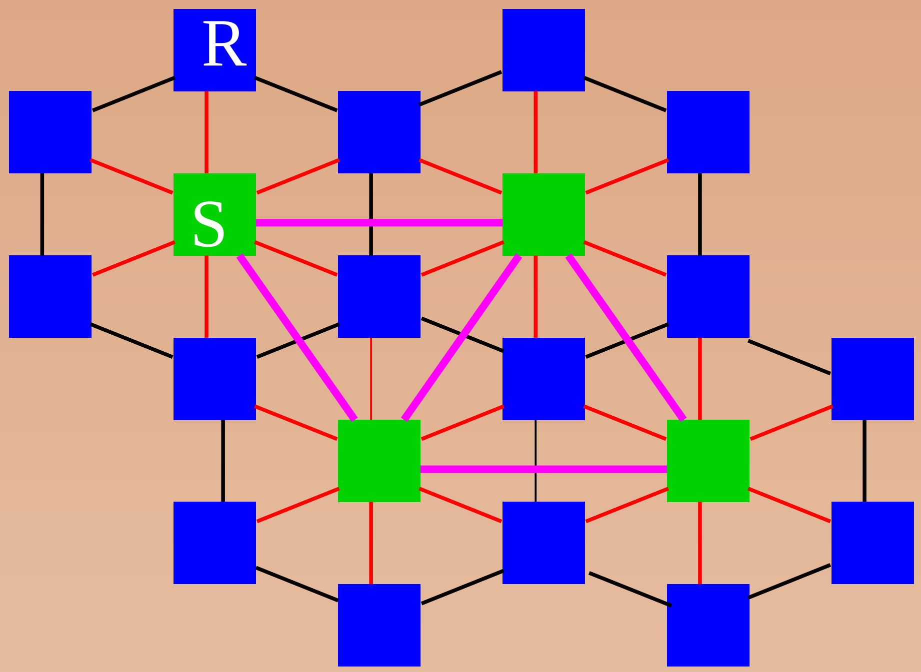
NOC Architecture

Axel Jantsch, Royal Institute of Technology, Stockholm

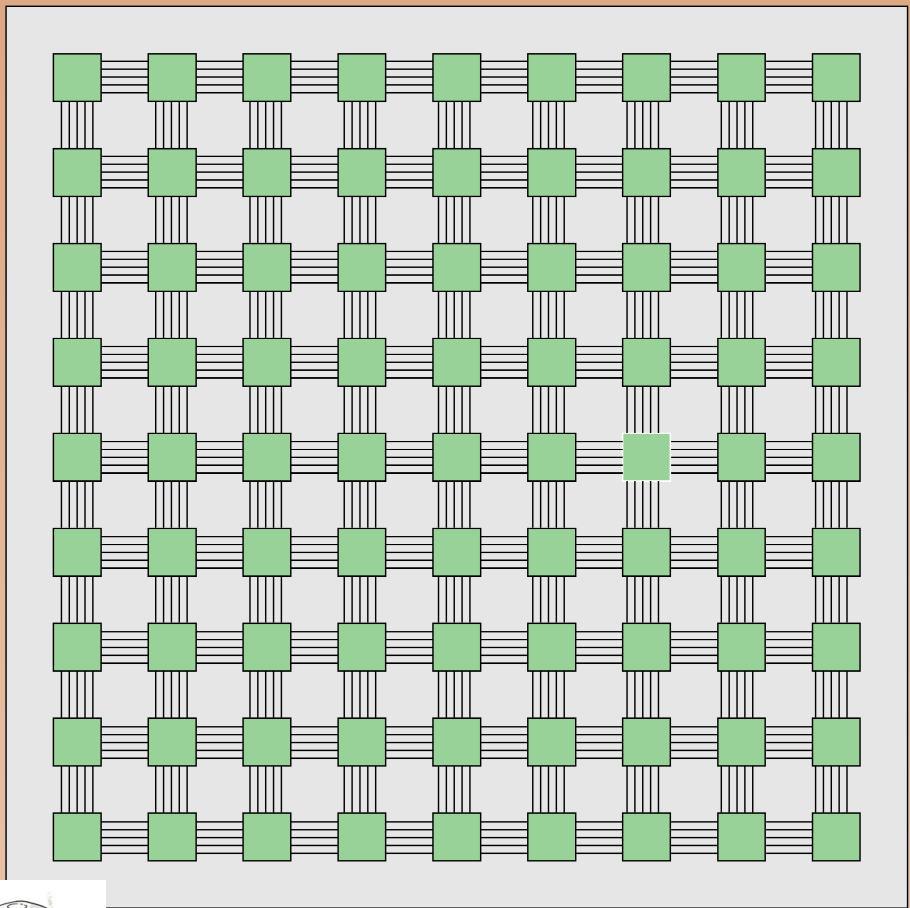
- Topology
- Switch Architecture
- Data link layer
- Network layer
- Transport layer
- Application layers
- Regions

Topology: Honeycomb



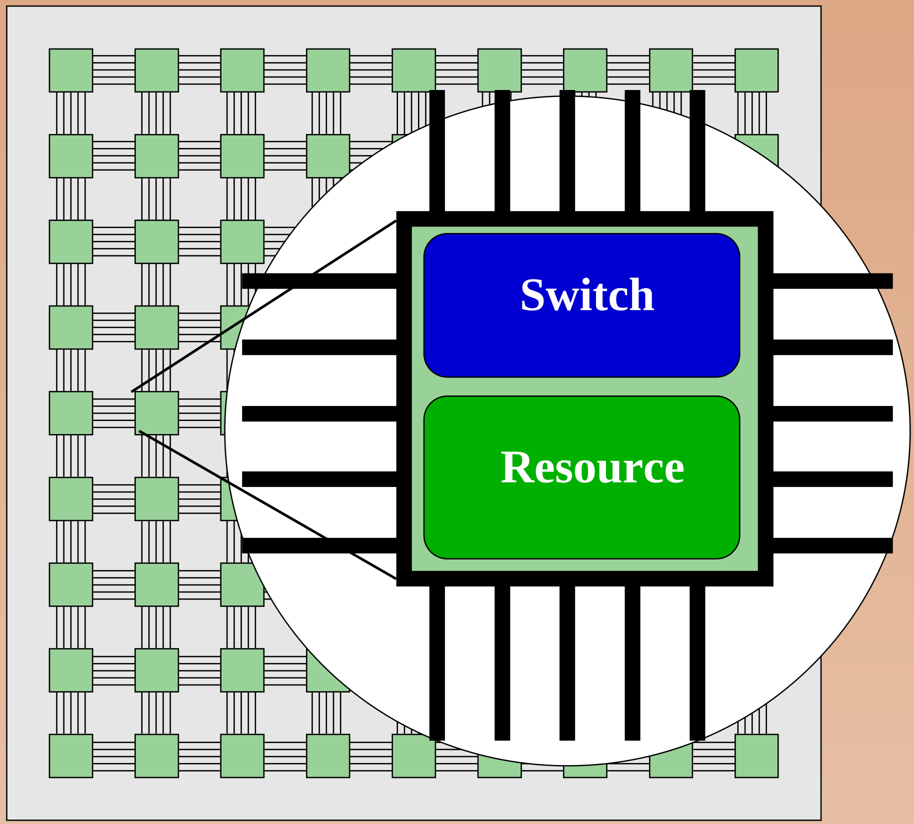
- Resource-to-switch ratio: 3
- A switch is connected to 6 resources and 6 switches
- A resource is connected to 3 switches and 3 resources
- Wiring intense topology
- Max number of hops grows with $n/3$

Topology: Mesh

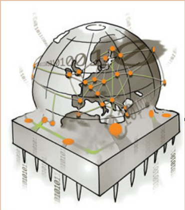


- Resource-to-switch ratio: 1
- A switch is connected to 4 switches and 1 resource
- A resource is connected to 1 switch
- Max number of hops grows with $2n$

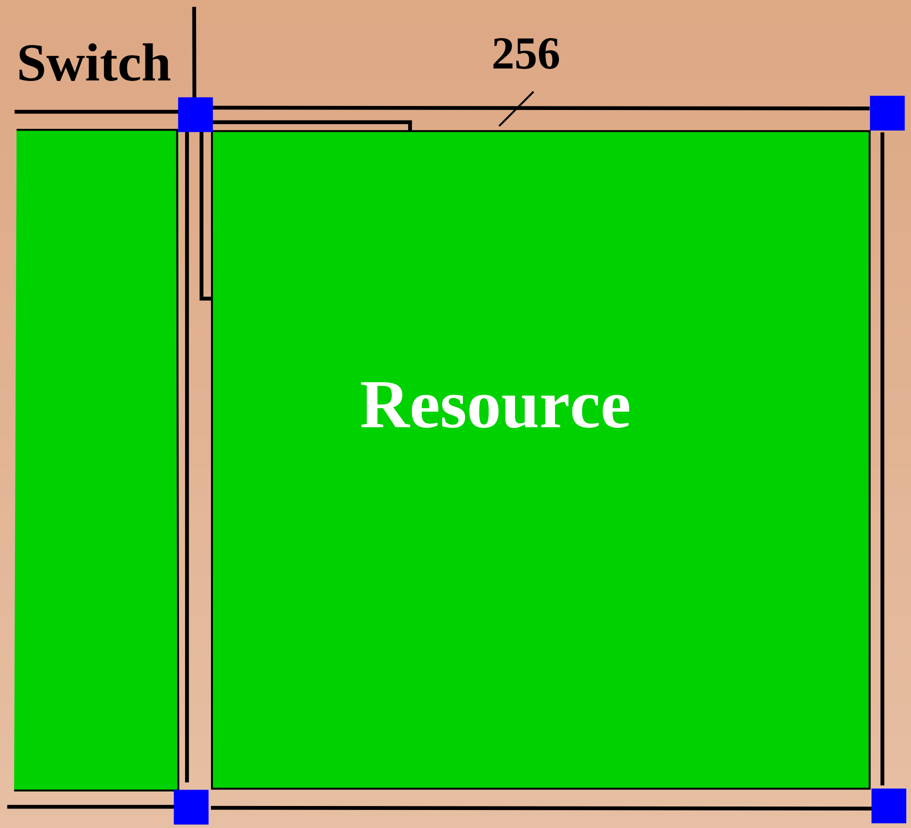
The Node in a Mesh



- How large are resources and the switches?
- What is the best geometry of switch and resource?
- How many wires and how long?



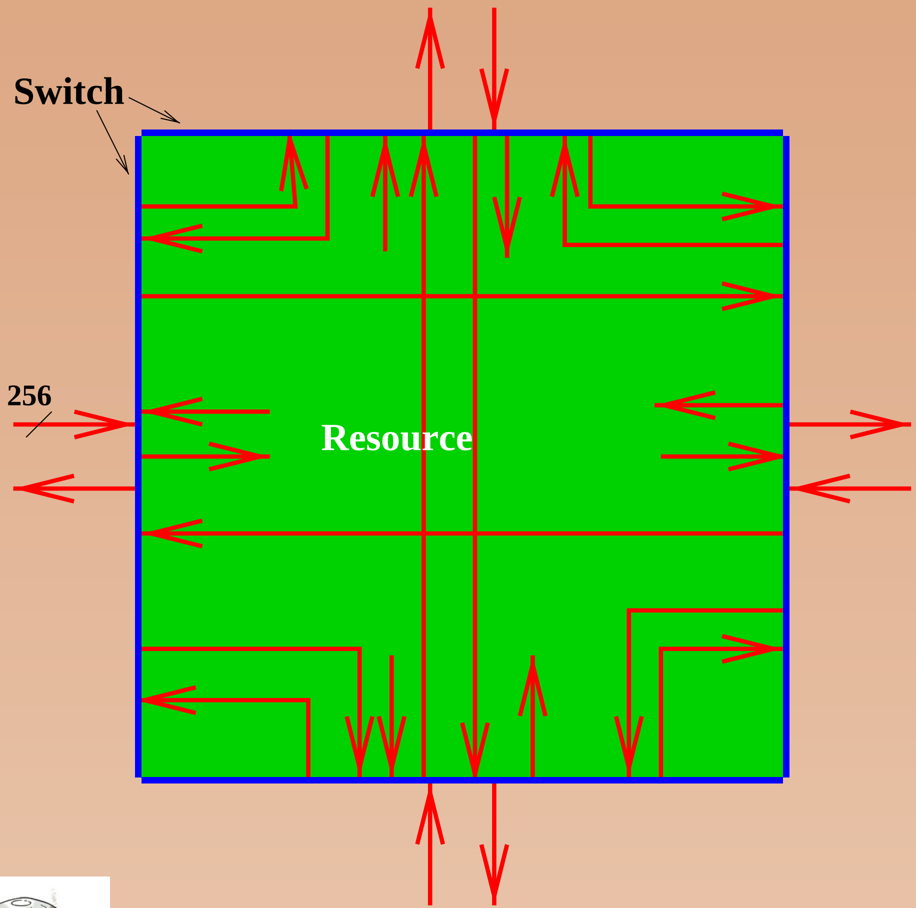
Square Switch



Scenario:

- $60nm$ CMOS
- $22mm \times 22mm$ chip size
- $300nm$ minimal wire pitch
- $2mm \times 2mm$ resource
- $100\mu m \times 100\mu m$ switch
- switch-to-switch connection: 256 wires
- switch-to-resource connection: 256 wires

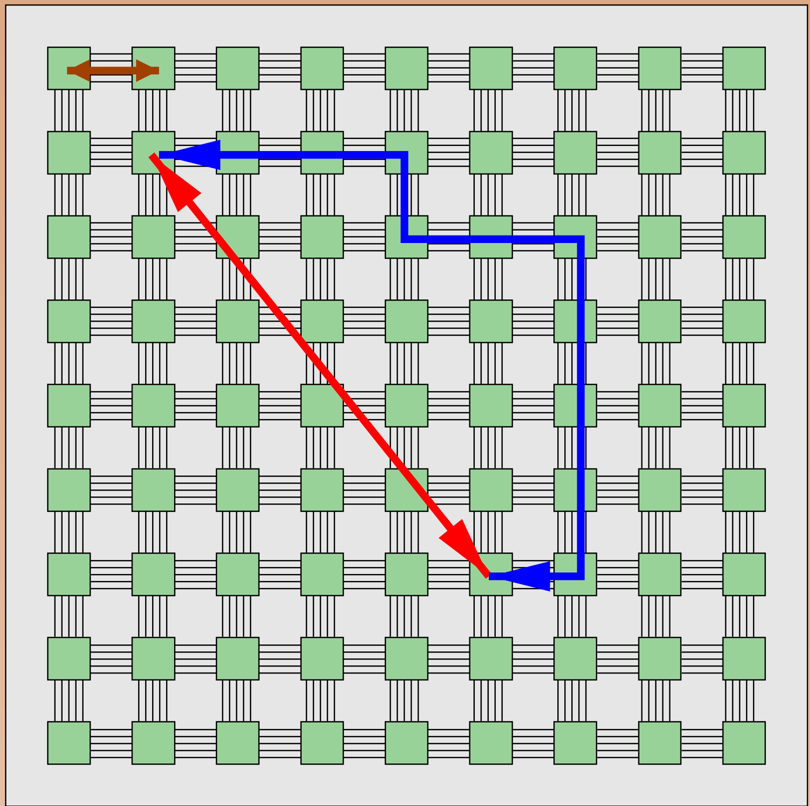
Thin Switch



Scenario:

- $60nm$ CMOS
- $22mm \times 22mm$ chip size
- $600nm$ minimal wire pitch for top layers
- $2mm \times 2mm$ resource
- $4 \times 50\mu m \times 2000\mu m$ switch
- switch-to-switch connection: 512 wires
- switch-to-resource connection: 512 wires

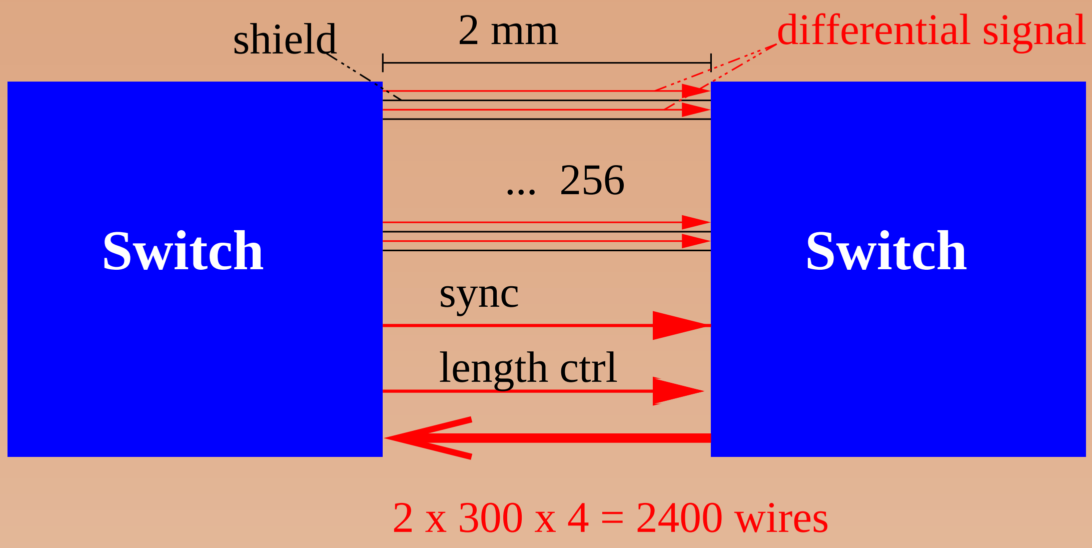
Communication is Key on Several Levels



Communication Layers and unit of communication:

- Physical layer: Word
- Data link layer: Cell
- Network layer: Packet
- Transport layer: Message
- Application layer

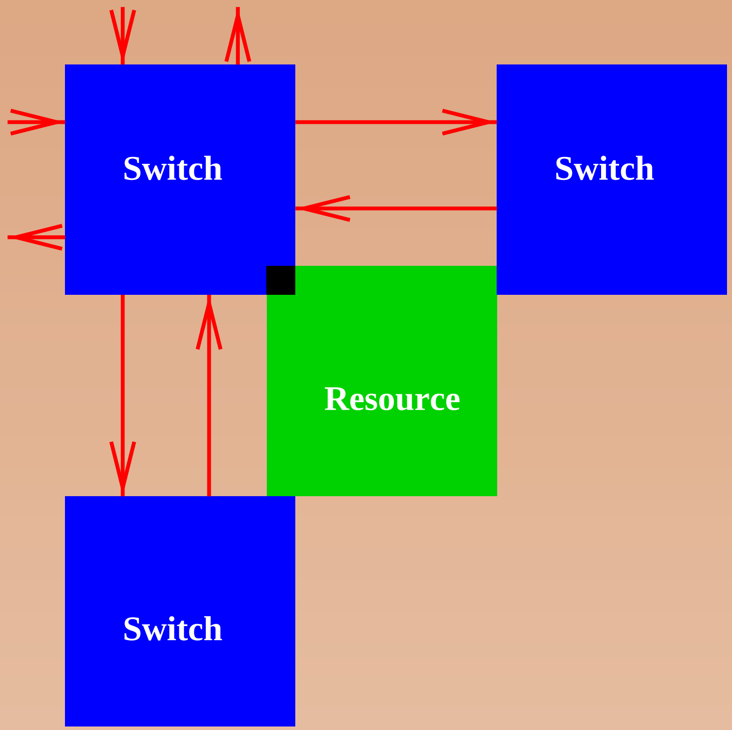
Physical Layer



Parameters:

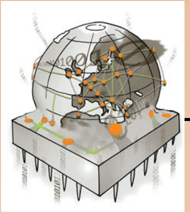
- Physical distance
- Number of lines
- Activity control
- Buffers and pipelining

Data Link Layer

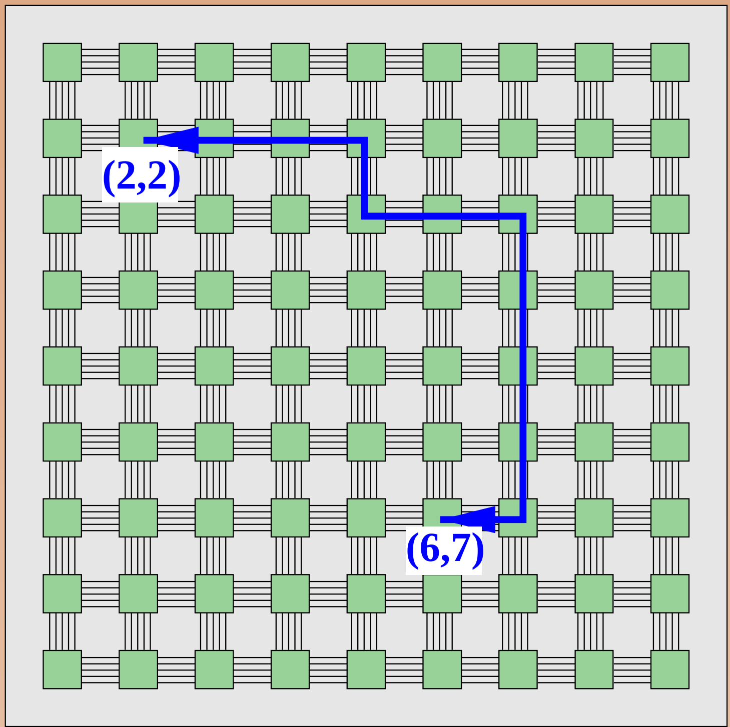


Parameters:

- Line frequency versus switch frequency (word vs cell)
- Buffering
- Error correction
- Power optimization; e.g. avoid activity and power optimized encoding

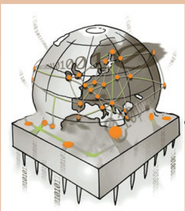


Network Layer

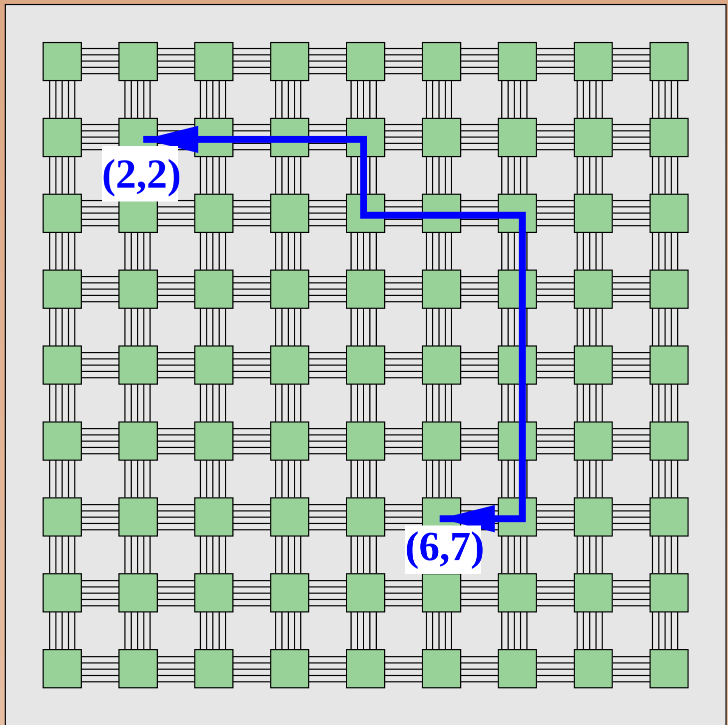


Parameters:

- cell size versus packet size
- Network address scheme, e.g. $4 + 4$ bit for 16×16 resources
- Routing algorithm
- Priority classes: e.g. 2 classes:
  1. high priority, fixed delay cells
  2. low priority, best effort delay cells
- Error correction

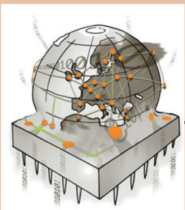


Transport Layer

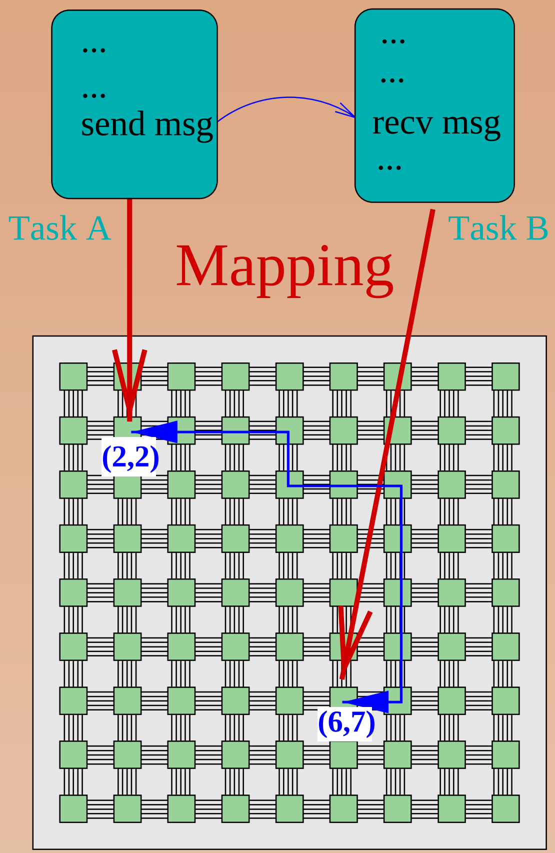


Parameters:

- Message size versus cell size
- Virtual channels with traffic profiles
- Signalling
- Priority classes of channels, e.g.
  1. constant bit rate traffic
  2. varying bit rate traffic
- Network resource management
- Error correction



Application Layers



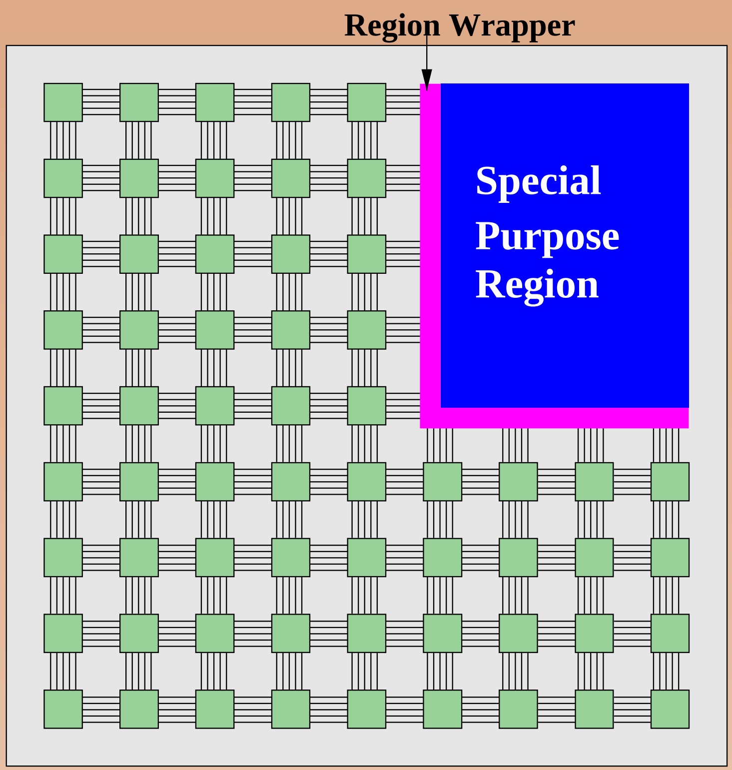
Interprocess communication at the task level:

- `send / receive` for individual messages
- `open; write/read; close` for channel based communication

Mapping issues:

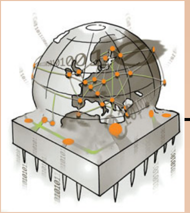
- Assigning tasks to resources
- Translating task addresses to resource/task addresses
- Establishing and closing channels
- Static allocation versus dynamic allocation

Regions



Parameters:

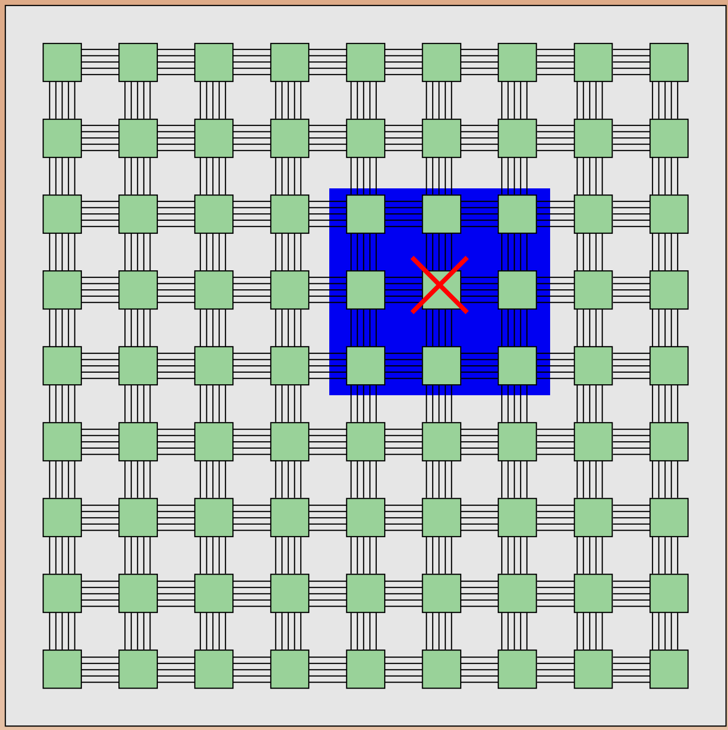
- Not all resources have the same size; e.g.
  - ⋆ Memory
  - ⋆ FPGA regions
  - ⋆ Special purpose architectures like multiprocessors
  - ⋆ mixed signal parts
- A region wrapper makes the region transparent
- The region wrapper can be
  - ⋆ at several protocol layers,
  - ⋆ in hardware or software
  - ⋆ local or distributed



NOC Operating System

- Monitoring
  - ⋆ Utilization of resources and switches
  - ⋆ Power consumption
  - ⋆ Statistics (errors, cells, etc.)
- Communication services (transport, presentation and application layers)
- Resource allocation and load migration
- Diagnostics and fault recovery
- Power management
- Development support services
  - ⋆ Libraries for run-time services
  - ⋆ Compilers, linkers and simulators

Dynamic Fault Handling Example



Assumptions:

- NOC-OS makes run-time diagnostics
- NOC-OS can identify faulty parts
- Network layer protocol can be reconfigured at run-time

If a resource becomes faulty:

1. NOC-OS detects a faulty resource
2. NOC-OS defines a new region to isolate the faulty resource
3. NOC-OS reconfigures the Network layer protocol to route around the faulty resource

Conclusion

- The NOC Architecture defines
  - ⋆ the communication infrastructure
  - ⋆ the resource-to-network interface
  - ⋆ the network services
- Strict layering of communication protocols and services allows
  - ⋆ the separation of the network backbone from the resources
  - ⋆ combination of different features and functionality at different levels
  - ⋆ the customization of a generic platform into an efficient product