Formalizing On Chip Communications in a Functional Style

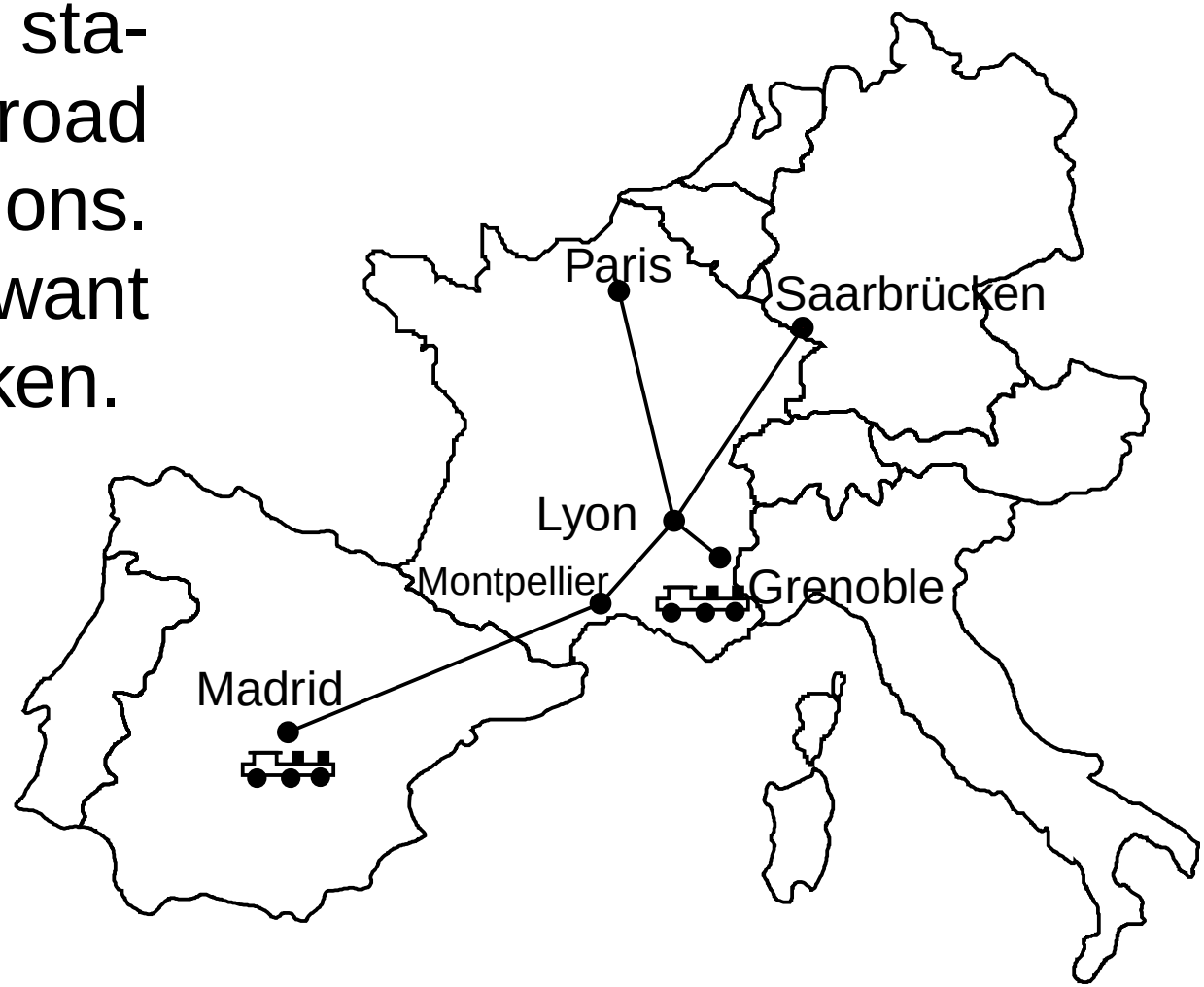
Julien Schmaltz

Saarland University

Institute for Computer Architecture

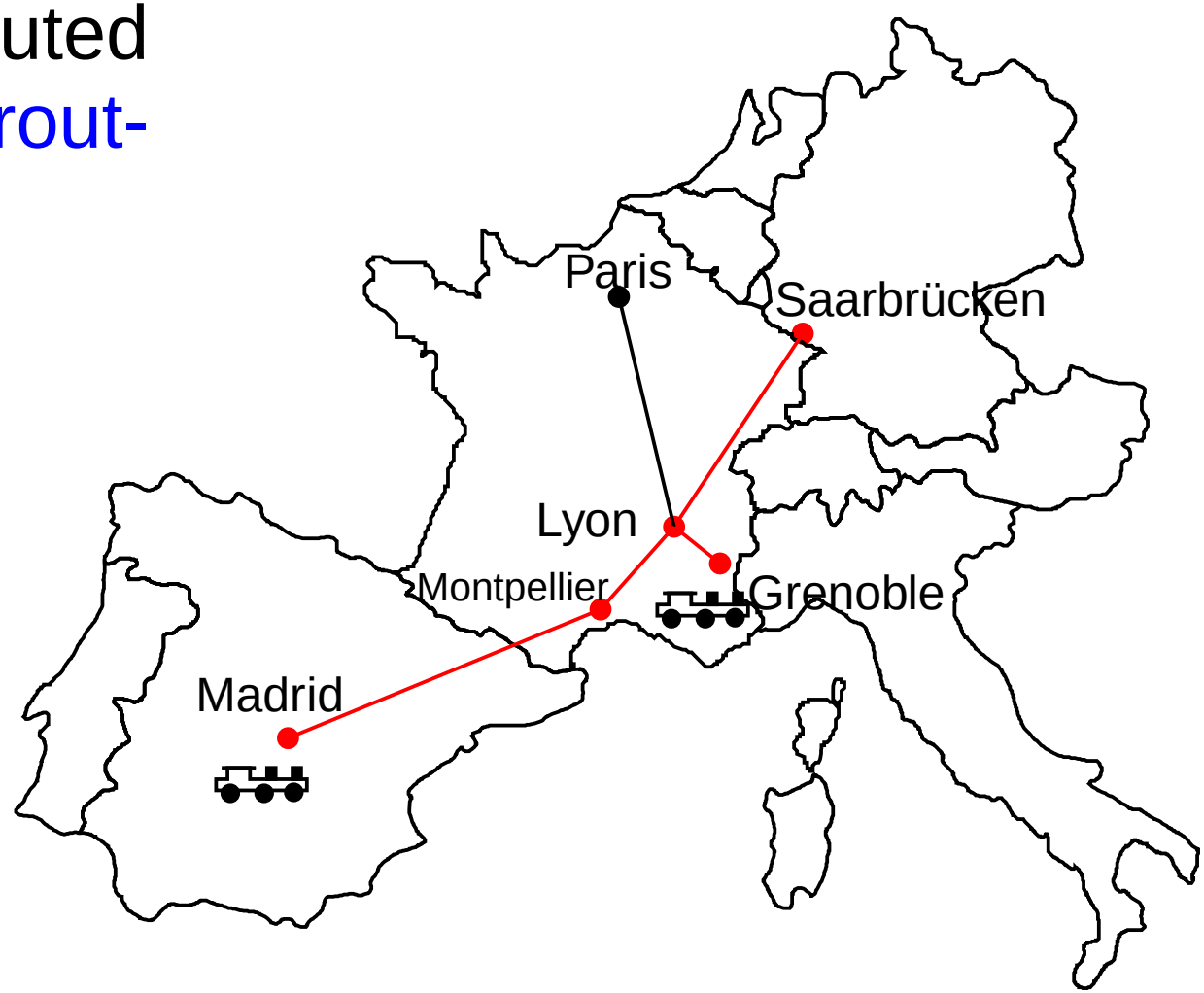
RailNet

One platform per station and one railroad between two stations. Peter and John want to go to Saarbrücken.



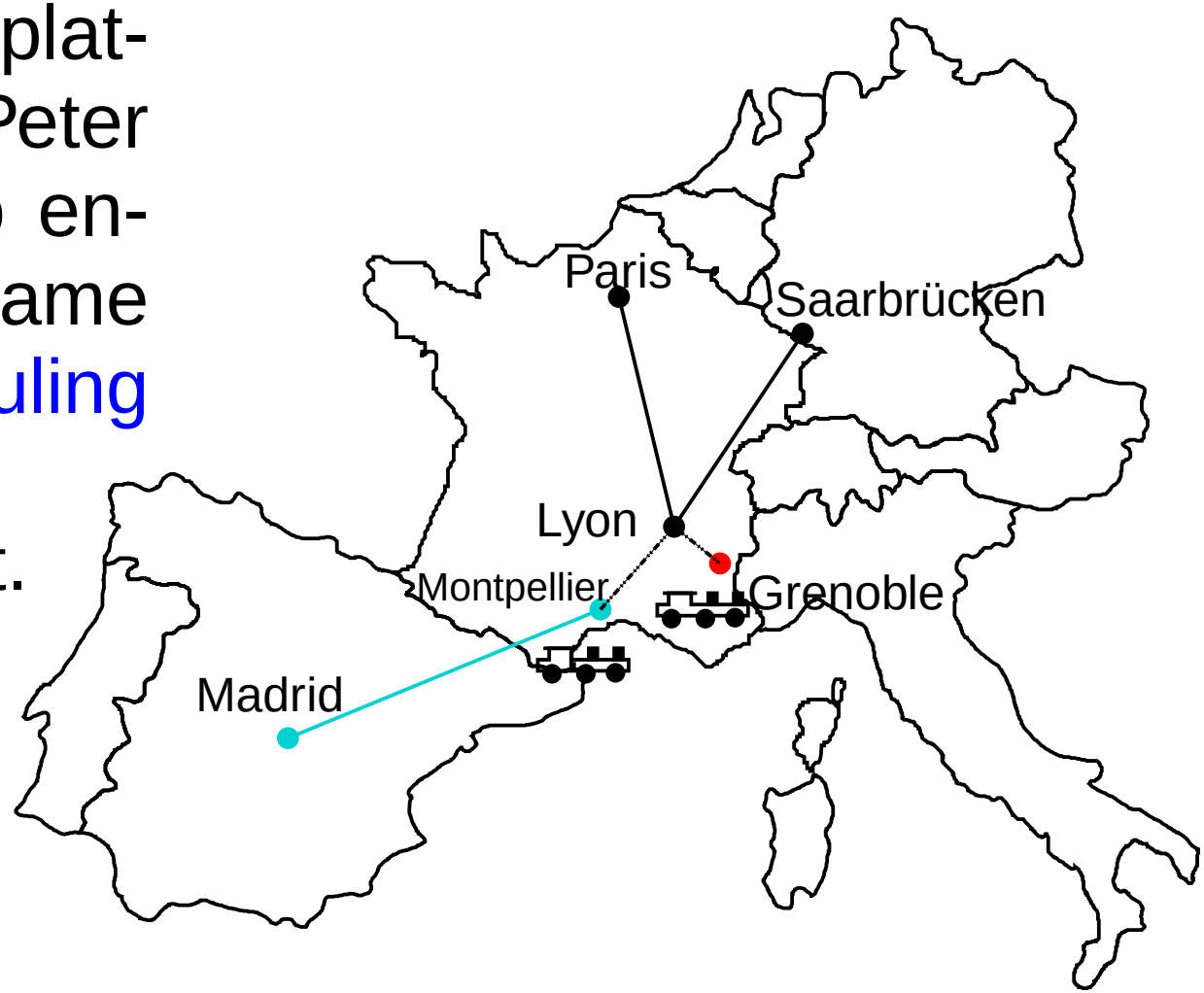
Route Computation

Routes are computed at origins: source routing.



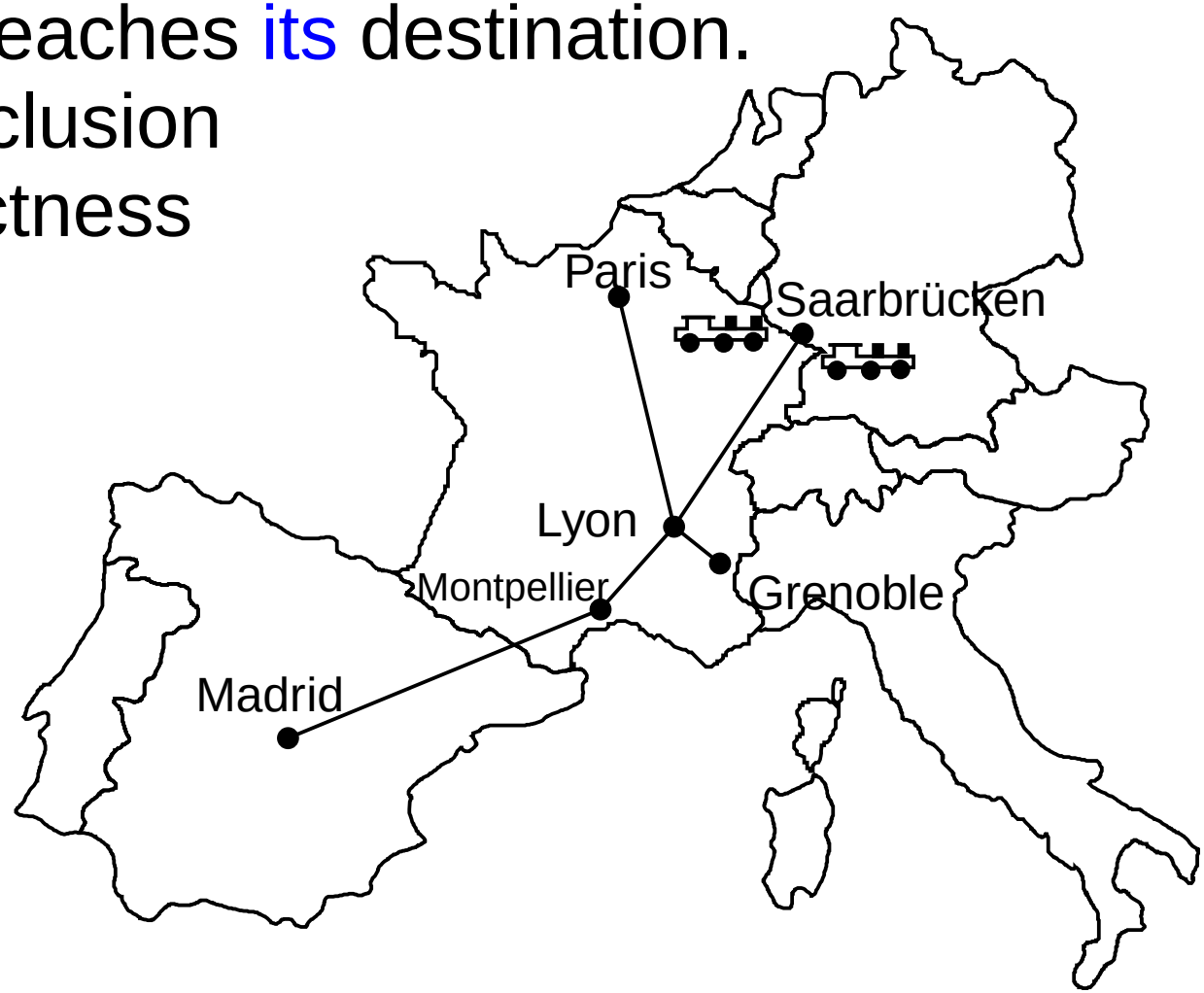
Conflicts Solving

There is only one platform per station. Peter and John want to enter Lyon at the same time. The scheduling policy solves this conflict.



Network Verification

Thm. Each train reaches its destination.
Proof: Mutual Exclusion
and routing correctness



Another Network Verification

Thm. Each plane reaches its destination.
Proof: Mutual Exclusion
and routing correctness

Particular and General

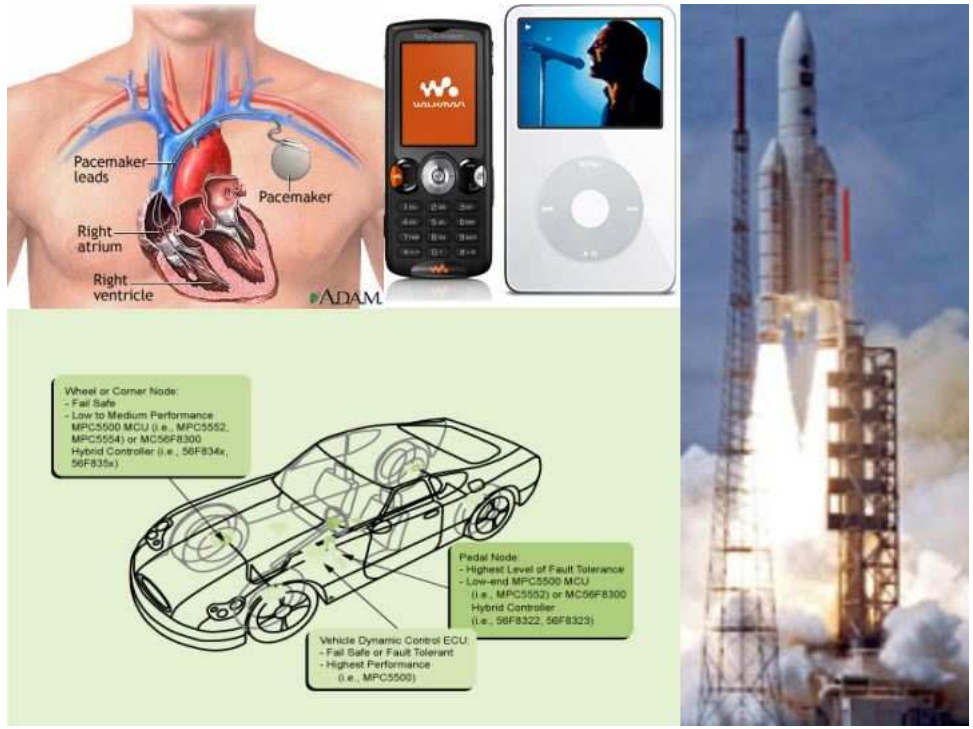
- What is particular to **one** network
  - Scheduling Policy
  - Number of platforms and railroads
- What is common to **any** network
  - Structure, routing + scheduling
  - Routing and overall correctness

Our achievement :
a **formal** model of a **generic** network

Outline

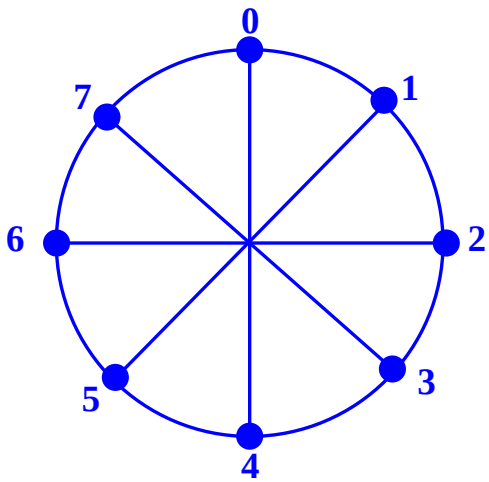
- Systems on a Chip
- Communication Principles
- $GeNoC$ Definition and Correctness
- Applications of $GeNoC$

Systems On a Chip



- Everywhere, critical systems
- Ever growing complexity (HW & SW)
- Safety and correct behavior

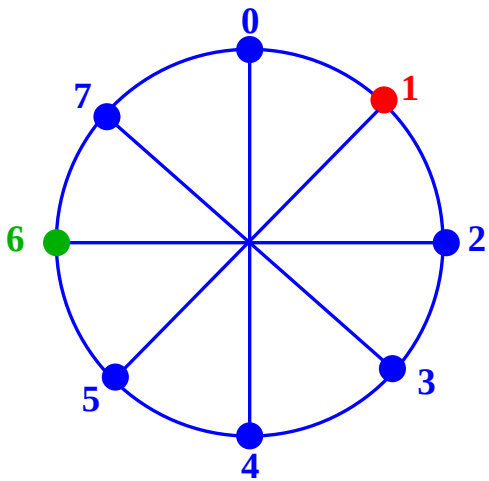
Octagon Network on Chip



- 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



$RelAd = (dest - current) \bmod 8$

if $RelAd = 0$

then stop

elsif $RelAd = 1 \vee 2$

then go clockwise

elsif $RelAd = 6 \vee 7$

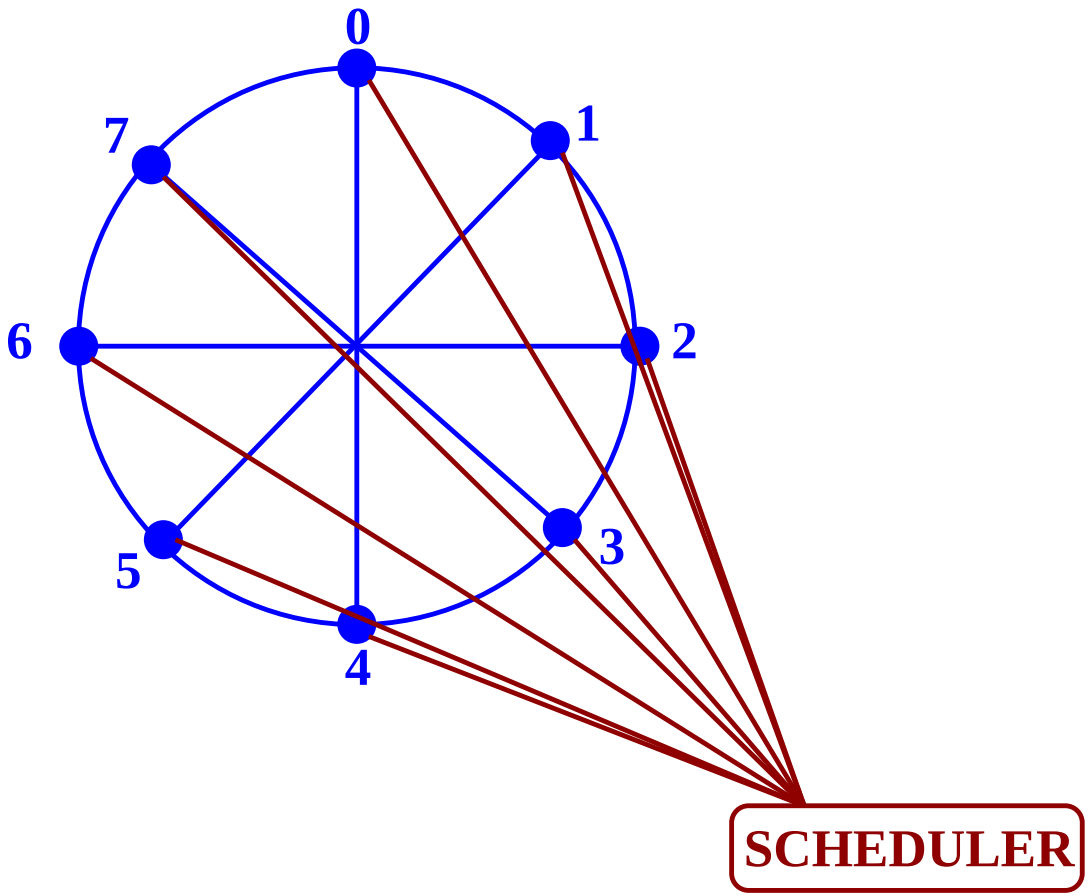
then go counter clockwise

else go across

endif

Example: route from 1 to 6

Octagon Scheduling Policy



Nodes connected to

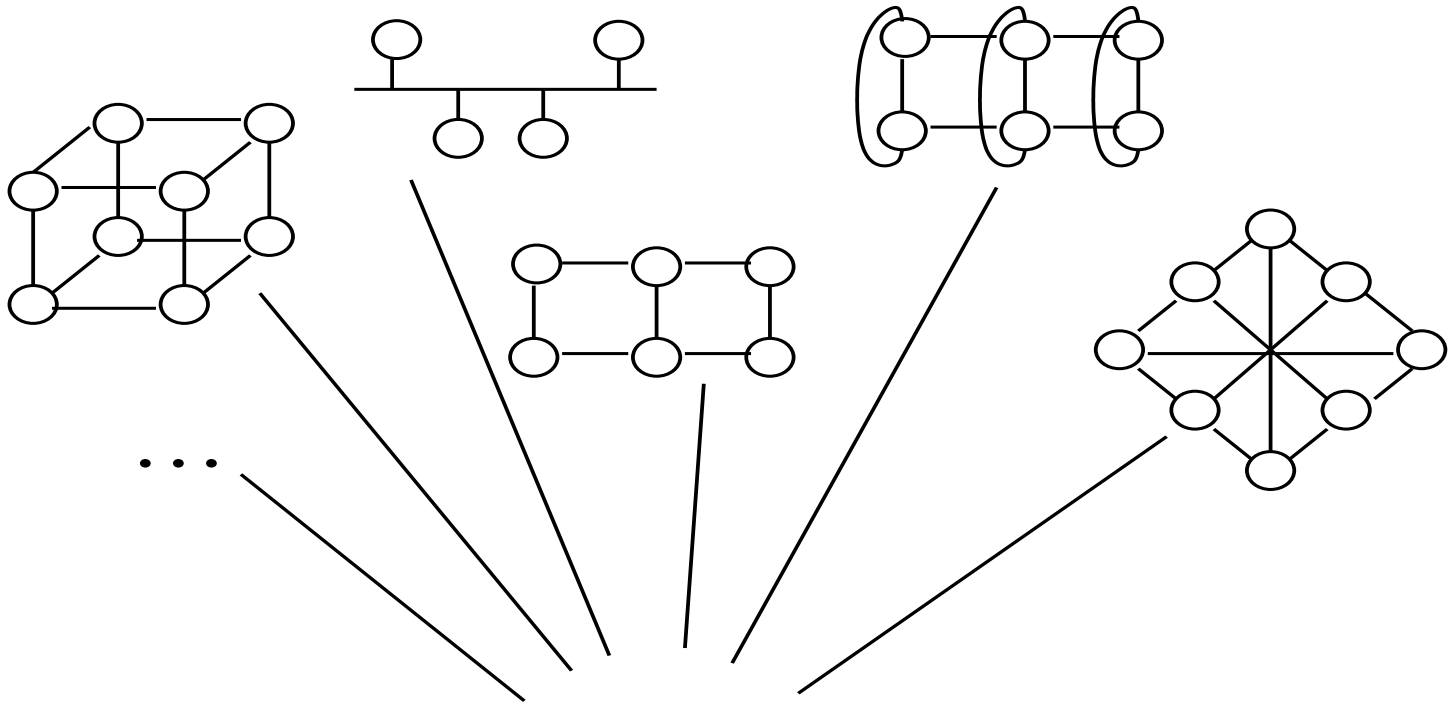
a central scheduler

Formal Verification of NoC's

- AMBA bus by model cheching (Roychoudhury *et al.*, 2003)
- AMBA by M.C. and HOL (Amjad, 2004)
- Æthereal protocol from Philips by PVS (Gebremichael *et al.*, 2005)
  - Low level of abstraction
  - Particular cases only
  - No general method

Global Objective

One model for **all** architectures



?

Contribution

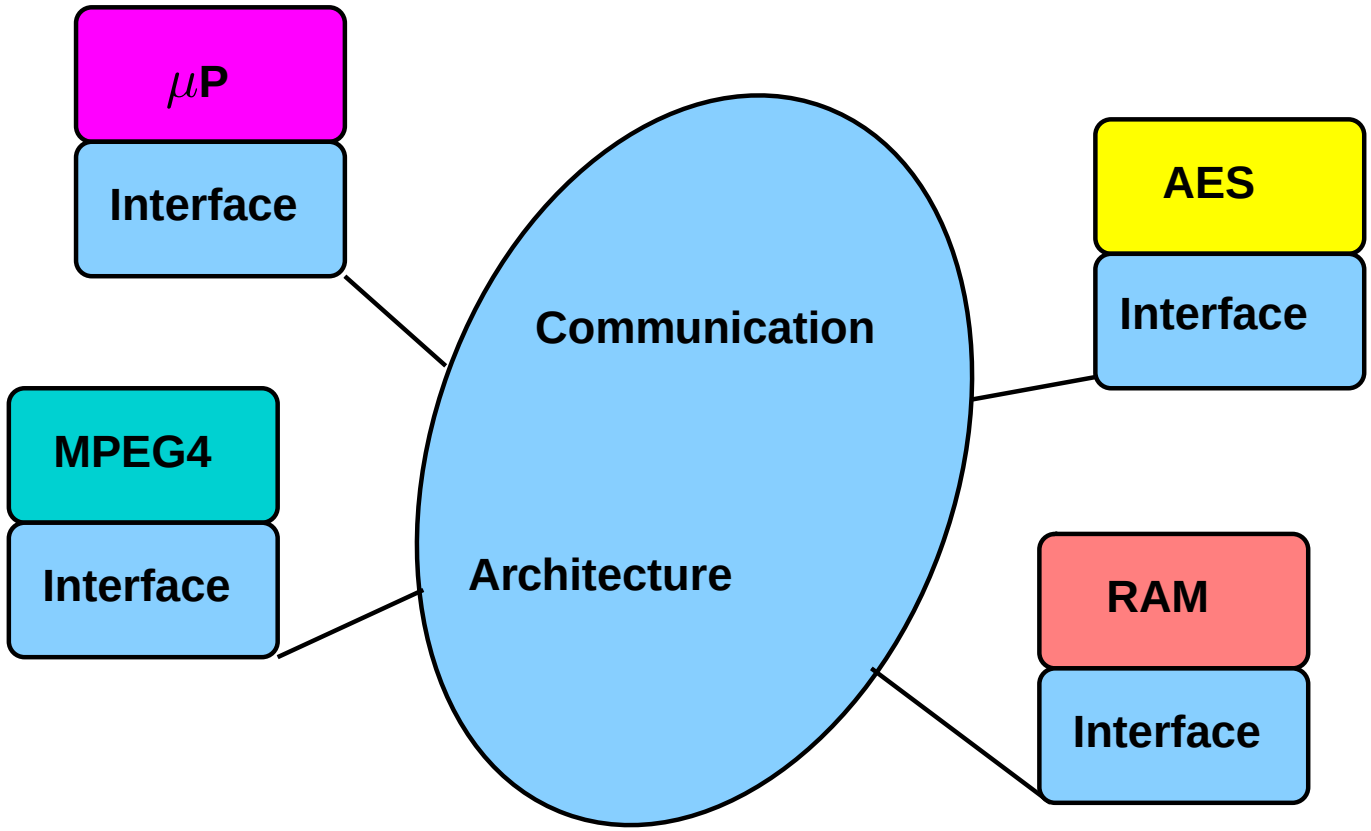
A functional formalism for communications: $GeNoC$ (Generic Network on Chip)

- Identifies the essential constituents and their properties
- Formalizes the interactions between them
- Correctness of the system is a consequence of the essential properties of the constituents
- (Mechanized support in ACL2)

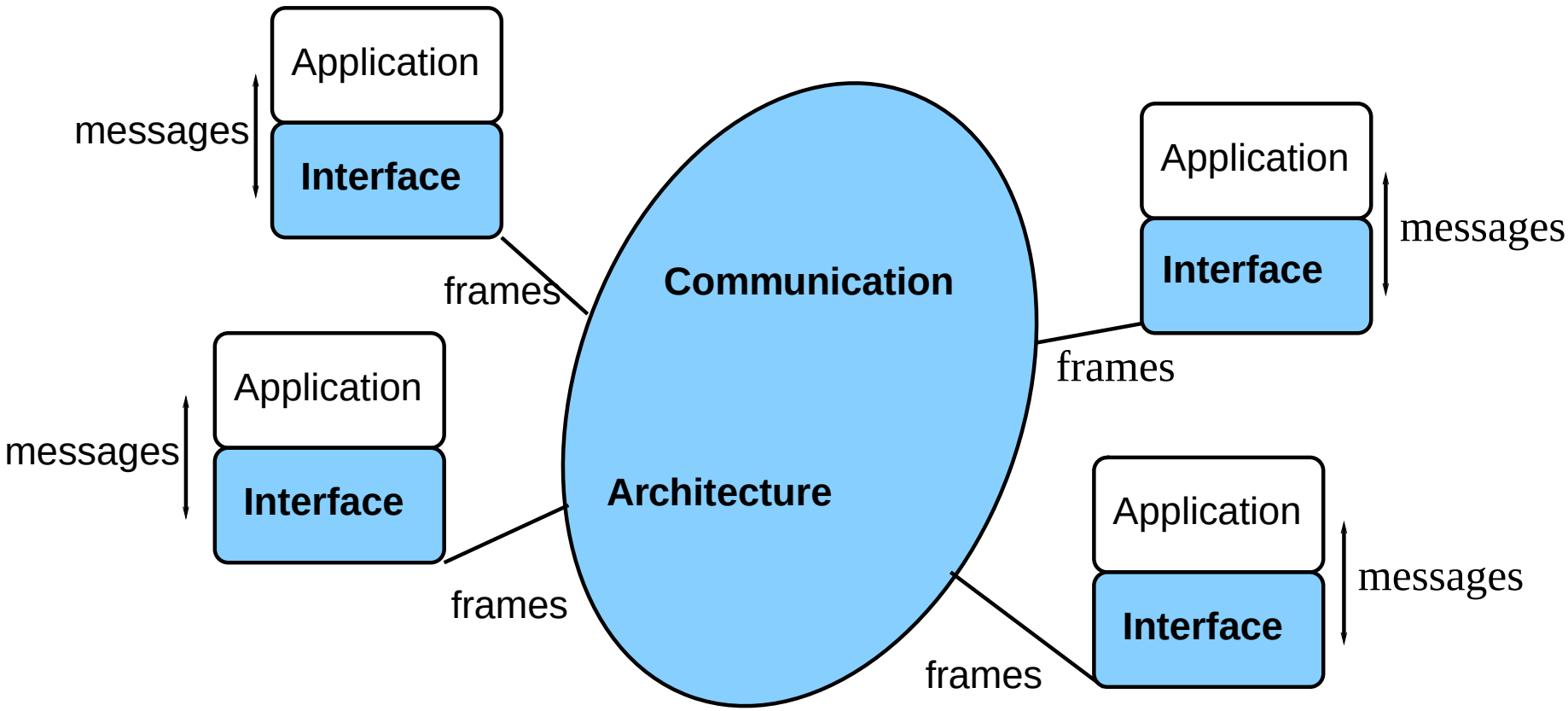
Outline

- Systems on a Chip
- Communication Principles
- $GeNoC$ Definition and Correctness
- Applications of $GeNoC$

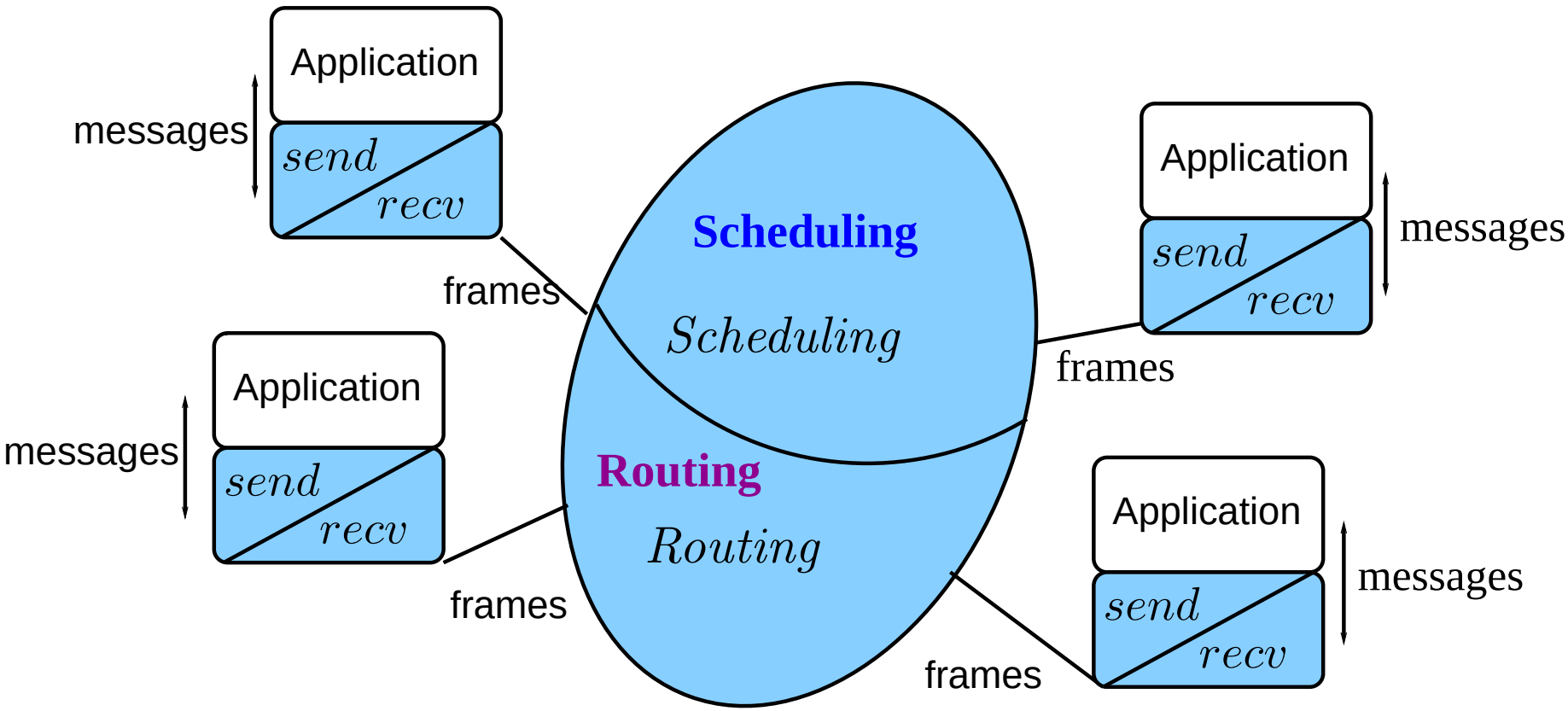
A Unique Model



A Unique Model

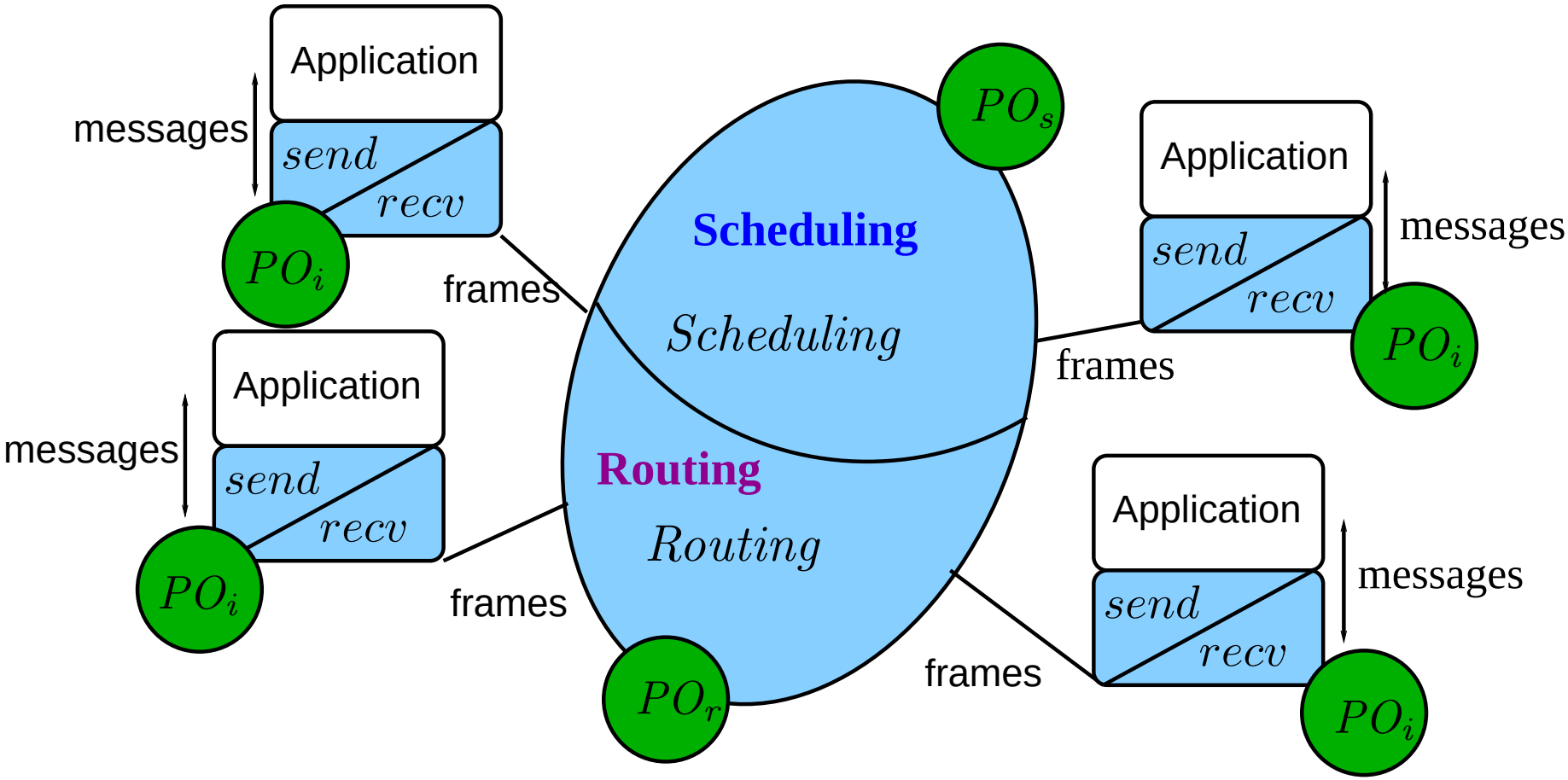


Functional Modeling

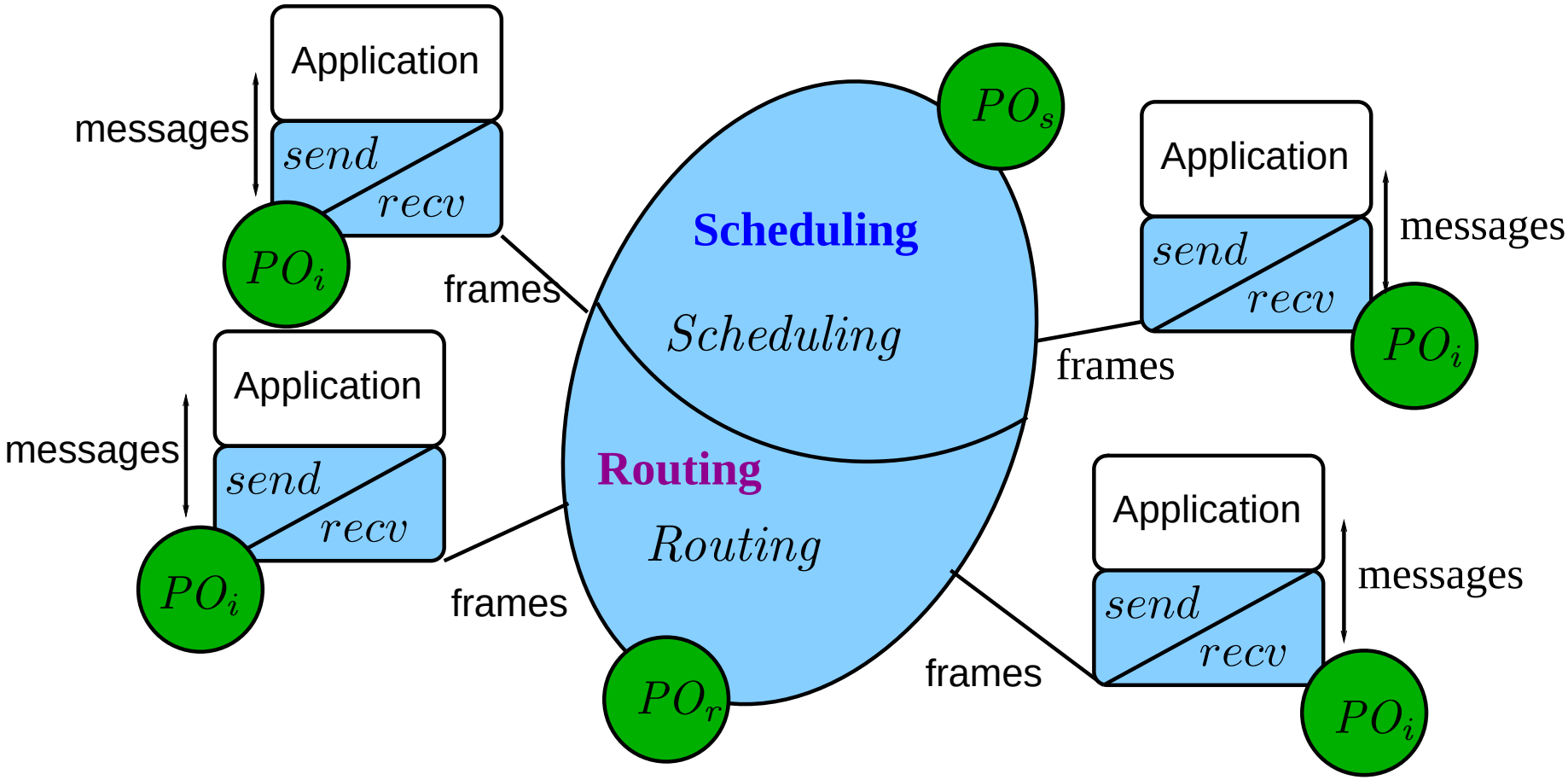


System $= \mathcal{F}(Routing, Scheduling, recv, send)$

Proof Obligations

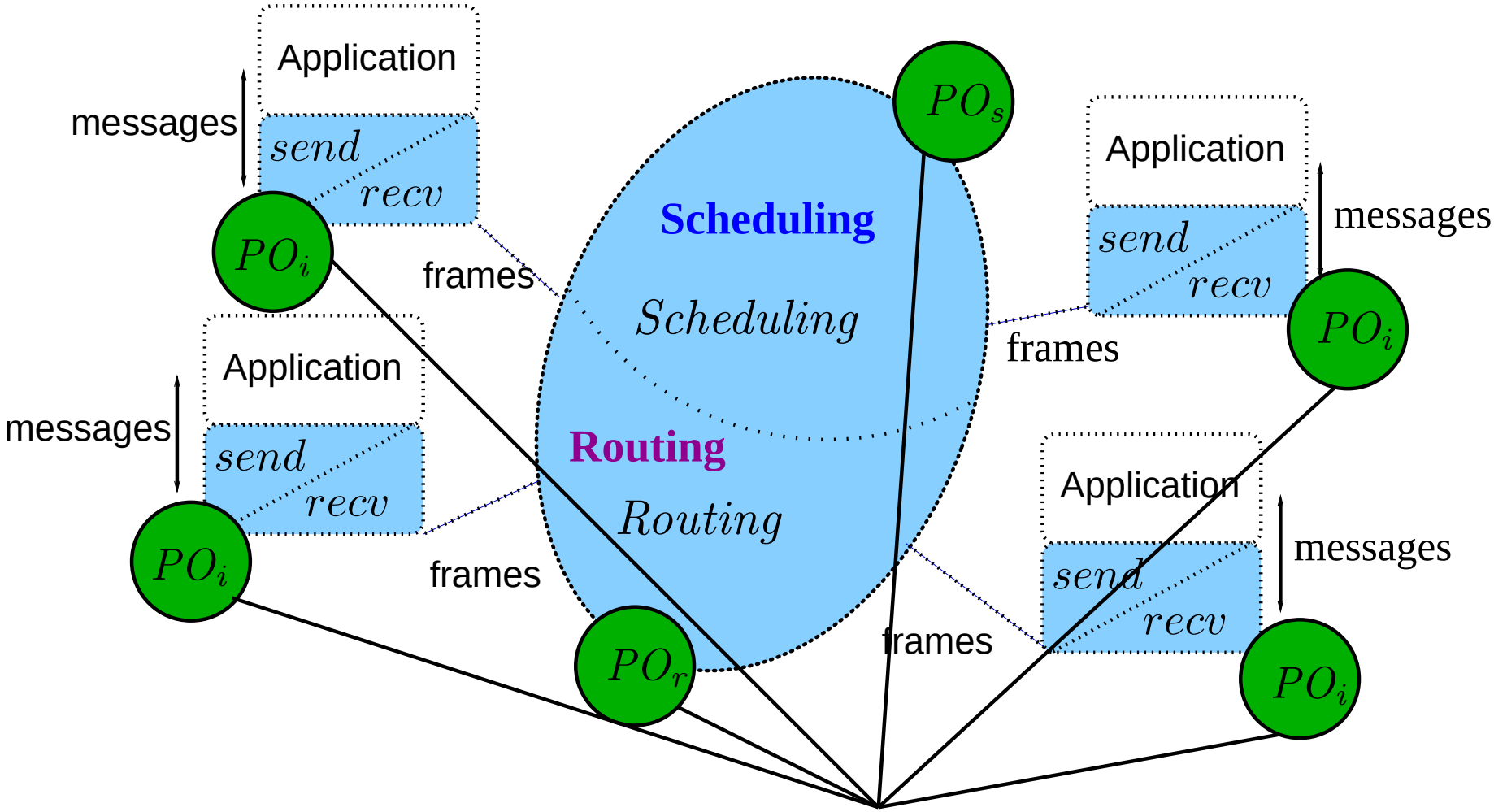


System Theorem



Thm: every message reaches its destination

System Theorem



Thm: every message reaches its destination

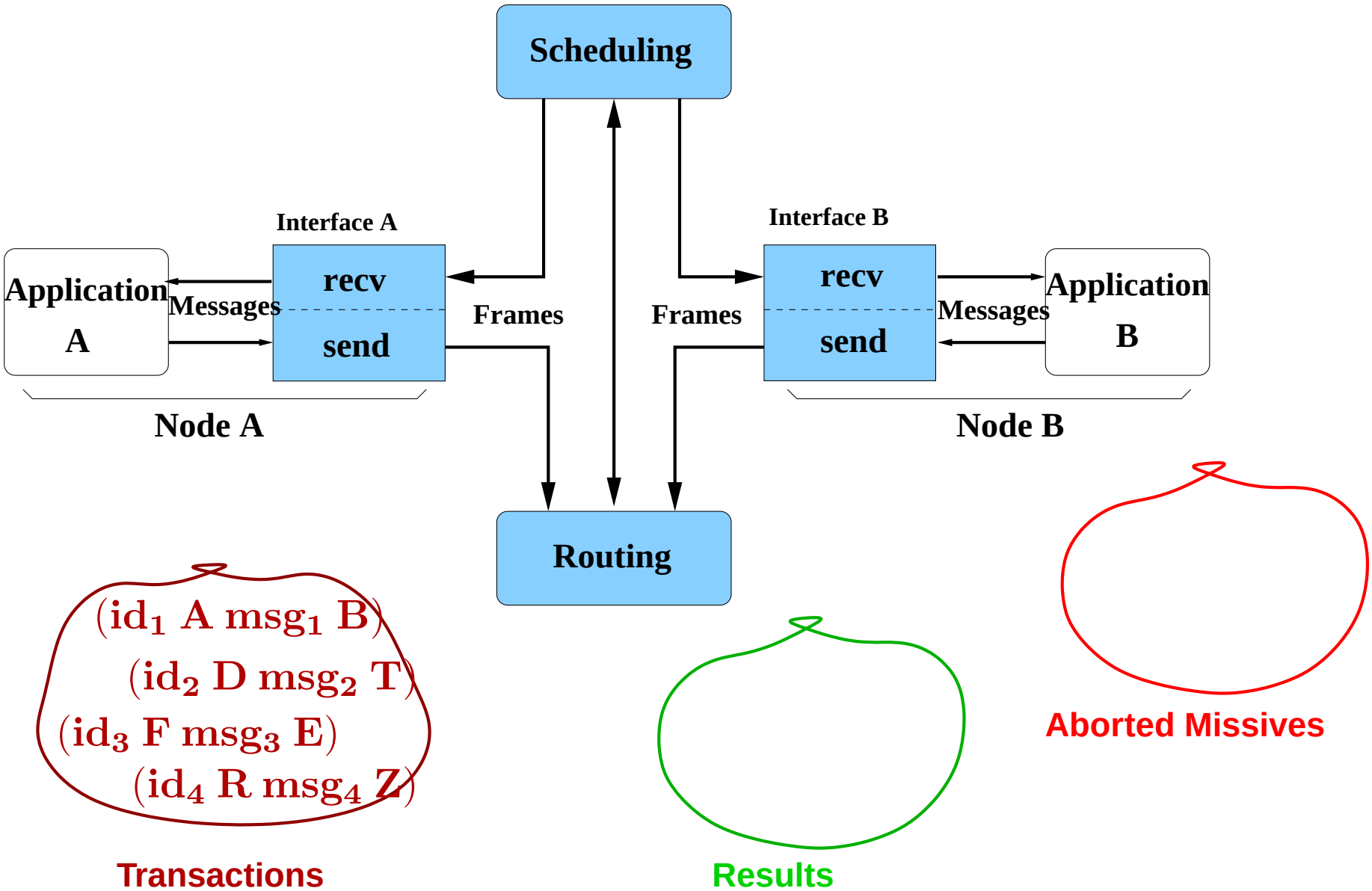
Outline

- Systems on a Chip
- Communication Principles
- $GeNoC$ Definition and Correctness
- Applications of $GeNoC$

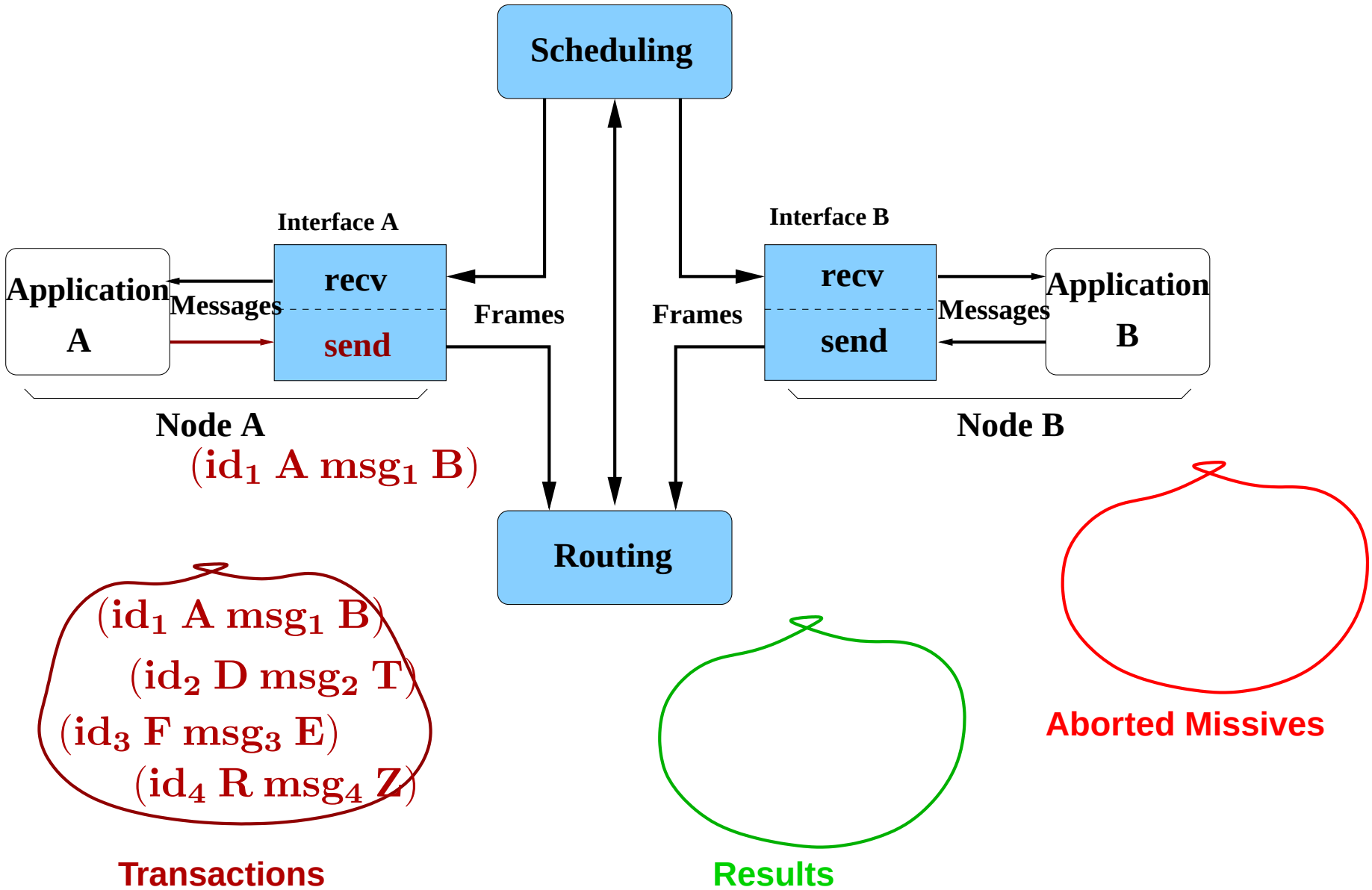
Overall Modeling Principles

- Function $GeNoC$
  - takes the list of pending communications
  - returns the list of results and the list of aborted communications
- Transactions
  - A transaction represents a pending communication, *i.e.* the intention of A of sending msg to B
  - It is a 4-tuple $(id\ A\ msg\ B)$

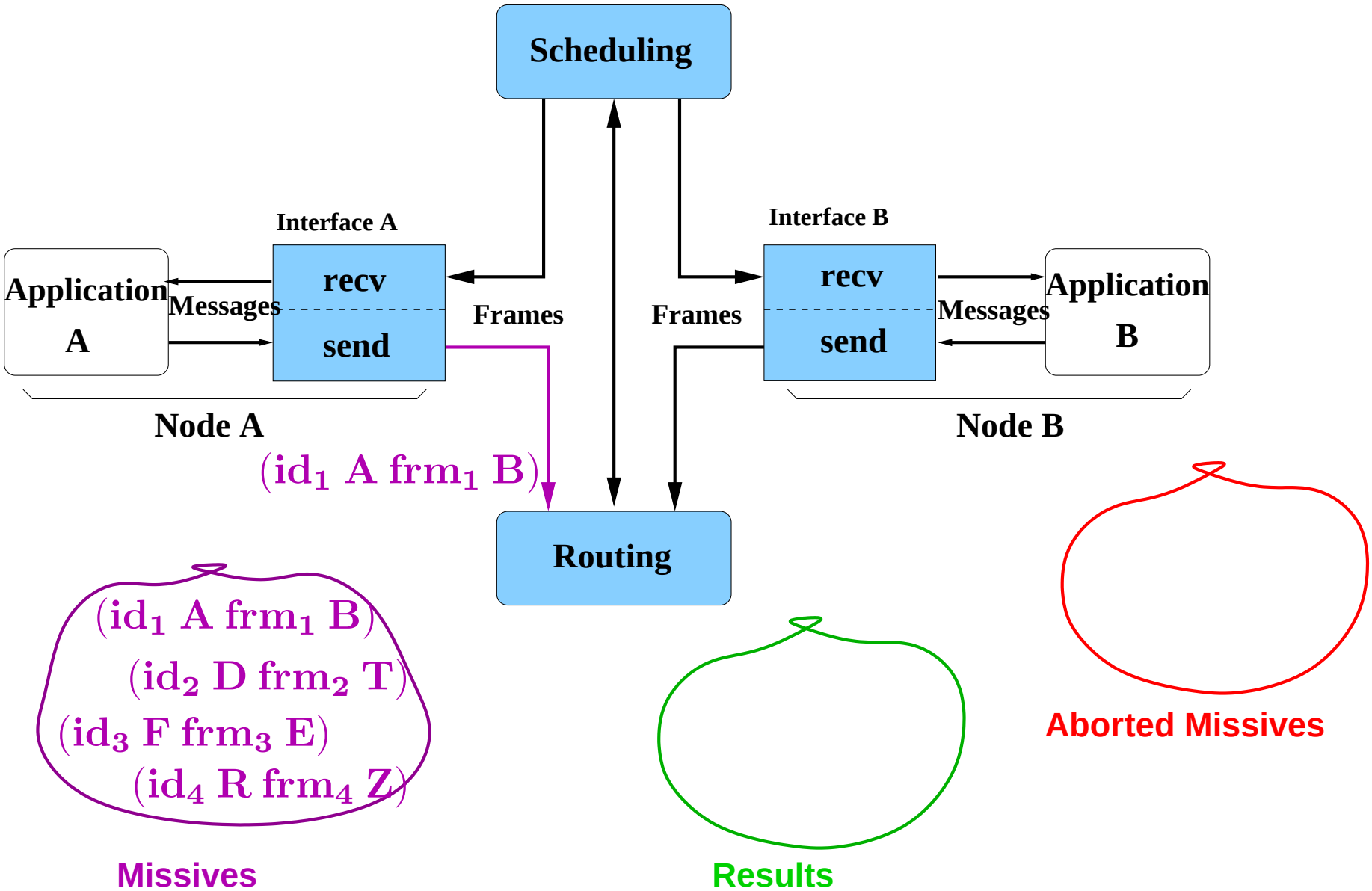
Function $GeNoC$



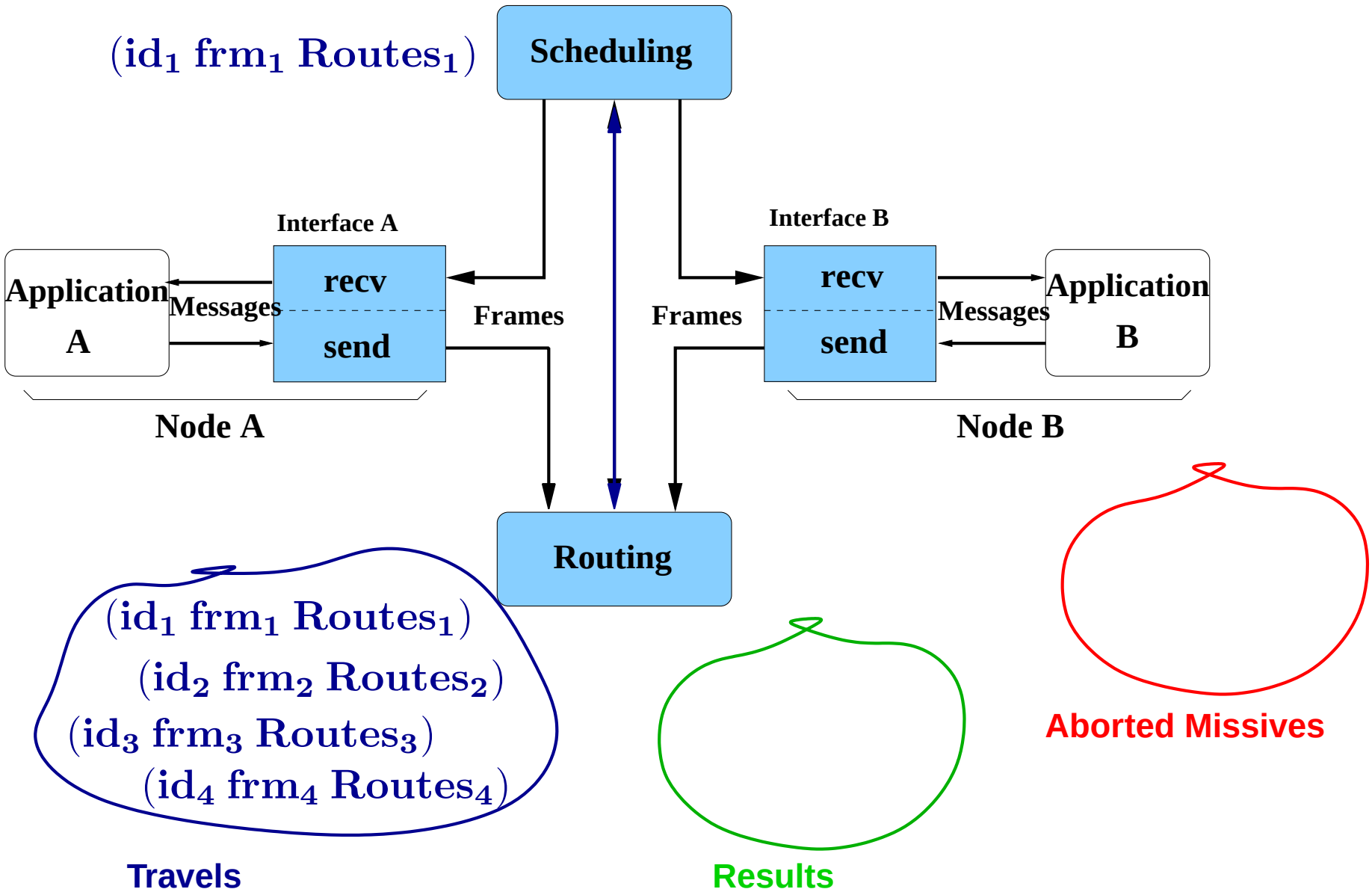
From transactions to missives



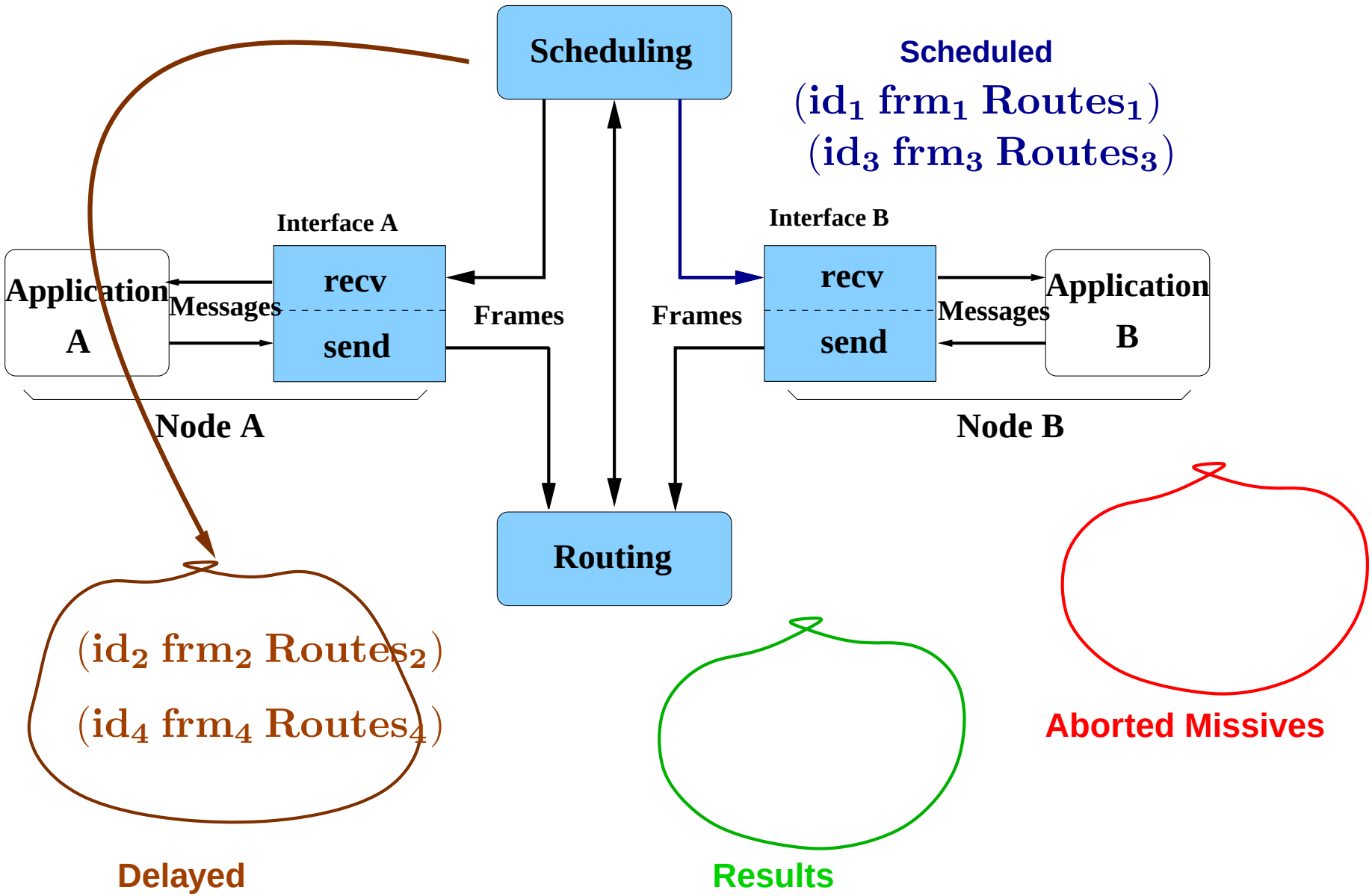
From transactions to missives



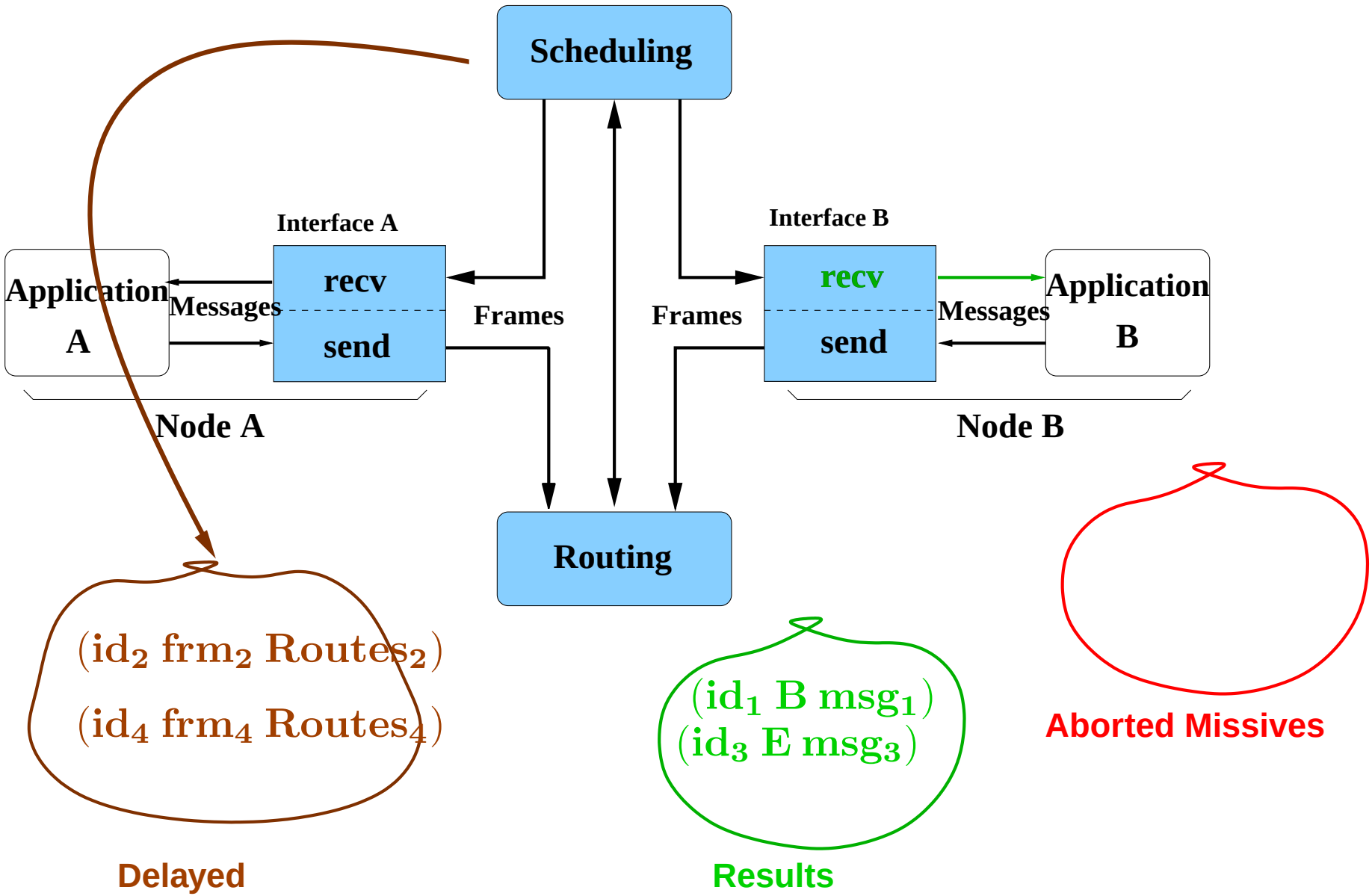
Routing Algorithm



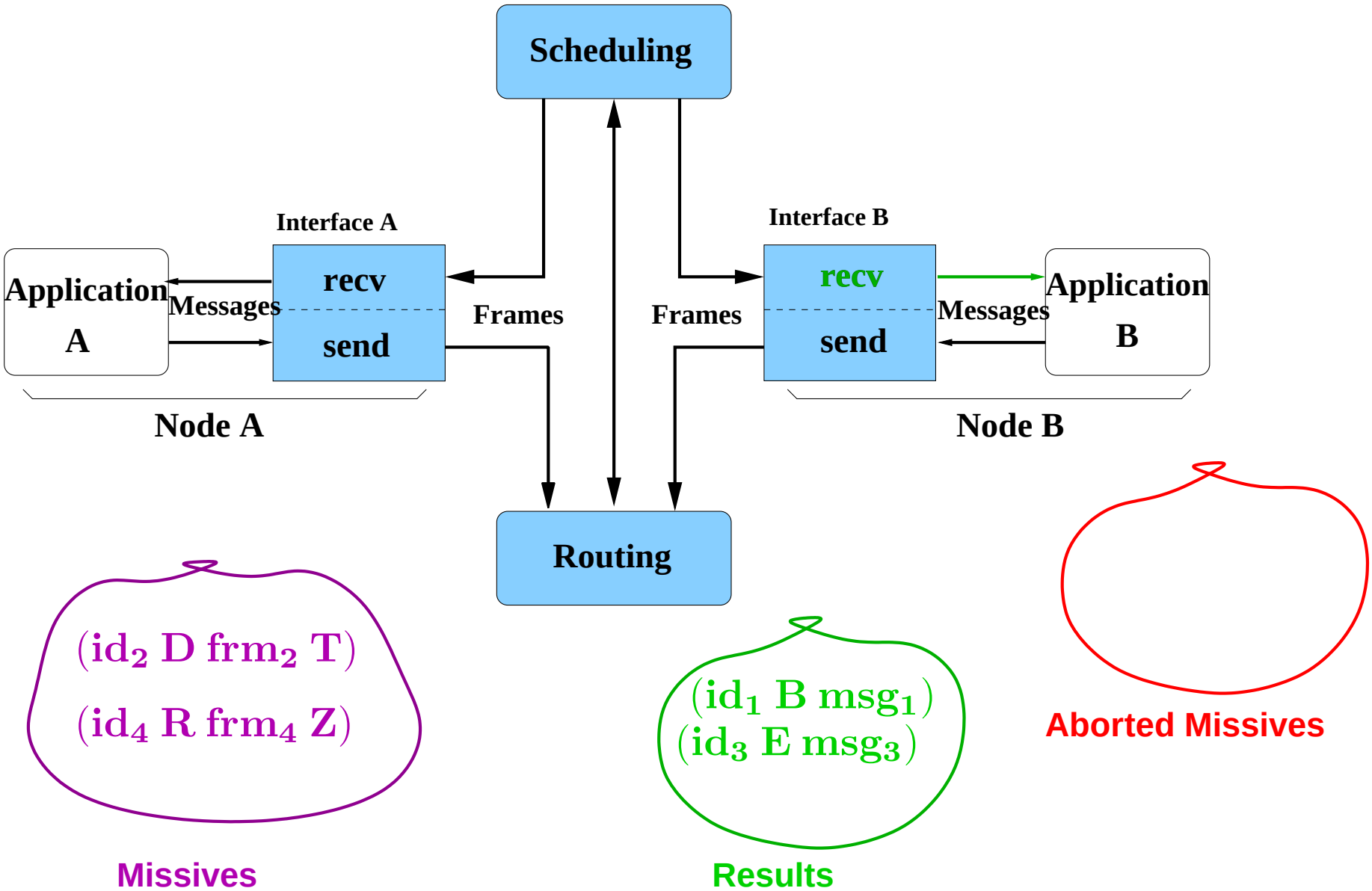
Scheduling Policy



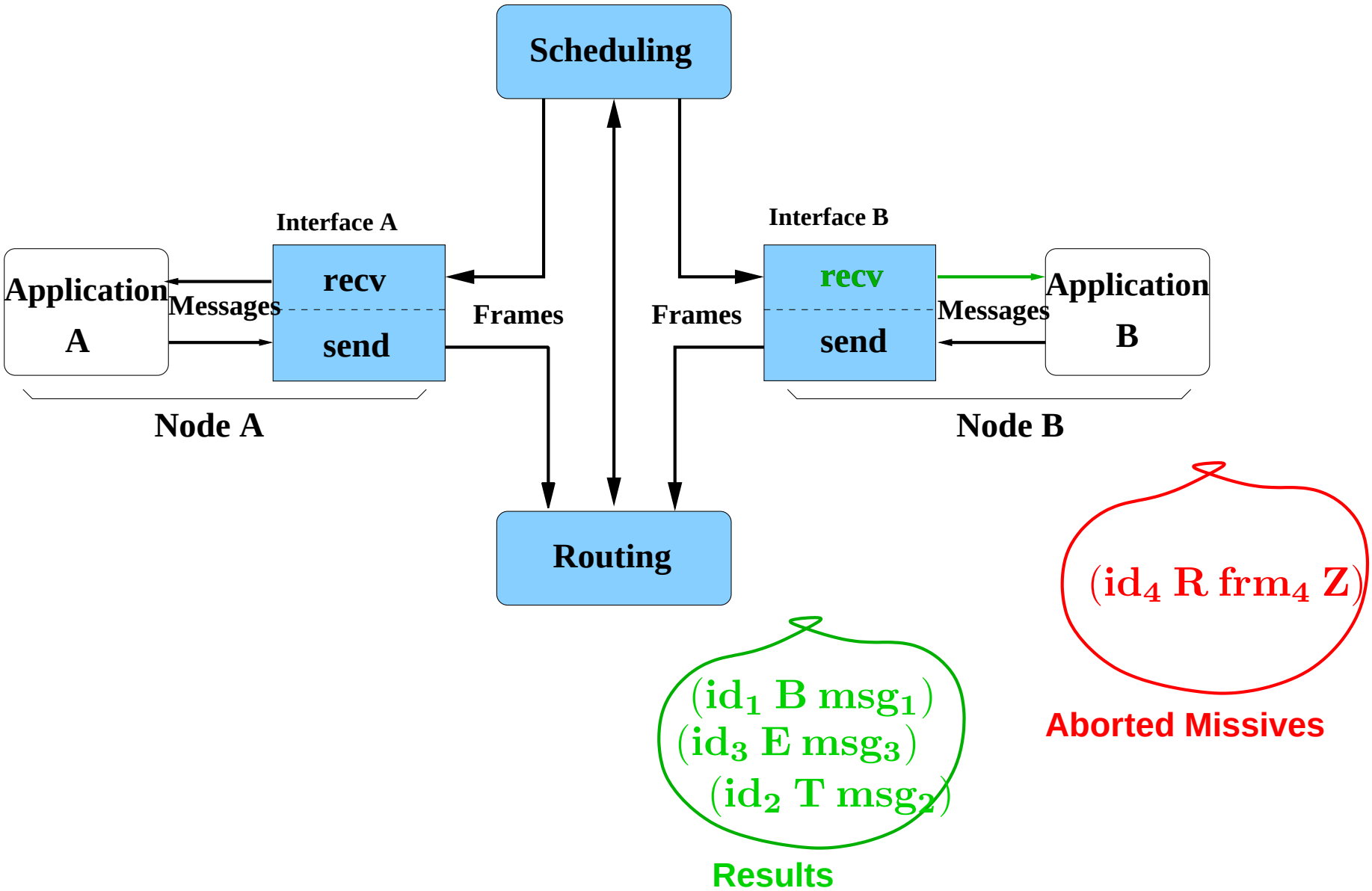
Results


Scheduling
Interface A
Interface B
Application A
Messages
recv
send
Frames
Frames
recv
send
Messages
Application B
Node A
Node B
Routing
$(\mathrm{id}_2\ \mathrm{frm}_2\ \mathrm{Routes}_2)$
$(\mathrm{id}_4\ \mathrm{frm}_4\ \mathrm{Routes}_4)$
$(\mathrm{id}_1\ \mathrm{B}\ \mathrm{msg}_1)$
$(\mathrm{id}_3\ \mathrm{E}\ \mathrm{msg}_3)$
Aborted Missives
Delayed
Results

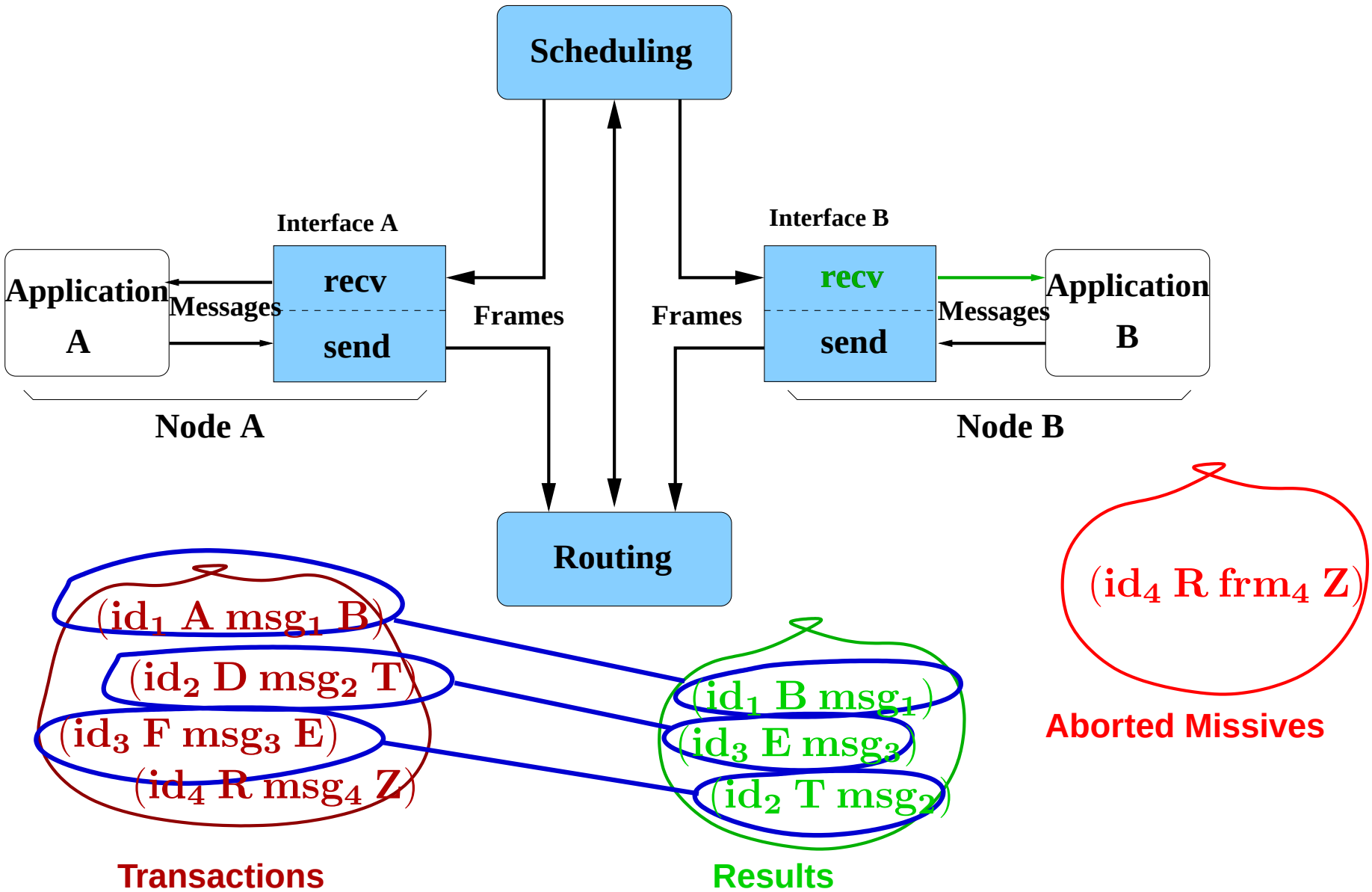
Aborted Missives



Aborted Missives



Correctness Criterion



Termination

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

- it is a prerequisite for mechanized reasoning (here ACL2)
- it is necessary to ensure liveness

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

Formal Definition

From a list of transactions, $\mathcal{T}$, the set of nodes $NodeSet$ and a list of attempt numbers att, function $GeNoC$ produces:

- The list $\mathcal{R}$ of results
- The list $\mathcal{A}$ for aborted missives

$$GeNoC: \ \mathcal{D}_{\mathcal{T}} \times GenNodeSet \times AttLst \rightarrow \mathcal{D}_{\mathcal{R}} \times \mathcal{D}_{\mathcal{M}}$$
$$(\mathcal{T}, NodeSet, att) \mapsto (\mathcal{R}, \mathcal{A})$$

Correctness Criterion

$$\forall res \in \mathcal{R},$$

$$\exists! trans \in \mathcal{T}, \begin{cases} & Id_{\mathcal{R}}(res) = Id_{\mathcal{T}}(trans) \\ \wedge & Msg_{\mathcal{R}}(res) = Msg_{\mathcal{T}}(trans) \\ \wedge & Dest_{\mathcal{R}}(res) = Dest_{\mathcal{T}}(trans) \end{cases}$$

For any result res, there exists a unique transaction $trans$ such that $trans$ and res have the same identifier, message, and destination.

Proof Obligations

- Interfaces
  - The composition $recv \circ send$ is an identity
- Routing $(id\ A\ frm\ B) \mapsto (id\ frm\ Routes)$
  - Missive/Travel matching
    - Same frame and identifier
    - Routes effectively go from the correct origin to the correct destination
- Scheduling
  - Mutual exclusion between $Scheduled$ and $Delayed$
  - No addition of new identifiers
  - Preserve frames and route correctness

Proof of the theorem

- Routing correctness + preserved by scheduling
  - $\longrightarrow$ right destination
- No modification on frames
  - $\longrightarrow$ every result is obtained by $recv \circ send$
- Interfaces correctness
  - $\longrightarrow$ received message = sent message
- Mutual exclusion between $Scheduled$ and $Delayed$ + no new identifiers
  - $\longrightarrow$ cut the proof in two parts

Outline

- Systems on a Chip
- Communication Principles
- $GeNoC$ Definition and Correctness
- Applications of $GeNoC$

Generic Routing Module

- Function $Routing$

$$Routing : \mathcal{D}_{\mathcal{M}} \times GenNodeSet \rightarrow \mathcal{D}_{\mathcal{V}}$$

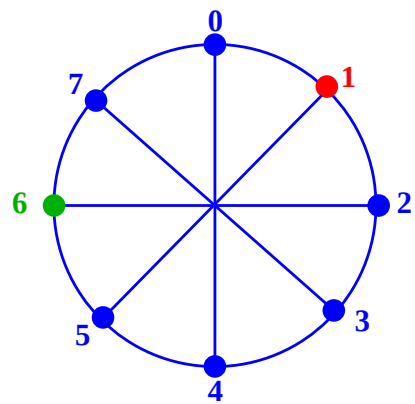
- Proof Obligations
  1. Termination
     - Distance decreases at each hop
  2. Correctness
     - From the right source to the right destination
     - Frames and identifiers are preserved

Octagon Modeling

- Unitary Moves:

$\mathcal{L}_{Oct}(s, d, N) \triangleq$

$$\begin{cases} d & \textbf{if } RelAd = 0 \\ Clockwise(s, 4N) & \textbf{if } 0 < RelAd \leq N \\ CounterClockwise(s, 4N) & \textbf{if } 3N \leq RelAd < 4N \\ Across(s, 4N) & \textbf{otherwise} \end{cases}$$



Octagon in $GeNoC$

- All Paths:
  $\rho_{Oct}(s, d, N) \triangleq$

$$\begin{cases} d & \textbf{if } s = d \\ s.\rho_{Oct}(\mathcal{L}_{Oct}(s, d, N), d, N) & \textbf{otherwise} \end{cases}$$

- Compliant Definition:
  $Routing_{Oct}(\mathcal{M}, N) \triangleq$

$$\forall m = (id\ A\ frm\ B),\ \text{build}\ v = (id\ frm\ \mathrm{R}_{Oct})$$

where

$$\mathrm{R}_{Oct} \equiv \rho_{Oct}(A, B, N)$$

Octagon Validation

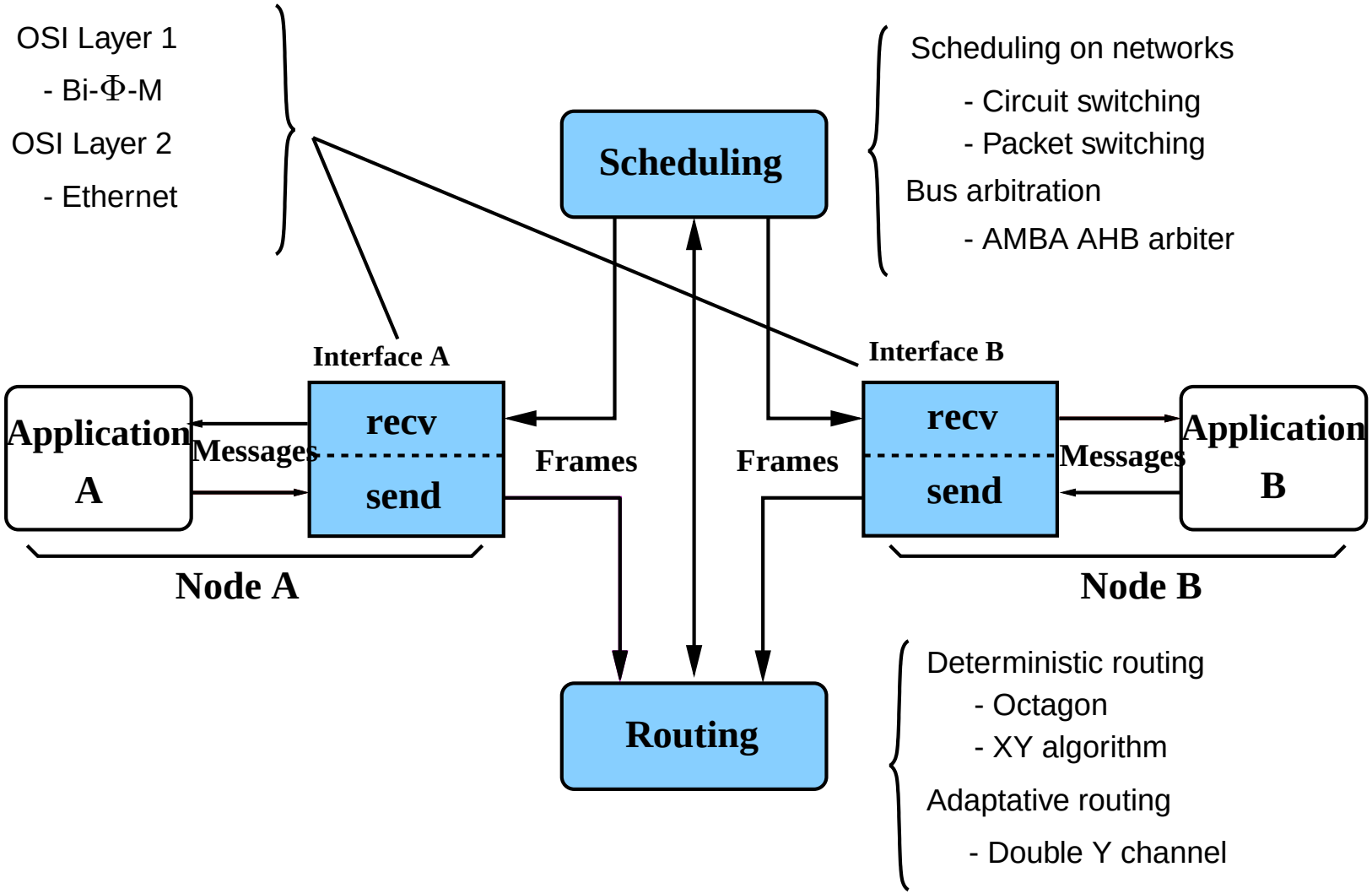
- Decreasing Measure:

$$Min[(d - s) \bmod 4N, (s - d) \bmod 4N]$$

- Routing Correctness:
  - $\rho_{Oct}(s, d, N)[0] = s \wedge Last(\rho_{Oct}(s, d, N)) = d$
  - Identifiers and frames are not modified

$$\forall v, \exists! m \in \mathcal{M}, \begin{cases} & Frm_{\mathcal{M}}(m) = Frm_{\mathcal{V}}(v) \\ \wedge & Id_{\mathcal{M}}(m) = Id_{\mathcal{V}}(v) \end{cases}$$

Applications of $GeNoC$



Conclusions

A generic model: $GeNoC$

- Identifies the essential constituents of **any** communication architecture
- Expression of the properties inherent in each one of them
- Formalizes the global property as a consequence of these properties
- Expressed in math *and* in the logic of ACL2

Perspectives: Extensions

- Master/Slave protocols
- Deadlocks (structural and protocol level)
- Adding queues and channels
  - wormhole routing in Hermes (TIMA, Grenoble, France)
- Adding time
  - Verisoft, very low level of automotive systems
- . . .

THANK YOU !!