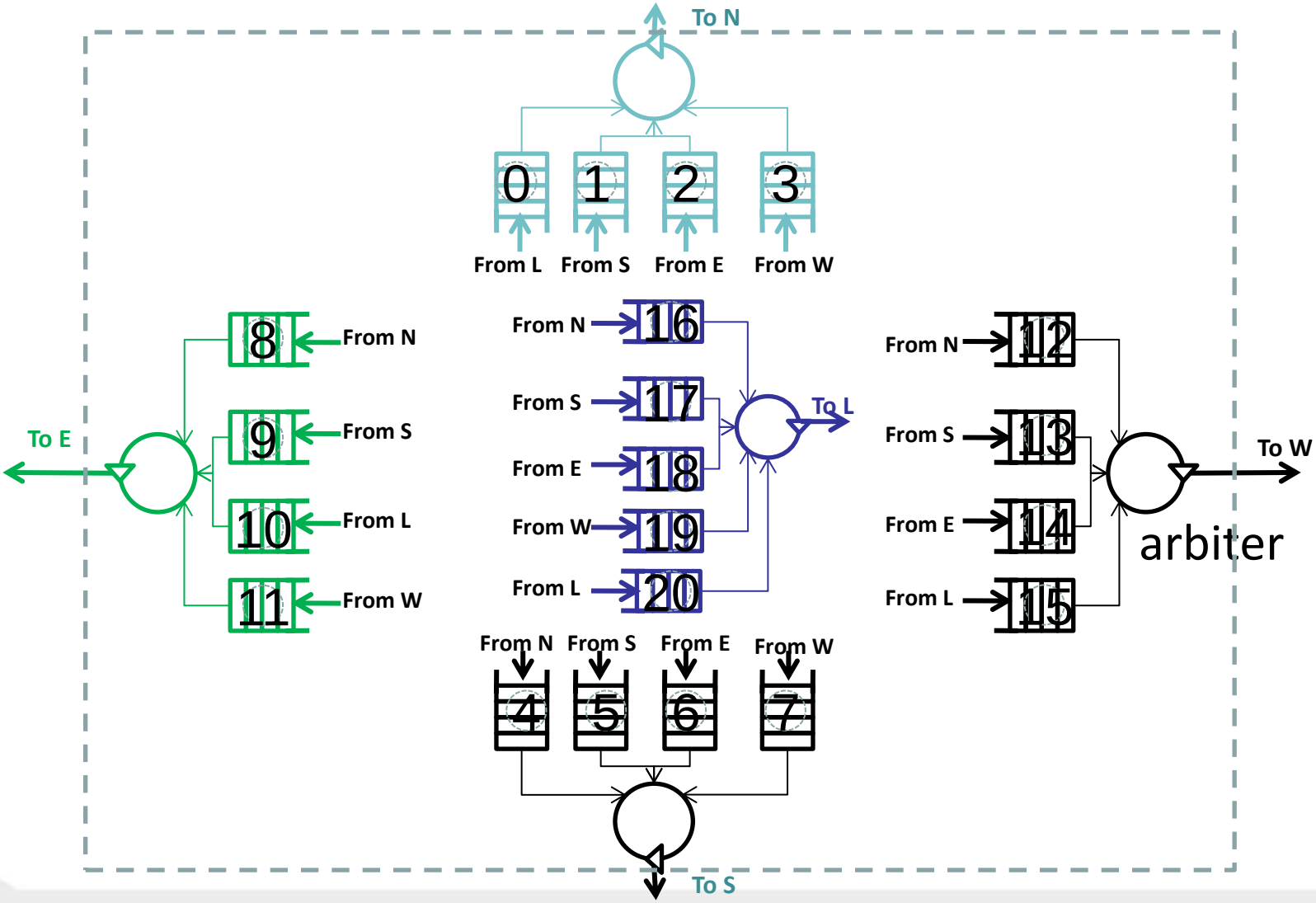



- NoC routers that only contain FIFO queues
  - Routers are 'work conserving': no idling if data ready to transmit
  - One set of queues per out-going link, with round-robin arbitration

# MPPA®-256 Data NoC Tx Model

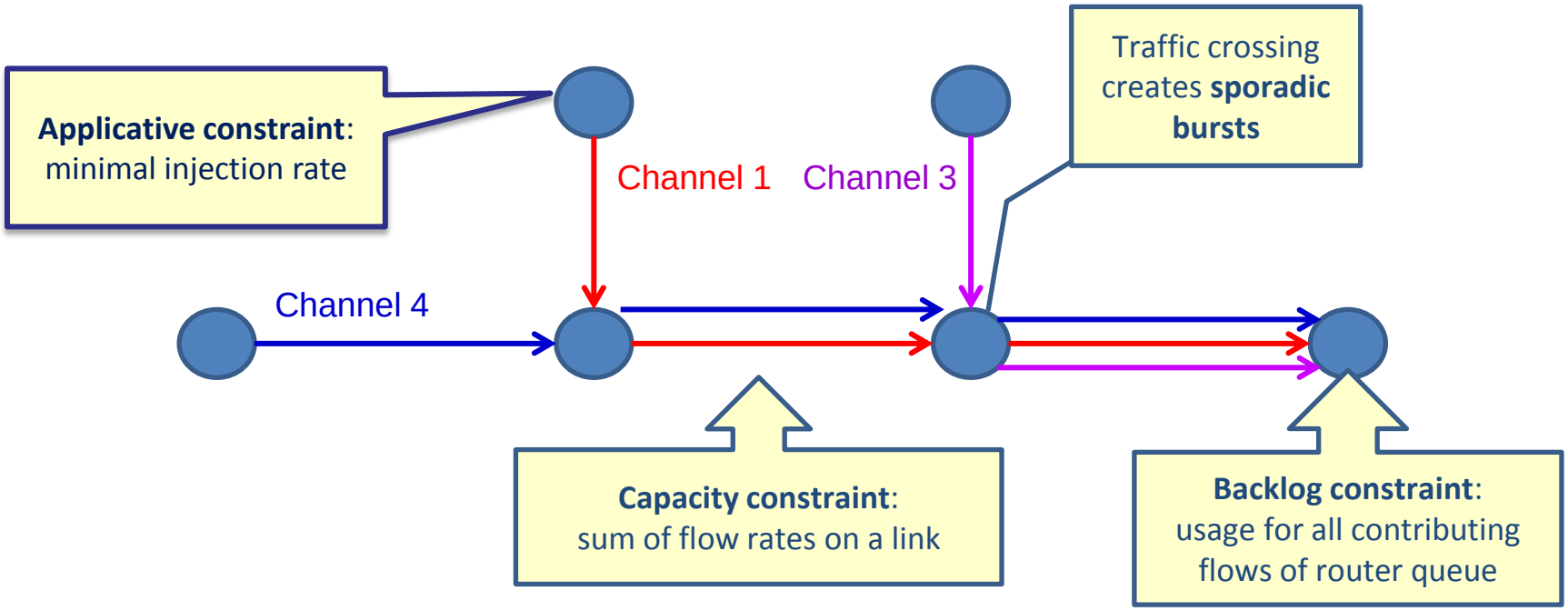


# MPPA® NoC Router Structure



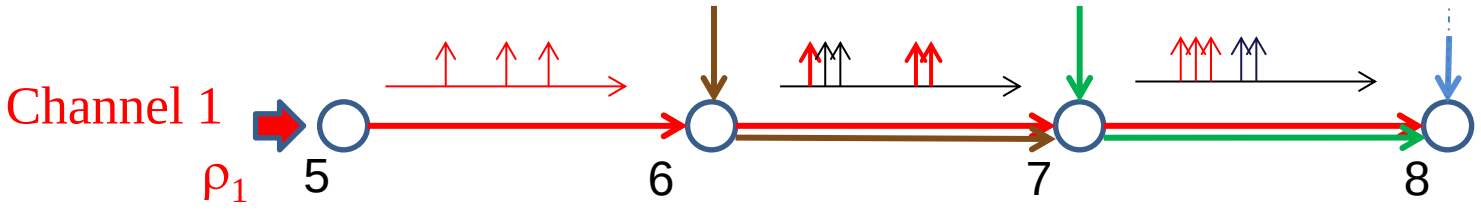
# MPPA® NoC Guaranteed Services Problem Statement

- Avoid NoC router queue overflow by configuring source ($\sigma_i$, $\rho_i$)
- On the MPPA® NoC, $\sigma$ is related to $\rho$ : $\sigma = \rho\,(1-\rho)T_w$
- The problem can be solved by only considering injection rates $\rho_i$



# MPPA® NoC Calculus Linear Constraints (1/3)

- Link capacity constraints
  - For each link traversed by a set of flows $\{ (\sigma_i, \rho_i) \} : \sum_i \ \rho_i \leq 1$
- Queue backlog constraints
  - For each queue buffering a link with flows $\{ (\sigma_i, \rho_i) \} : \sum_i \ \sigma_i \leq Q_{size}$
- Propagation of $(\sigma, \rho)$ between source and hop $k$ [Cruz 1991]
  - $(\sigma', \rho) = (\sigma + \sum_{L=1}^{L=k} \rho d_L \, , \rho)$ with $d_L = (n_L - 1)P_{size}$
  - $n_L$ is the number of directions merging to link L

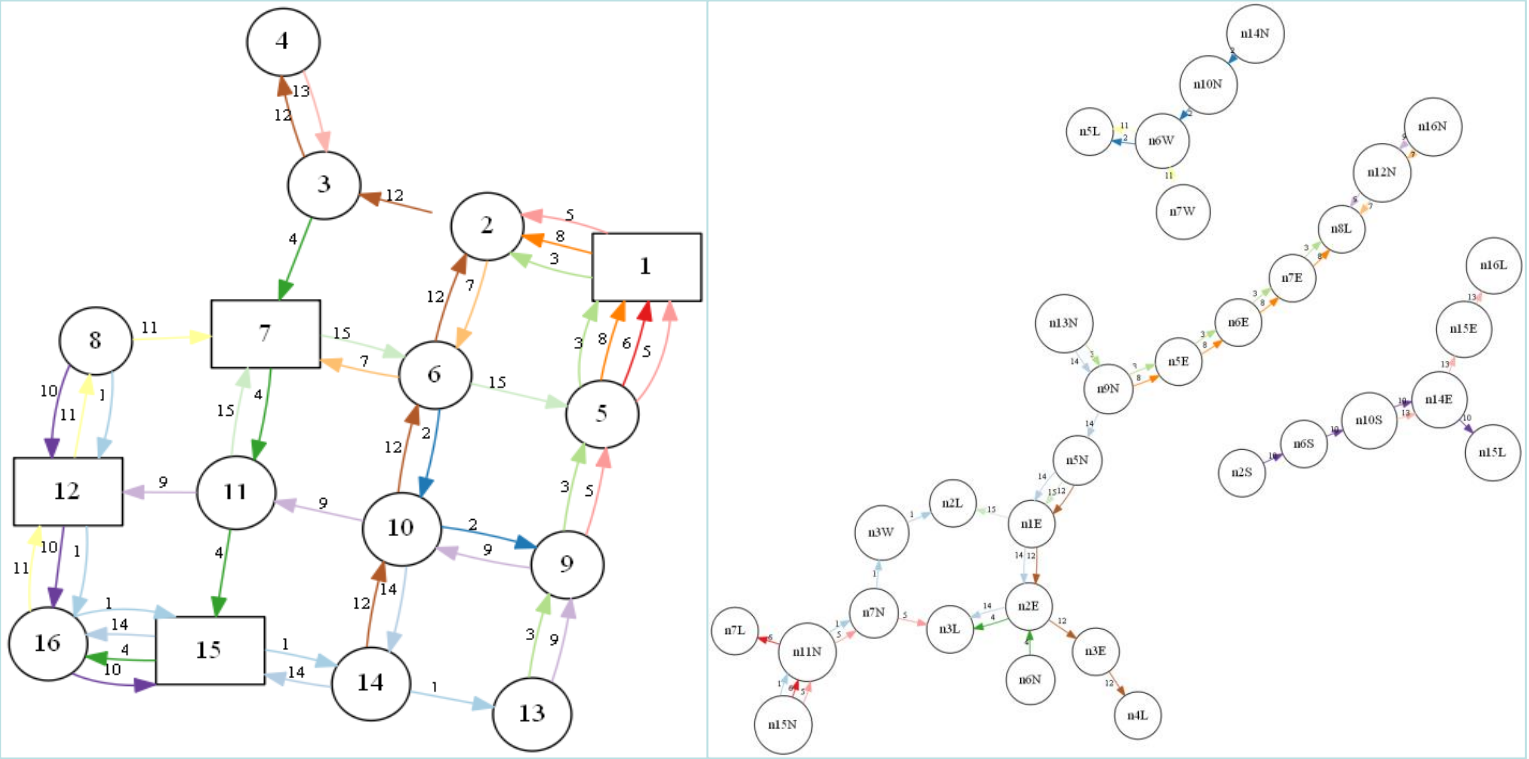


# MPPA® NoC Calculus Linear Constraints (2/3)

- Approximating non-linear term in queue backlog constraints
  - $\sum \sigma_c = \sum \rho_c(1-\rho_c)T_w \leq \frac{n_L-1}{n_L} T_w$
  - Valid because $\sum \rho_c \leq 1$ for $n_L$ flows on L
- Linearized queue backlog constraints
  - For all c in CC set of contributing channels
  - For all L in RR links traversed upstream
  - $\sum_{c \in CC} \sum_{L \in RR} d_{L,c}\rho_c \leq Q_{size} - \frac{(n_L-1)}{n_L} T_w$

# MPPA® NoC Calculus Linear Constraints (3/3)

$$[C] \times \begin{bmatrix} \rho_1 \\ \dots \\ \rho_N \end{bmatrix} \leq \begin{bmatrix} capacity\ bounds \\ backlog\ bounds \end{bmatrix}$$



# Objective function

- Non-Linear version
  - Based solely on the flow rates
  - Simple sum $F = \sum_i \rho_i$ does not give enough to small flows
  - Use *proportional fairness* instead

$$F = \sum_i \log \rho_i$$

- Linear version
  - Non-regular piece-wise linearization of the logarithm
  - 50% of pieces within $[0; 0.0001]$, 50% within $]0.0001; 1]$
  - Experimental results required less than 100 linear pieces
- Execution times
  - Linearization with regular piece-wise linearization: speed-up of 10+
  - Linearization with non-regular piece-wise linearization: speed-up of 100+

# Linear vs. Non-Linear Model : experimental results

- Solving the same instance on both models
  - Original model with quadratic constraints via SQP
  - Presented linearized version via GLPK
- One toy example and two industrial applications
  - MotionEstimation
  - STAP (Space-Time Adaptive Processing)

| | | Variance | Sum |
|---|---|---|---|
| Toy | SQP<br>GLPK | 0.027316<br>0.040702 | 1.6128<br>1.4041 |
| MotionEstimation | SQP<br>GLPK | 0.19835<br>0.215123 | 8.4326<br>8.091047 |
| STAP | SQP<br>GLPK | 0.016439<br>0.009790 | 6.2612<br>6.150907 |

# Computation of Worst Case Traversal Time

- Assume that flows with rates $\Gamma$ has been computed
  - $\rho$ : rate of virtual channel
  - $T_w$ : sliding window size
  - $P_{size}$ : packet size
  - n : number of hops
  - d : router fixed delay
- Then the worst case traversal time is [Zhang 95] :

$$t_{\max} = (1-\rho)T_w + \frac{n-1}{\rho}P_{size} + n(d + P_{size})$$

$$(1-\rho)T_w = \frac{\sigma}{\rho}$$

*Bursts incidents*

*Flow regulation*

*Packet granularity*

- More precise than summing the individual worst cases
  - Property known as « pay bursts only once »

# MPPA®-256 For Time-Critical Computing

- Architecture & Implementation
  - Fully timing-compositional VLIW cores
    - Includes LRU caches and hardware looping
  - Multi-banked local memory system without bus interferences
    - Dual address mapping: interleaved or blocked
  - Network on Chip with guaranteed services
    - Flow regulation at the source similar to AFDX
  - Low-latency local connexions
  - Deterministic Ethernet
- Programming models for time-critical applications
  - POSIX-Like programming model
    - Processes on clusters and threads on cores
    - Synchronous and asynchronous POSIX I/O with call-back
    - POSIX timers and signals