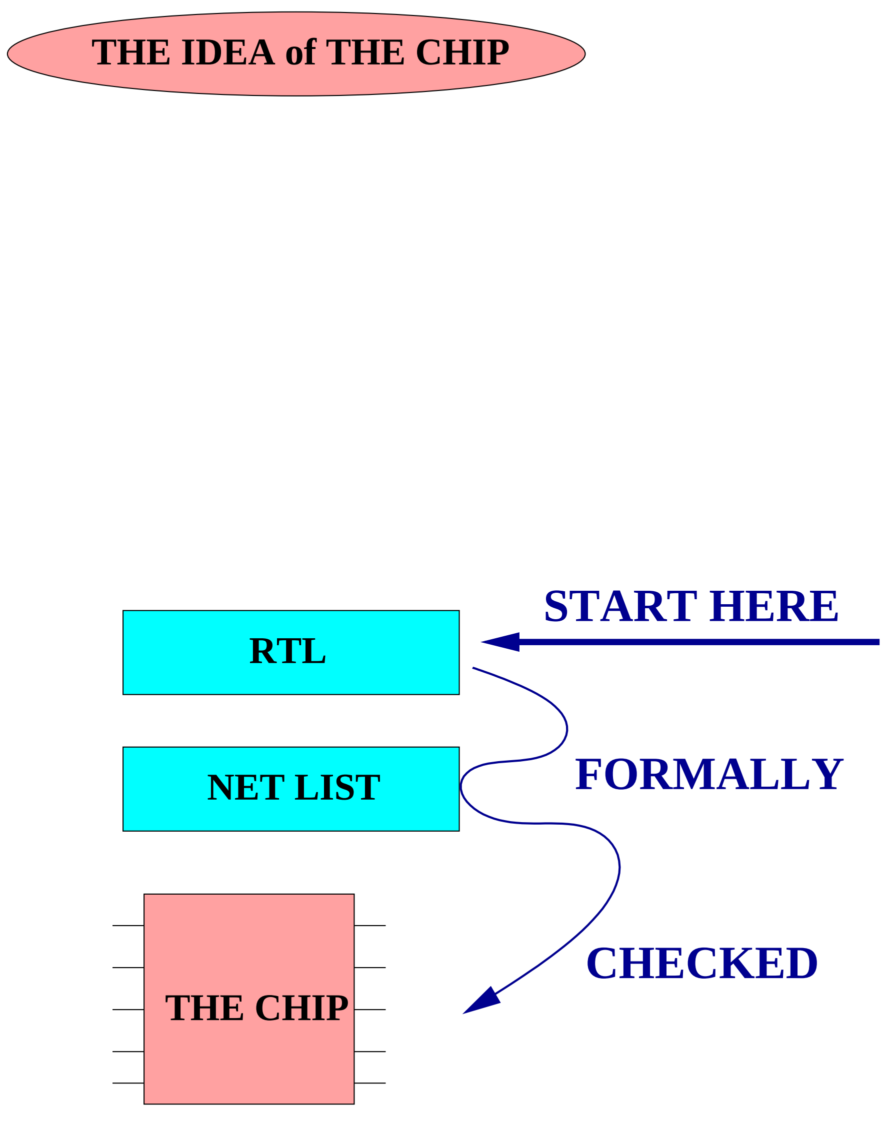
A Generic Network on Chip Model

Julien Schmaltz and Dominique Borrione

TIMA Laboratory - VDS Group

46, av. Felix Viallet 38031 Grenoble, France

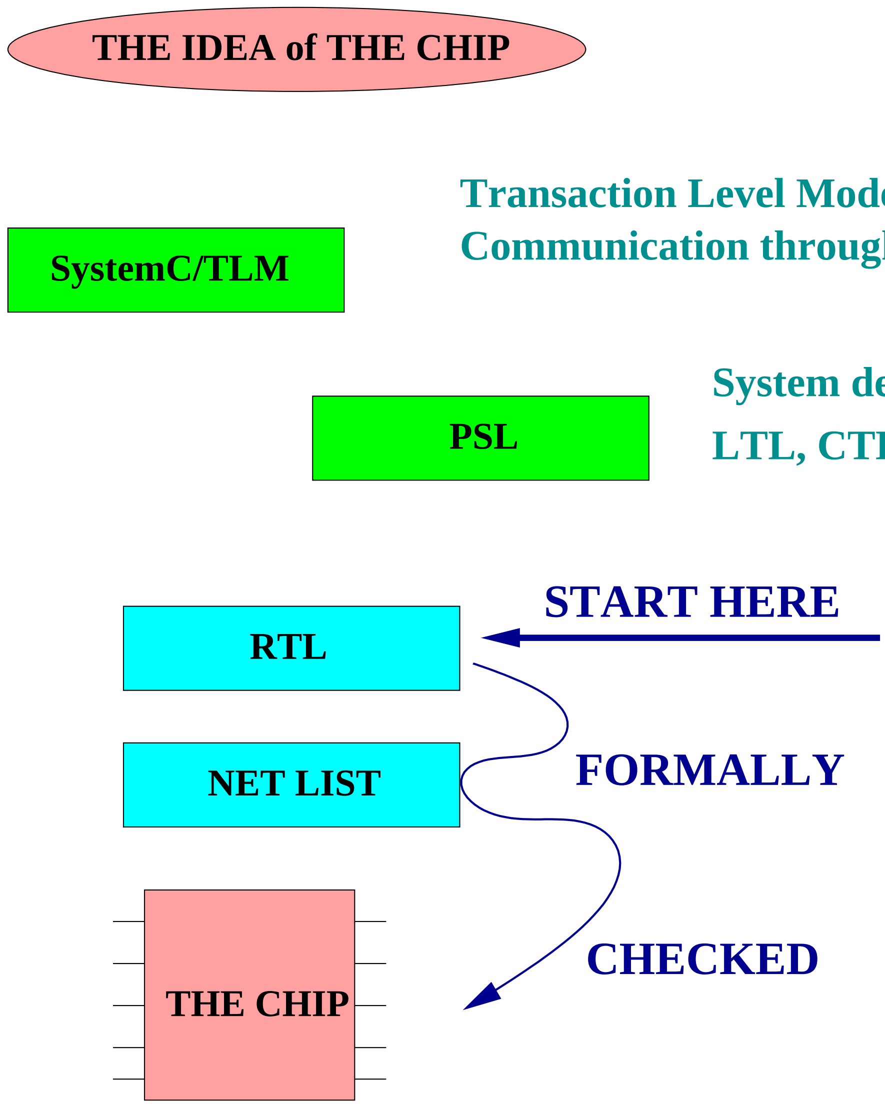
SoC Design Verification Flow



Communication via signals
Synchronisation (clock, reset)
Manipulation of bit vectors

Basic Boolean Gates

SoC Design Verification Flow



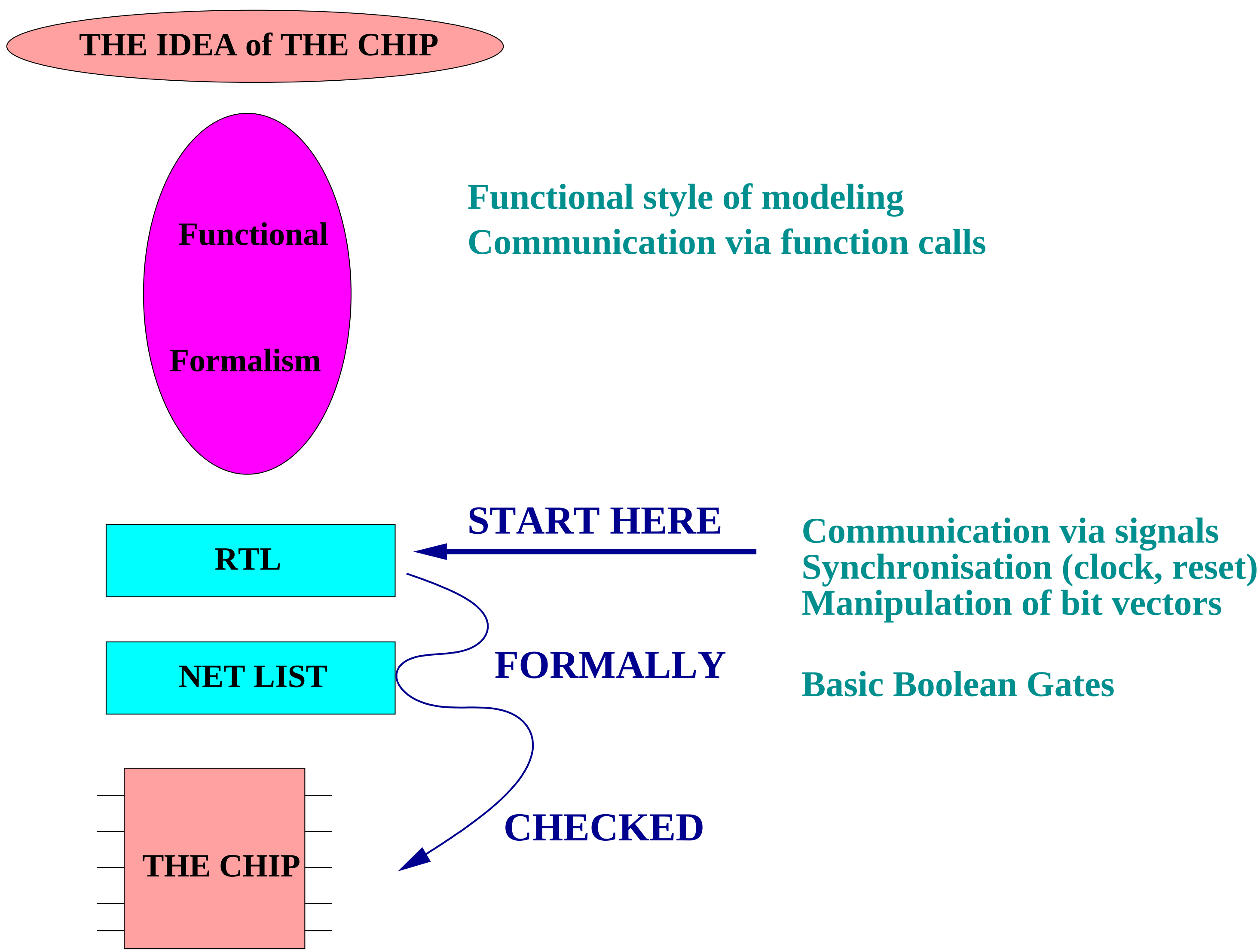
Transaction Level Modeling
Communication through abstract channels

System described by properties
LTL, CTL, regular expressions, ...

Communication via signals
Synchronisation (clock, reset)
Manipulation of bit vectors

Basic Boolean Gates

SoC Design Verification Flow



Contribution

Our study considers:

- the initial design steps
- a functional modeling style
- on chip communications

Our achievement is:

A Generic Network On Chip Model ($GeNoC$)

A Generic Network on Chip

$GeNoC$ consists in:

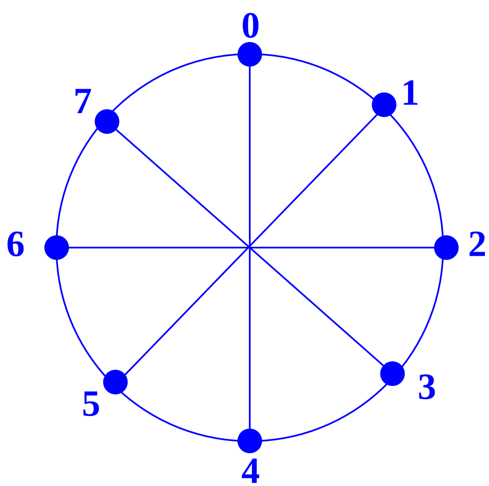
- a functional model of a communication architecture
- correctness criteria about the model
- partitioning the model

Modular framework for both design and validation

Outline

- Octagon Network on Chip
- Communication Principles
- $GeNoC$ Definition and Correctness
- ACL2 Theorem and Proof
- $GeNoC$ Routing

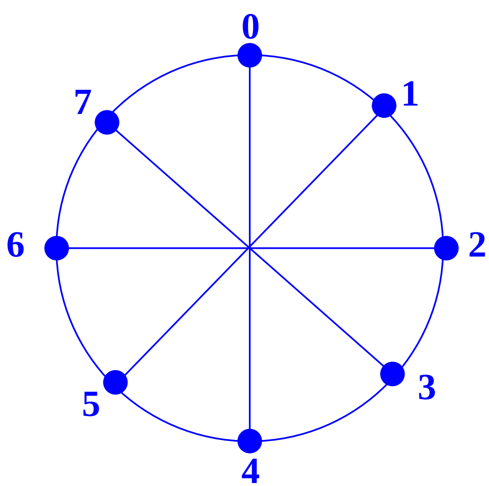
Octagon Network on Chip



- 8 nodes
- extensible to $4 * i$
- bidirectional links
- simple shortest path routing algorithm

- Design by STMicroelectronics ref: DAC'01 and IEEE Micro 2002 by F. Karim *et al*.

Routing Algorithm



$REL\_ADDR = (dest - current)\ mod\ 8$

if $REL\_ADDR = 0$

then stop

elsif $REL\_ADDR = 1 \vee 2$

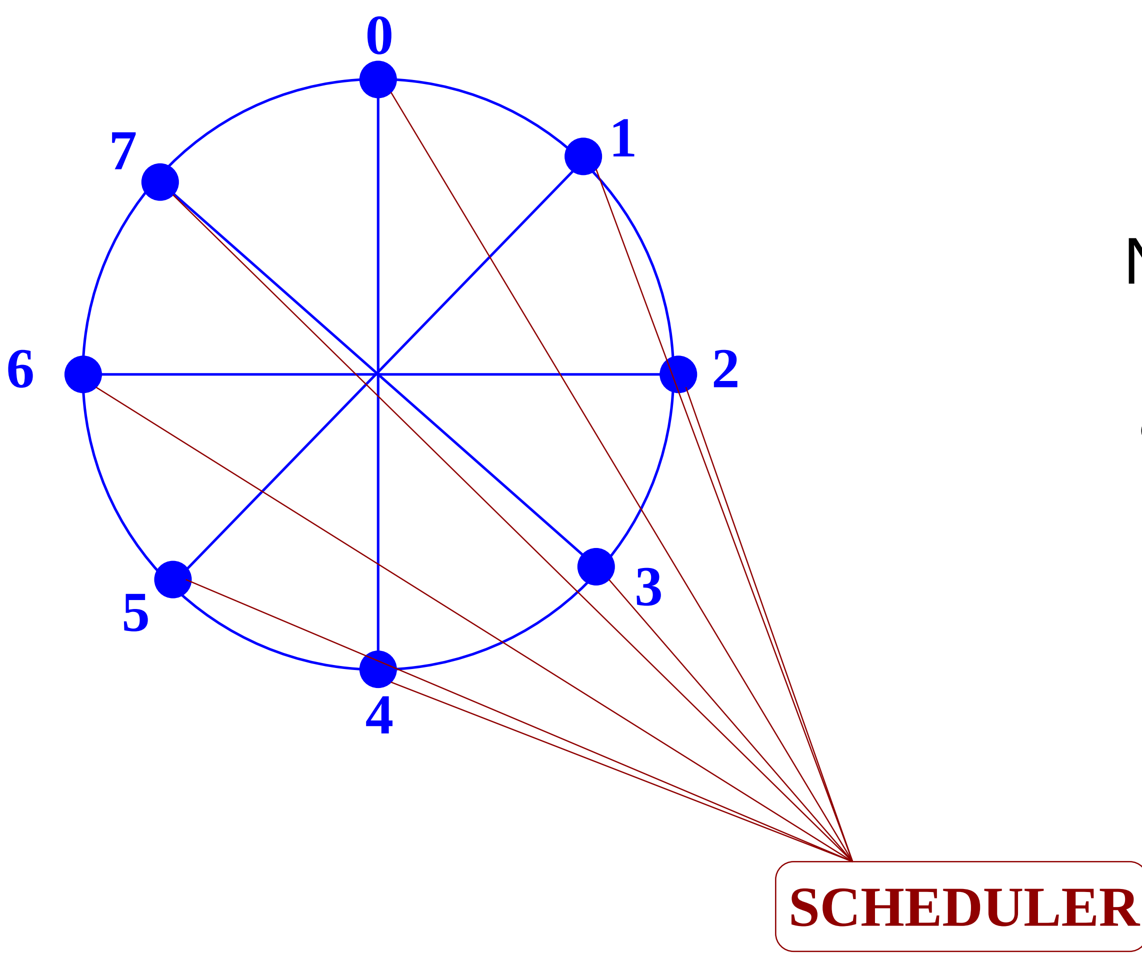
then go clockwise

elsif $REL\_ADDR = 6 \vee 7$

then go counter clockwise

else go across

Octagon Scheduling Policy



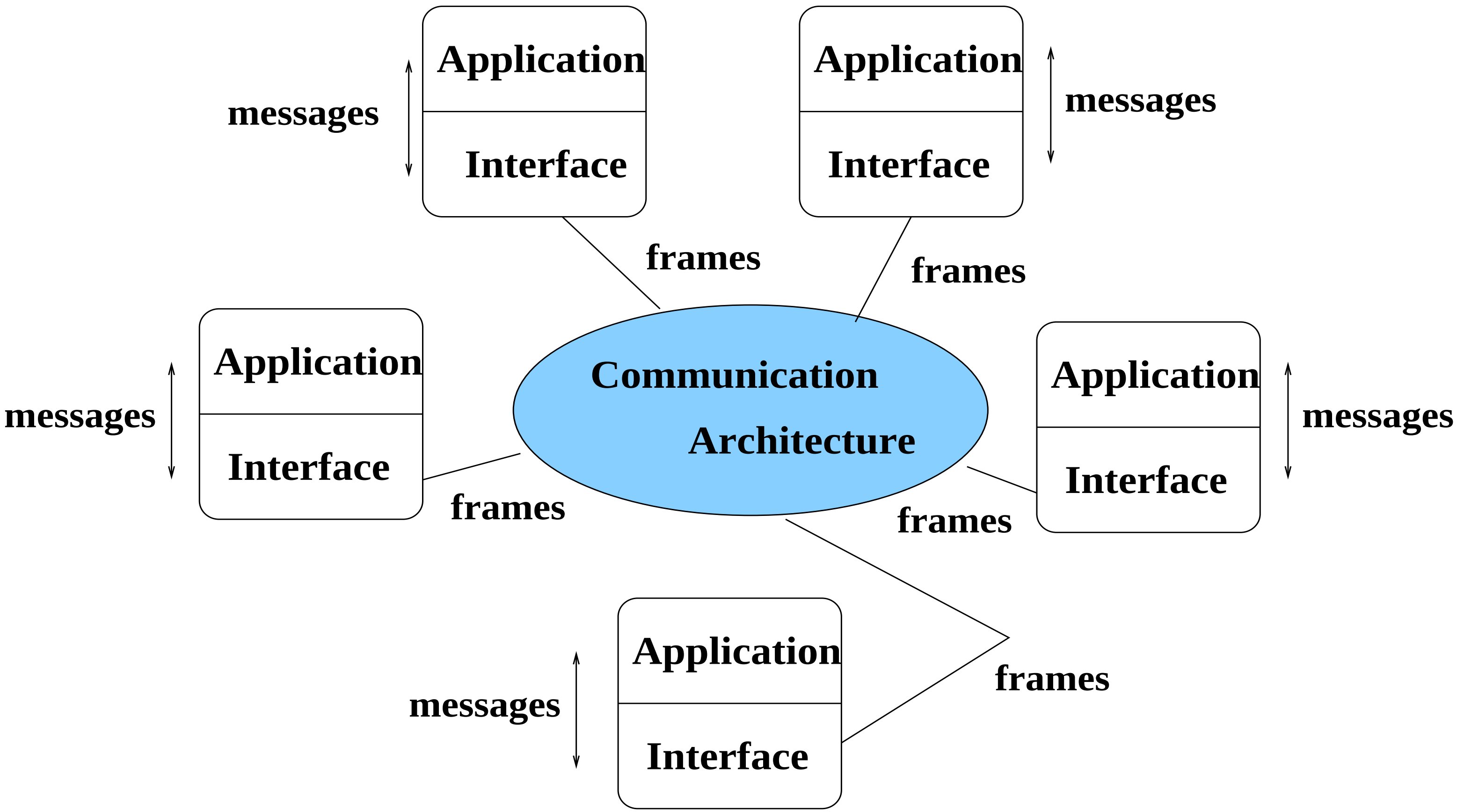
Nodes connected to

a central scheduler

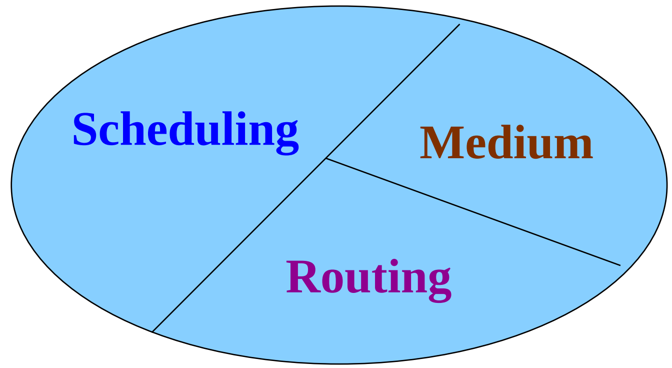
Outline

- Octagon Network on Chip
- Communication Principles
- $GeNoC$ Definition and Correctness
- ACL2 Theorem and Proof
- $GeNoC$ Routing

Communication Principles



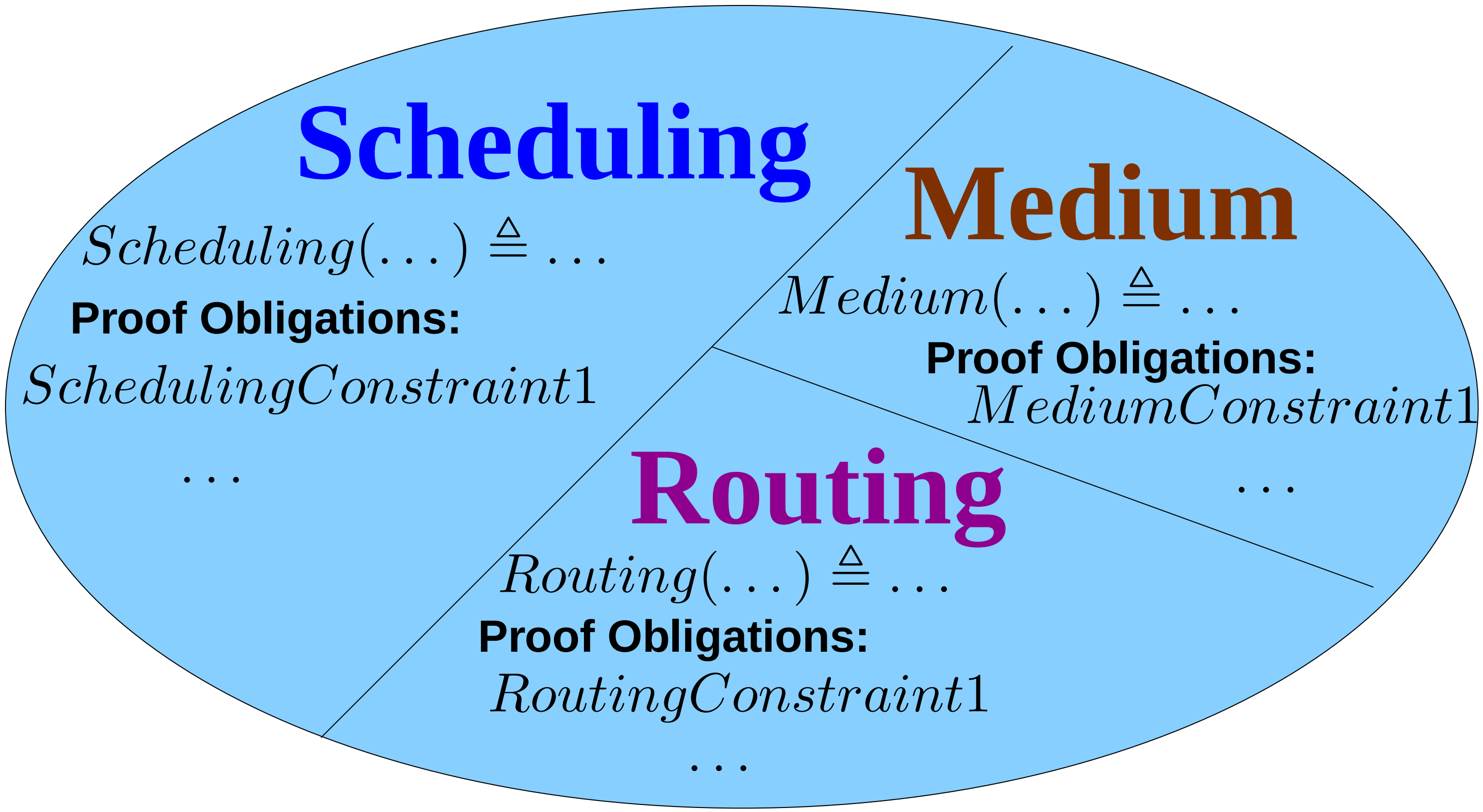
$GeNoC$ **Principles**



Partition of the communication architecture into:

- Routing: to compute a route between two nodes
- Scheduling: to schedule or to delay a communication
- Medium (topology): to convey a frame from one node to another

$GeNoC$ Principles



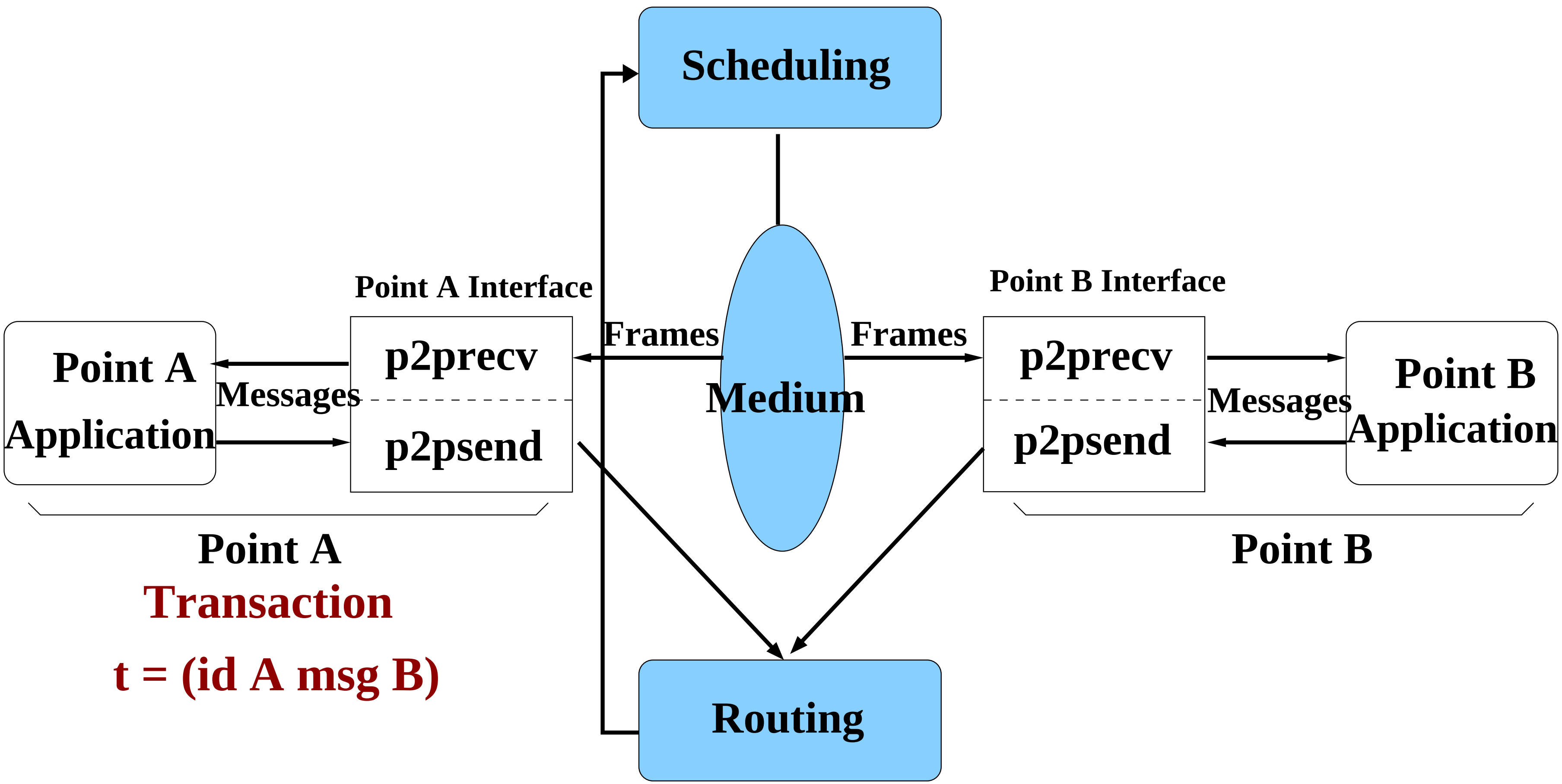
$$GeNoC(\ldots) \triangleq \mathcal{F}(Routing, Scheduling, Medium)$$

$GeNoC$ correctness proved from the constraints

Outline

- Octagon Network on Chip
- Communication Principles
- $GeNoC$ Definition and Correctness
- ACL2 Theorem and Proof
- $GeNoC$ Routing

$GeNoC$ **Definition**



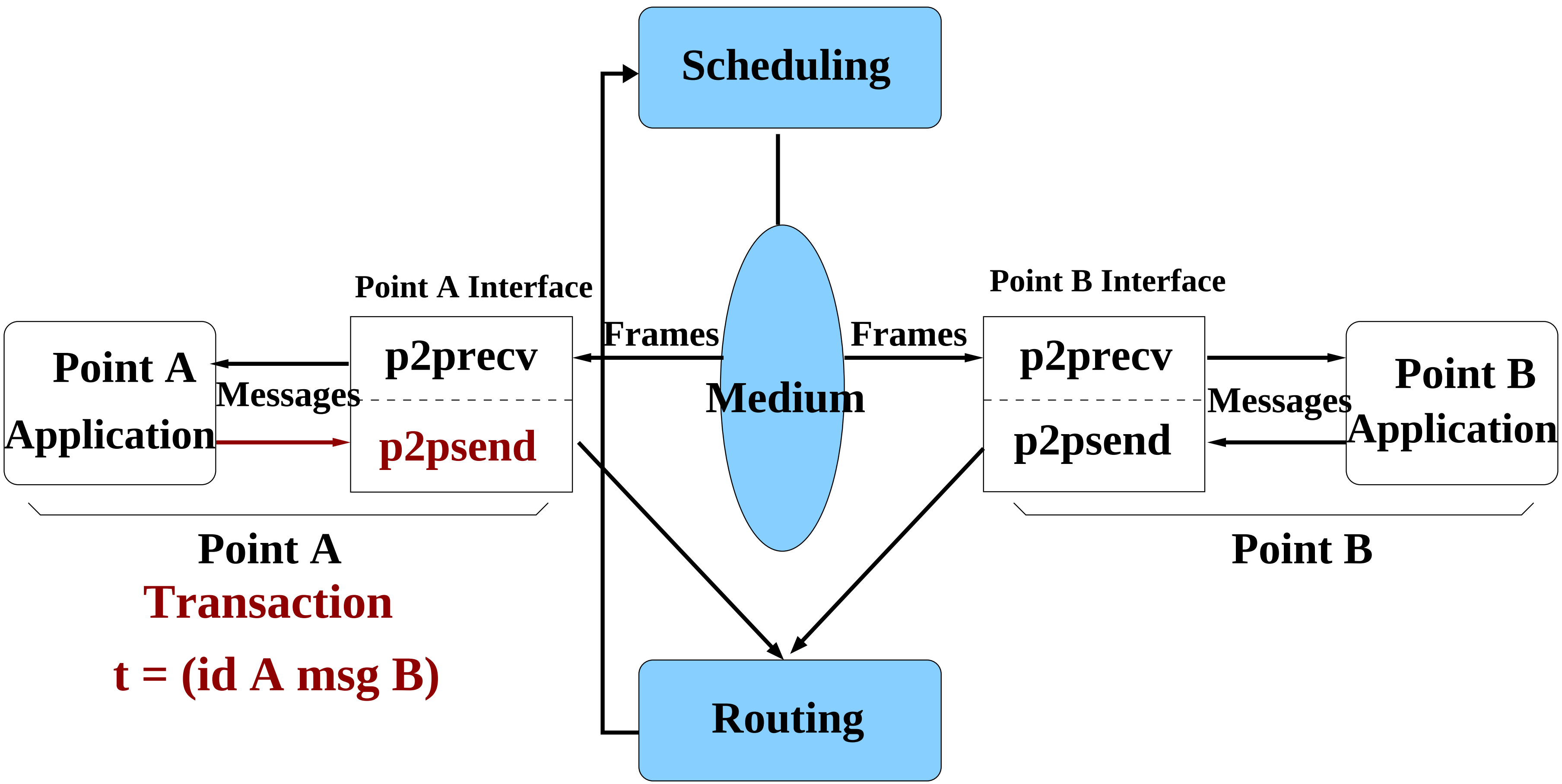
$GeNoC$ **Transaction**

A transaction t is a tuple of the form $(id\ A\ msg\ B)$ where:

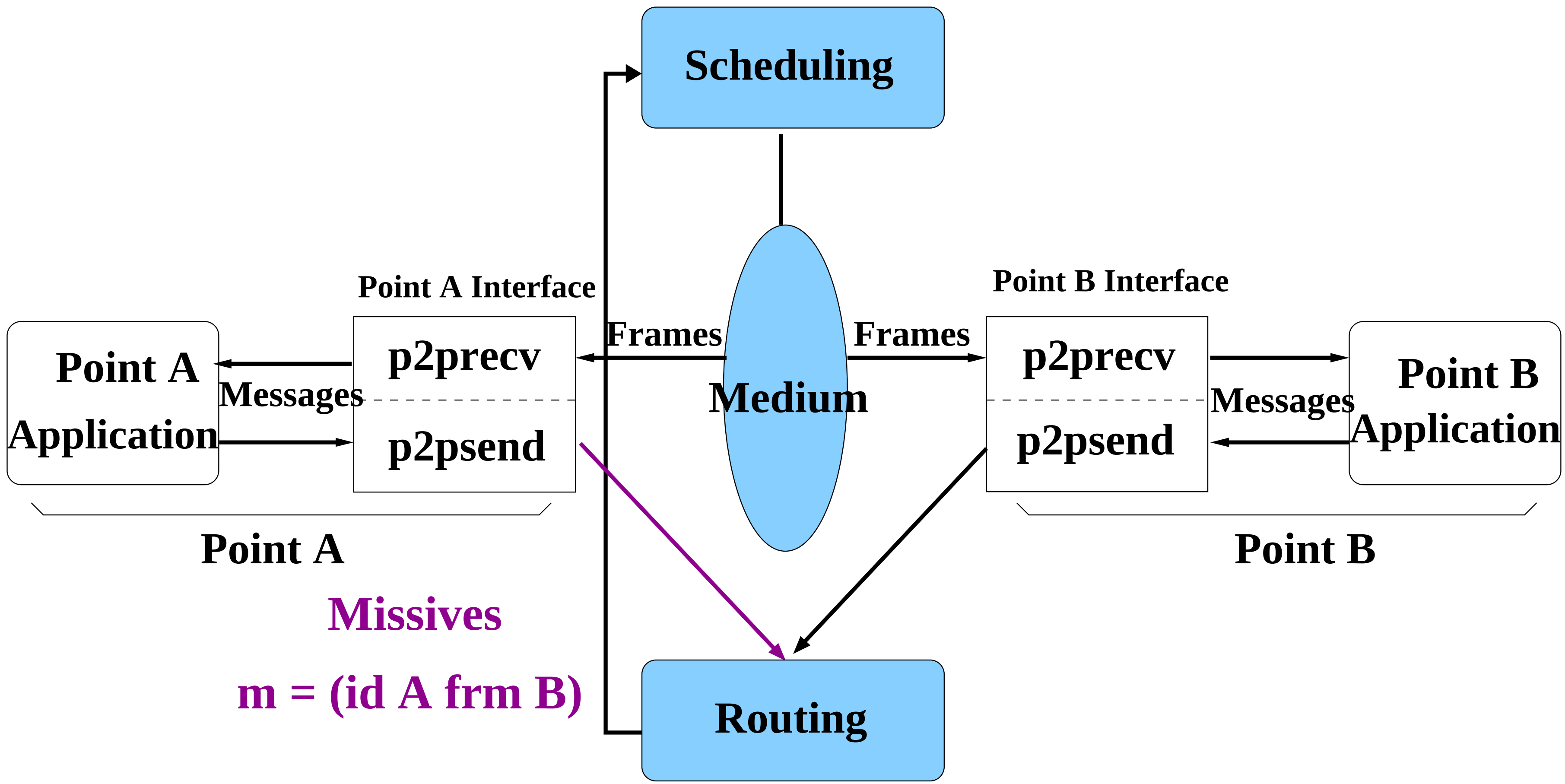
- $id \in \mathbb{N}$: is a unique identifier
- msg is an "abstract" message
- A is the origin, and B is the destination of msg

A transaction represents the intention of A to send msg to B.

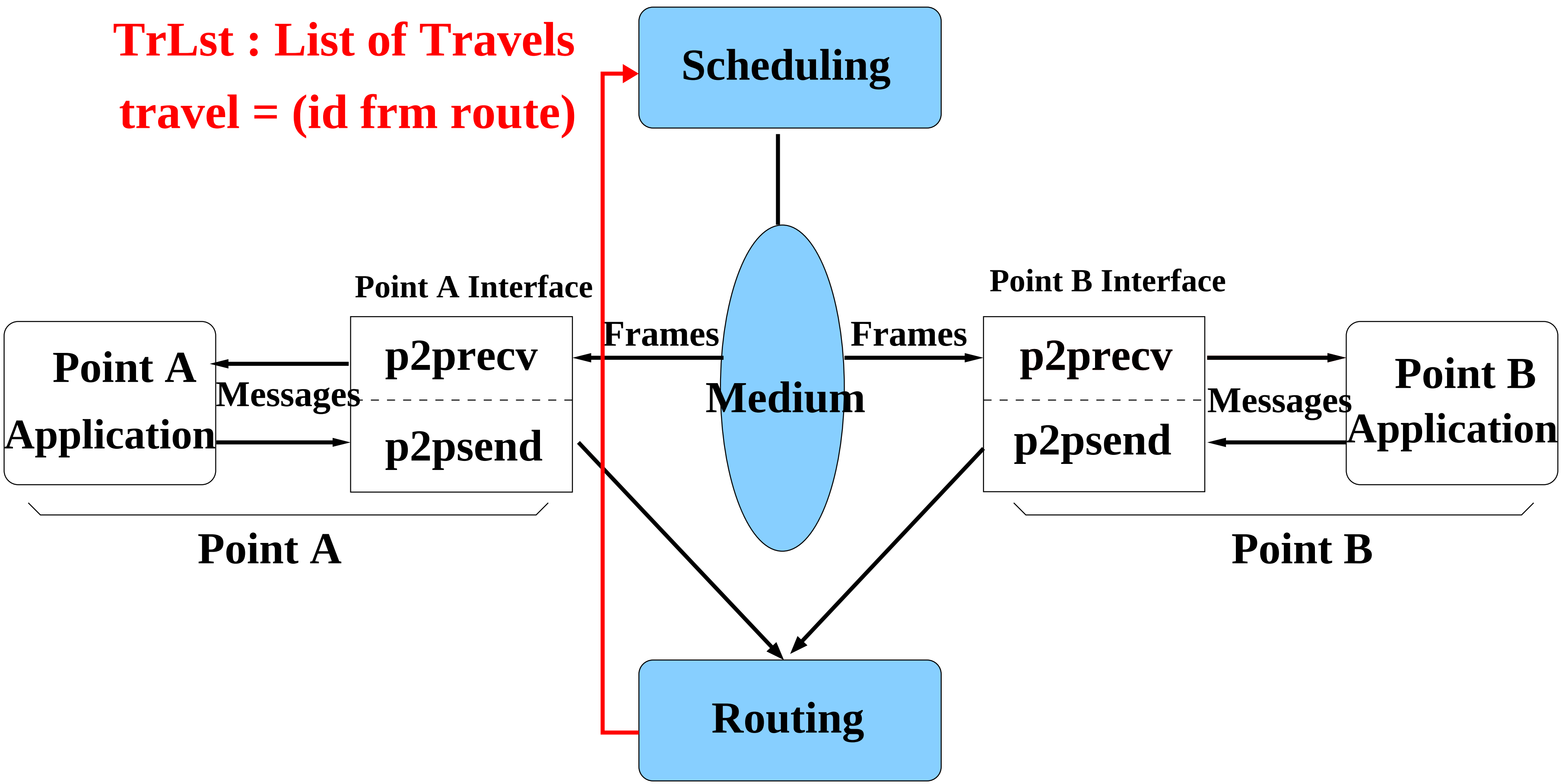
From transactions to missives



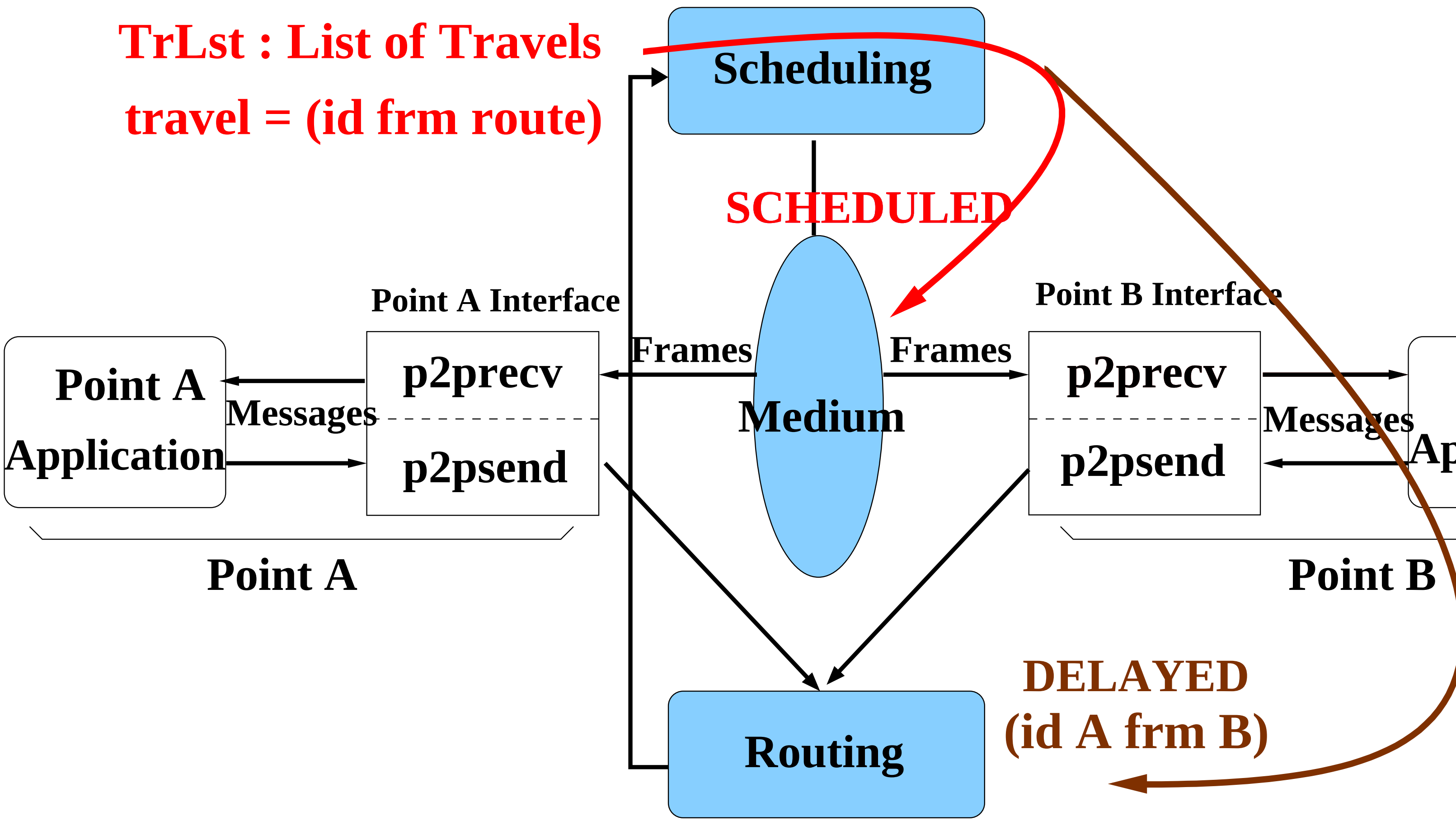
From transactions to missives



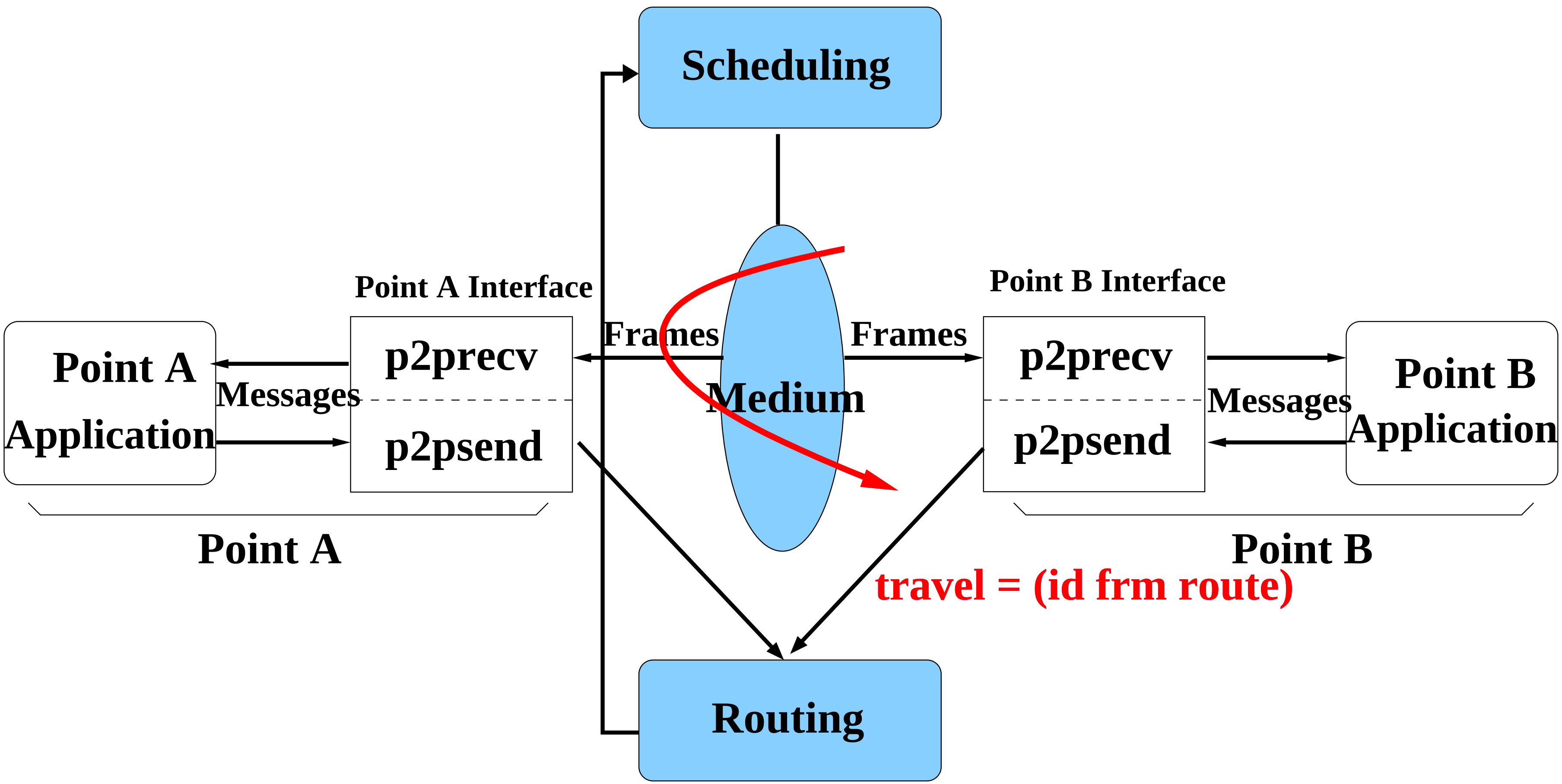
$GeNoC$ Routing



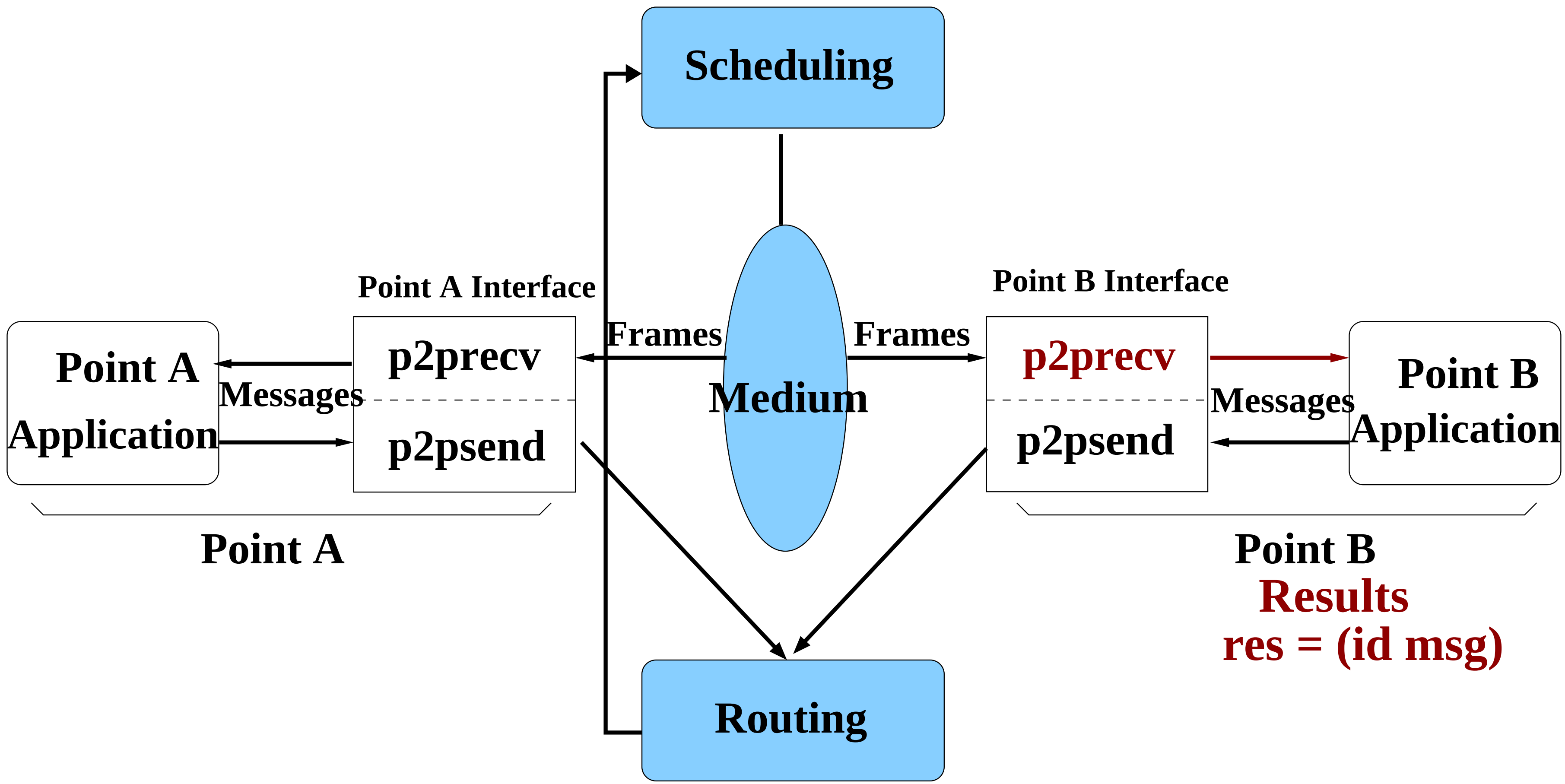
$GeNoC$ **Scheduling**



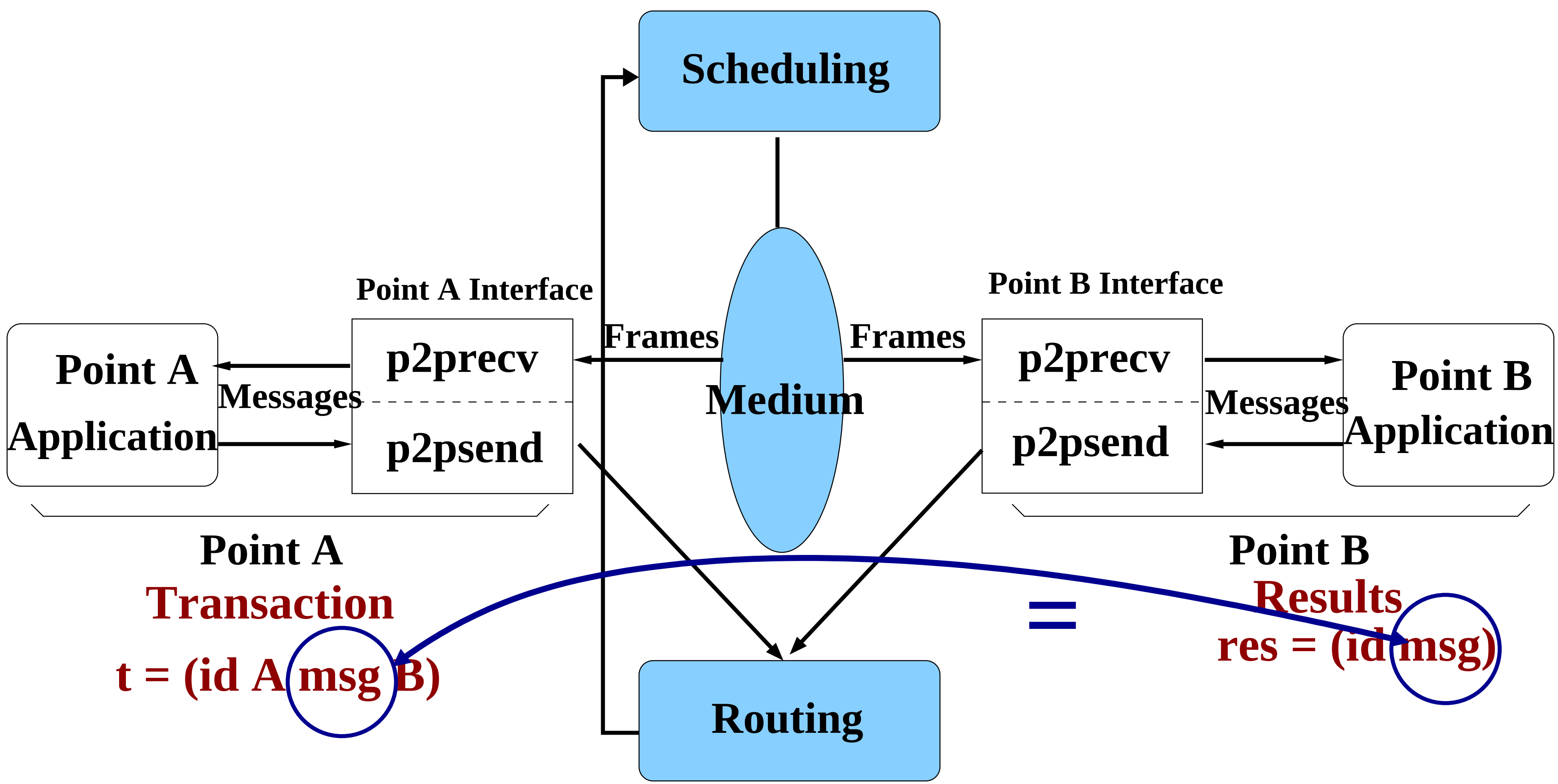
$GeNoC$ **Medium**



GeNoC Receive



$GeNoC$ **Correctness**



$GeNoC$ **Termination**

$GeNoC$ is a recursive function and must be proved to terminate because:

- it is a prerequisite for mechanized reasoning (here ACL2)
- the input list of transactions is *finite*

To ensure the termination, we associate to every point a *finite* number of attempts. At every recursive call of $GeNoC$, every point with a pending transaction consumes one attempt.

$GeNoC$ **Definition**

- $\mathcal{T}$: input list of transactions
- $NodeSet$: set of existing nodes
- $GenericNodeSet$: generic domain of $NodeSet$
- $AttLst$: domain for the lists of attempts
- $\mathcal{R}$: list of results

$$GeNoC : \mathcal{P}(\mathcal{T}) \times GenericNodeSet \times AttLst$$

$$\rightarrow \mathcal{P}(\mathcal{R}) \times \mathcal{P}(\mathcal{T})$$

Completed Transactions

Aborted Transactions

$GeNoC$ **Correctness**

Theorem.

$$\forall (id_r \; msg_r) \in \mathcal{R},$$
$$\qquad \exists ! (id_t \; a_t \; msg_t \; b_t) \in \mathcal{T} \mid id_r = id_t \wedge msg_r = msg_t$$

For any result r, there exists a unique initial transaction t such that t has the same id and msg as r.

$GeNoC$ **Correctness**

Theorem.

$$\forall (id_r\ msg_r) \in \mathcal{R}, \quad \exists ! (id_t\ a_t\ msg_t\ b_t) \in \mathcal{T} \mid id_r = id_t \wedge msg_r = msg_t$$

For any result r, there exists a unique initial transaction t such that t has the same id and msg as r.

This theorem is complemented by a condition (*TravelCondition*) which proves that every transaction is received by the correct destination.

Outline

- Communication Principles
- Octagon Network on Chip
- $GeNoC$ definition and correctness
- $GeNoC$ Definition and Correctness
- ACL2Theorem and Proof
- $GeNoC$ Routing

ACL2 Theorem

* ACL2 is quantifier free

$\mathcal{T}$ = initial transactions

$$\downarrow$$

$$\mathcal{T}/\mathcal{R}_{ids}$$

$$\downarrow$$

$$Messages(\mathcal{T}/\mathcal{R}_{ids})$$

$\mathcal{R}$ = list of results

$$\downarrow$$

$$Messages(\mathcal{R})$$

$$Messages(\mathcal{T}/\mathcal{R}_{ids}) = Messages(\mathcal{R})$$

Constraints Overview

- Medium
  - (M1) $Medium(TrLst) = TrLst$ iff every route in $TrLst$ is consistent with the topology
- Routing
  - (R1) Every route ends with the correct destination (*Travel Condition*)
  - (R2) Every route is consistent with the topology
  - (R3) Frames are not modified by the routing function

Constraints Overview

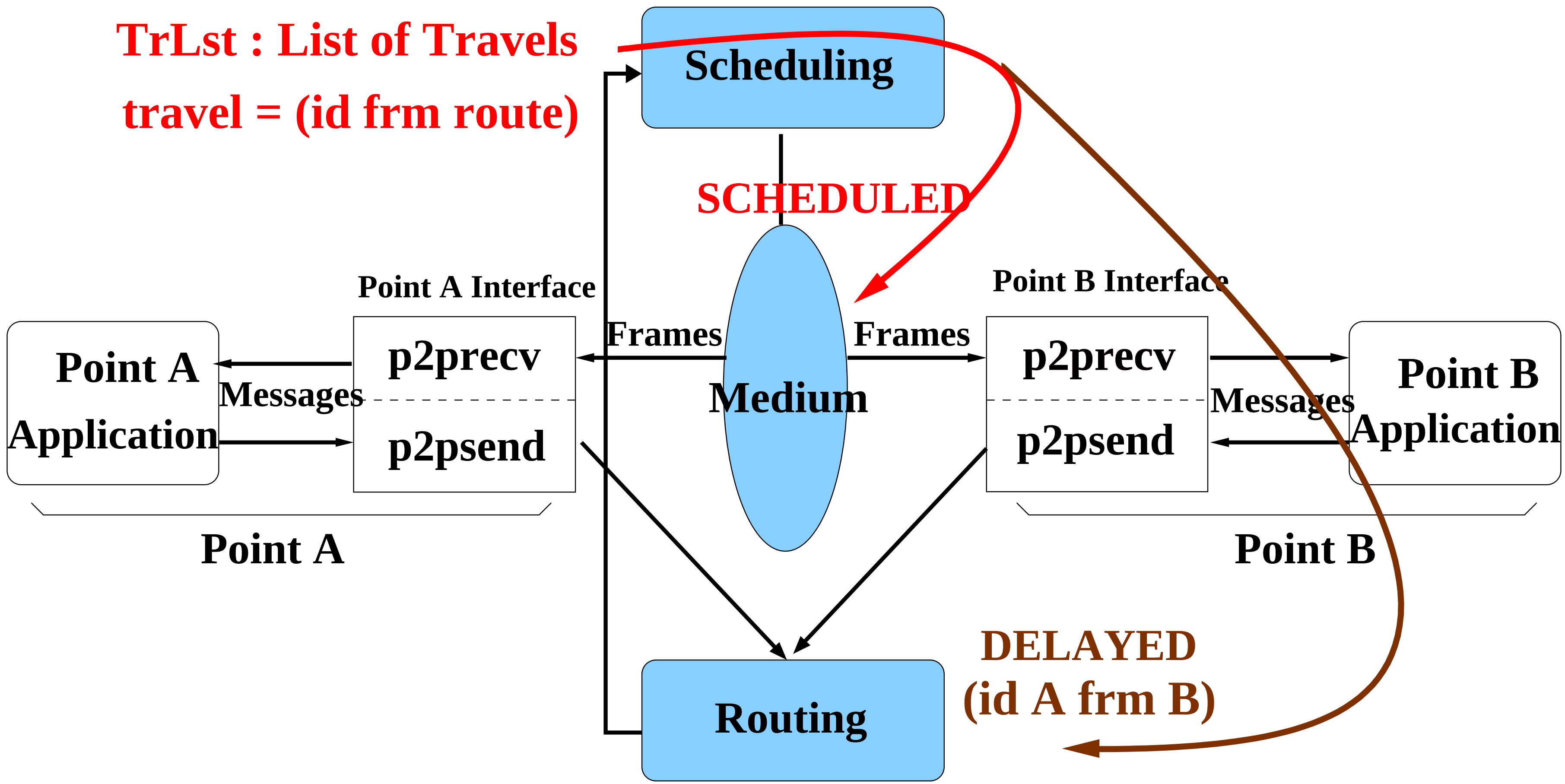
- Scheduling
  - (S1) Preserve route correctness
  - (S2) $Delayed$ and $Scheduled$ are distinct sublists of $TrLst$
- Interfaces
  - (I1) $p2p_{recv} \circ p2p_{send}(msg) = msg$

ACL2 Proof Sketch

Proof by induction on the structure of $GeNoC$

- Base Case: trivial
  ($SumOfAttempts = 0 \Rightarrow R_{ids} = \epsilon$)
- Induction Step
  - Induction Hypothesis $\Rightarrow$ $Delayed$ is correct
  - Remaining Goal: prove that $Scheduled$ is correct

ACL2 Proof Sketch



Proof of $\mathit{Scheduled}$ Correctness

- Constraints M1, R2 and S1 $\Rightarrow$ remove calls to Medium
- Constraints S2 and R3 $\Rightarrow$ every result is produced by $p2p_{recv} \circ p2p_{send}(msg)$
- Constraints I1 concludes that the message that is received is equal to the message that is sent $\square$

Outline

- Communication Principles
- Octagon Network on Chip
- $GeNoC$ definition and correctness
- $GeNoC$ Definition and Correctness
- ACL2 Theorem and Proof
- $GeNoC$ Routing

Routing

- The routing algorithm is represented by function $Routing$:
  - $Routing : \mathcal{P}(Missives) \times GenericNodeSet \rightarrow \mathcal{P}(Travels)$
- Function $Routing$ associates a route to a travel without modifying it
  - $ToMissives(Routing(Missives, NodeSet)) = Missives$

Desired Topology

The predicate $AvailableMovesp(TrLst, NodeSet)$ defines what the topology of the network should be, *i.e* the designer's intent.
It defines the different available moves in the network.

Routing

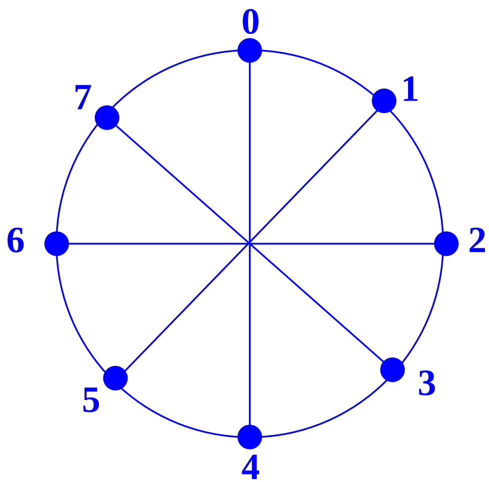
A route is correct if:

$$CorrectRoutesp(Routing(Missives, NodeSet))$$

$$\wedge$$

$$Availablemovesp(Routing(Missives, NodeSet))$$

Octagon Topology



bidirectional links

Num_Node = number of nodes

Num_Node is natural and a multiple of 4

- $NodeSet$ = naturals up to $Num\_Node - 1$
- available moves:
  - clockwise, counterclockwise, across ($OctagonAvailableMovesp$)

Octagon Routing Validation

- We prove that $OctagonRouting$ satisfies:
  - $CorrectRoutesp$
  - $OctagonAvailableMovesp$
- We prove that each path is bounded by $\frac{Num\_Node}{4}$

Conclusions

- About the ACL2 model
  - 1500 lines, 40 functions and 85 theorems
  - Intensive use of the *functional instanciation* principle
  - Most induction schemes are guessed automatically by ACL2
- A generic model for networks on chip ($GeNoC$)
  - Functional modeling style at first design steps
  - Considers a *complete* communication system
  - Based on three independent parts
    - **Modular approach for design**

Conclusions

- Correctness criteria
  - Messages are either lost or reach properly their expected destination
  - Sufficient constraints on each part
    - **Modular approach for validation**
- Applications of $GeNoC$:
  - Octagon
  - a 2D-mesh with an XY-routing algorithm

Perspectives

- Extending $GeNoC$
  - Work on adaptive routing algorithms
  - Extend the model to master/slave communications
- Translation to standard HDLs
  - SystemC
  - VHDL Synthesis Subset 2004

THANK YOU !!