Communication Refinement for a Network-on-Chip Platform

Axel Jantsch and Zhonghai Lu
Royal Institute of Technology, Stockholm

April 2003

Overview

Nostrum Architecture and Platform

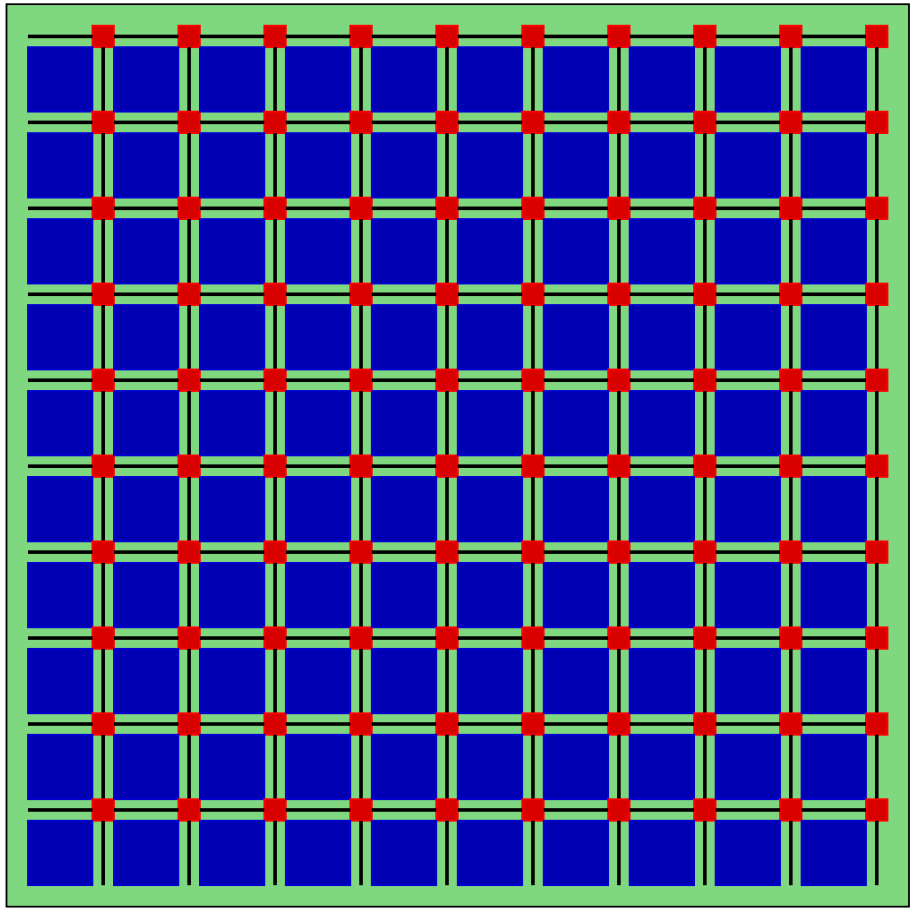
Communication Stack

Communication Patterns

Refinement

Summary

Nostrum Topology: Mesh



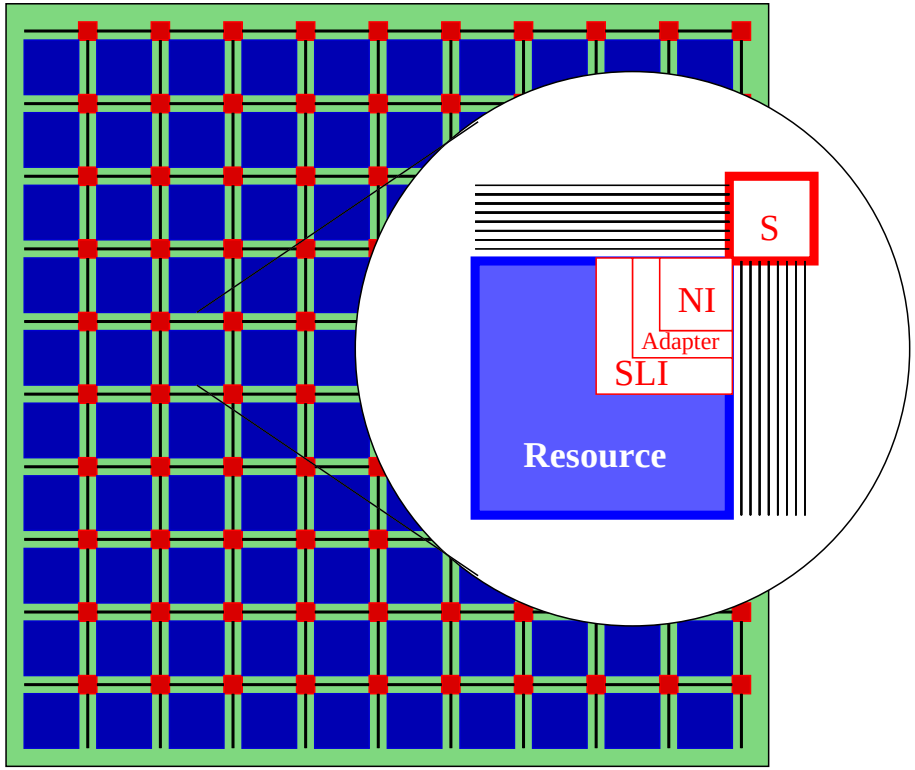
Characteristics:

- 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 $2n$

Motivation:

- Regularity of layout; predictable electrical properties
- Expected locality of traffic

The Node in a Mesh



NI: Network Interface:

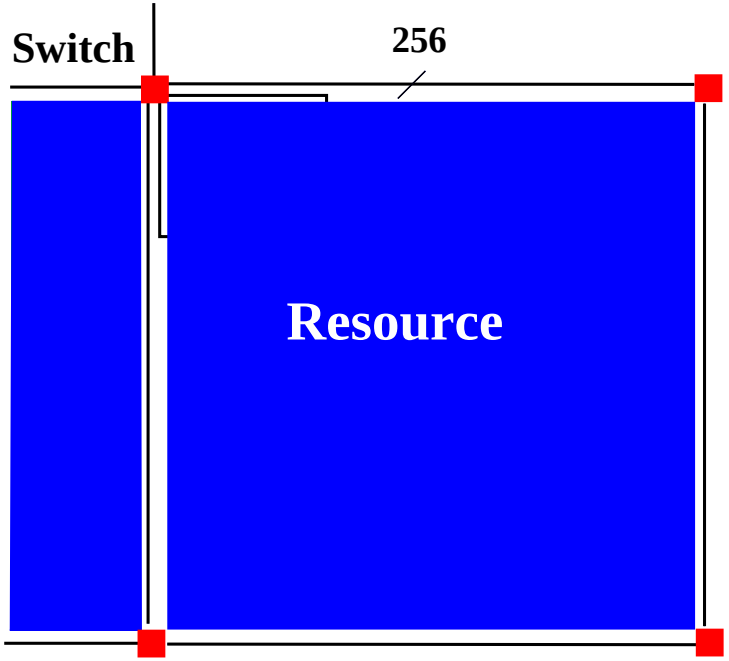
- Compulsory
- HW
- Implements the network layer protocol

Adapter: Resource specific interface circuit;

SLI: Session Layer Interface:

- Optional
- Hardware and/or software
- Implements the session layer protocol

Node Geometry



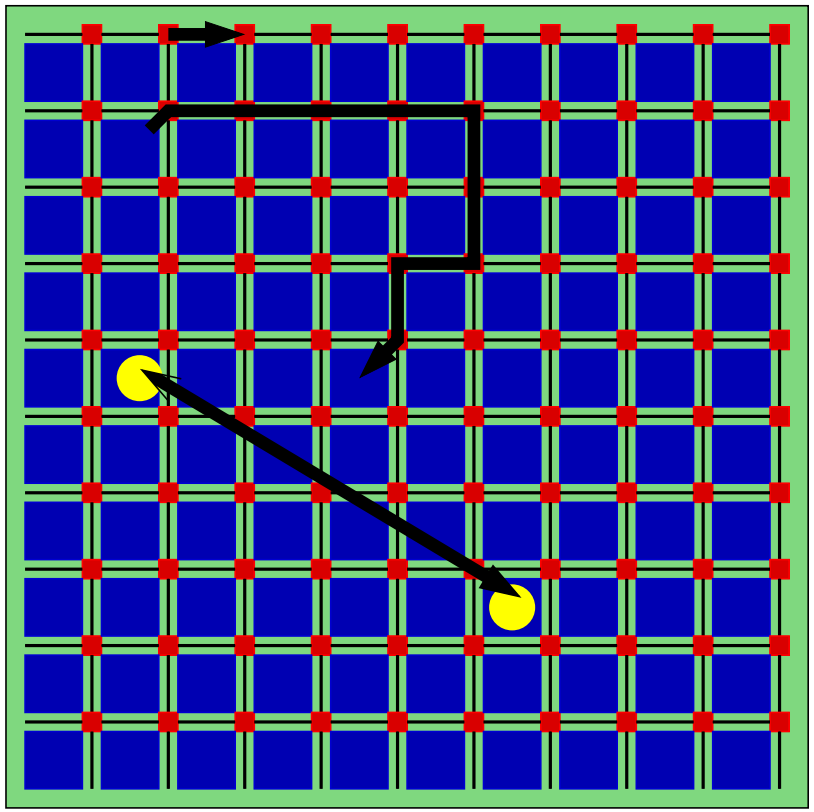
Scenario:

- $60nm$ CMOS
- $22mm \times 22mm$ chip size
- $300nm$ minimal wire pitch
- $2mm \times 2mm$ resource
- $100\mu m \times 100\mu m$ switch
- $\Rightarrow 1333$ wires on four metal layers
- switch-to-switch connection: 256 shielded and differential data signals;
- switch-to-resource connection: 256 data signals

Nostrum Platform

- Communication Infrastructure
- Resource management and arbitration services
- Design methodology

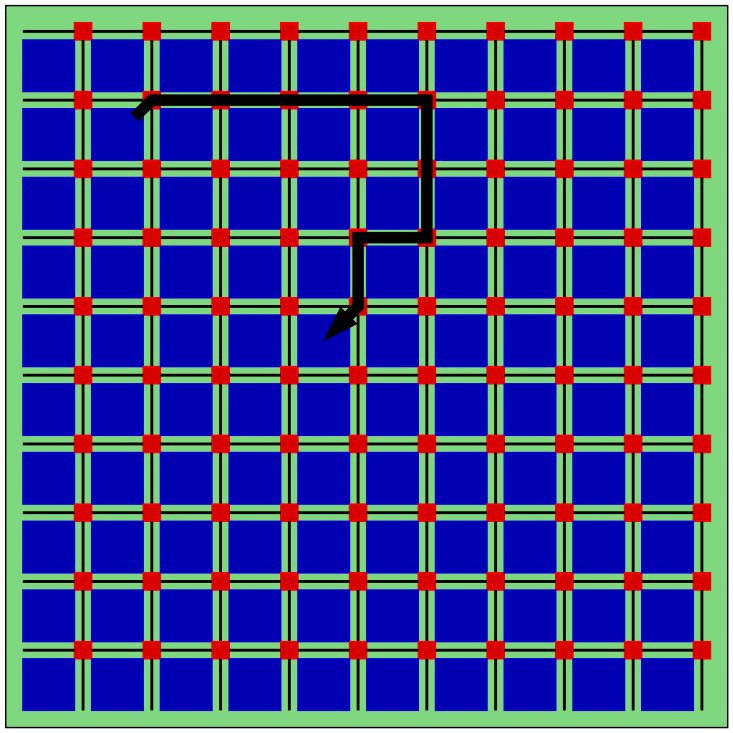
Nostrum Protocol Stack



Communication Layers:

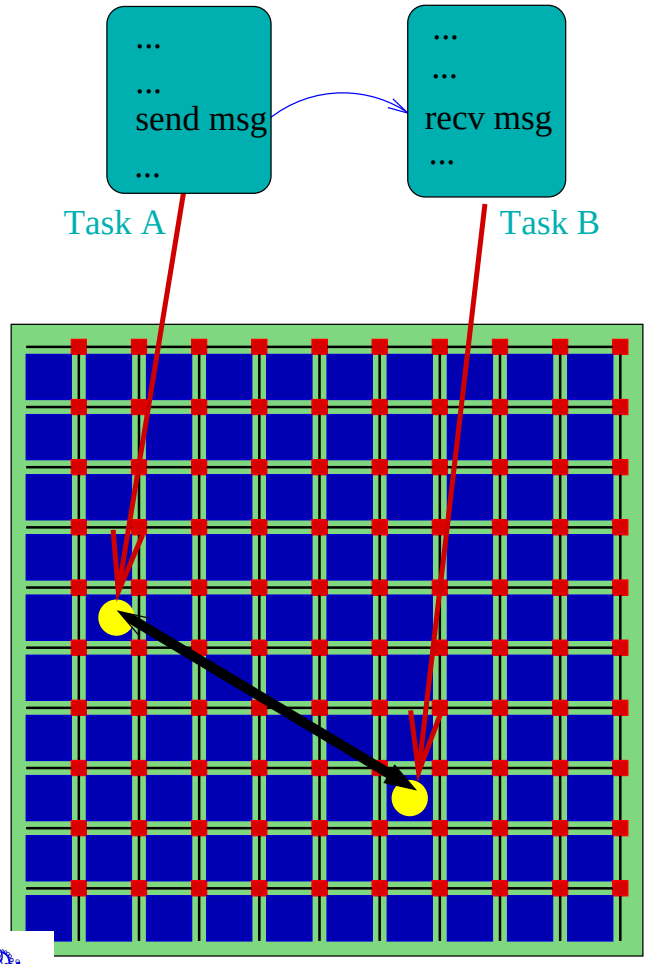
- Physical layer: switch-to-switch and switch-to-resource
- Data link layer: switch-to-switch and switch-to-resource
- Network layer: resource-to-resource
- Session layer: process-to-process
- Application layer: application-to-application

Network Layer



- Link layer packet = network layer packet
- Lossless communication
- Best effort service:
  - ⋆ Relative x-y addressing
  - ⋆ Out-of-order packet arrival
  - ⋆ Deflective routing with no buffers and no routing tables
- Virtual circuits with guaranteed bandwidth; varying latency
- Virtual circuits with guaranteed latency
  - ⋆ Circuit build-up and tear-down
  - ⋆ In-order packet arrival
  - ⋆ Addressing by circuit identifier

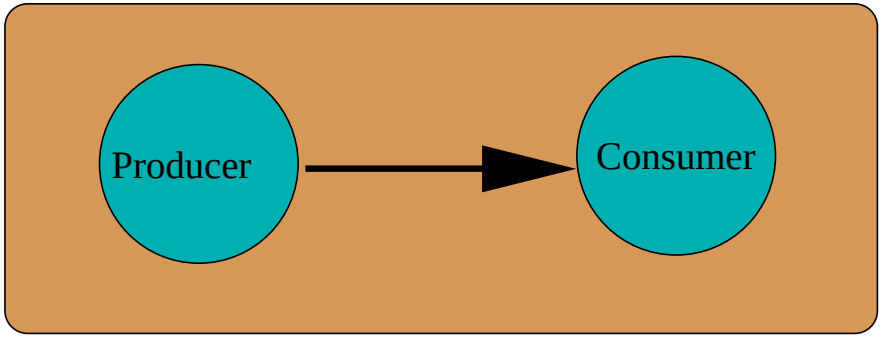
Session Layer



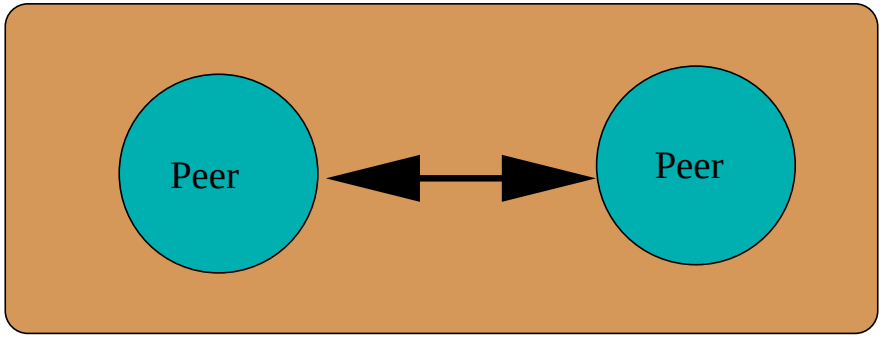
- Message passing communication:
  - ⋆ `open/listen/accept/bind` primitives to open a channel
  - ⋆ `send/receive` to communicate
  - ⋆ `close` to tear down the channel
  - ⋆ blocking/non-blocking send/receive
- Shared memory communication:
  - ⋆ allocation
  - ⋆ read/write
  - ⋆ free
  - ⋆ interruptible/non-interruptible
- User controlled synchronisation
- Reliability levels

We focus on Message passing, Synchronous communication, End-to-end data acknowledgement

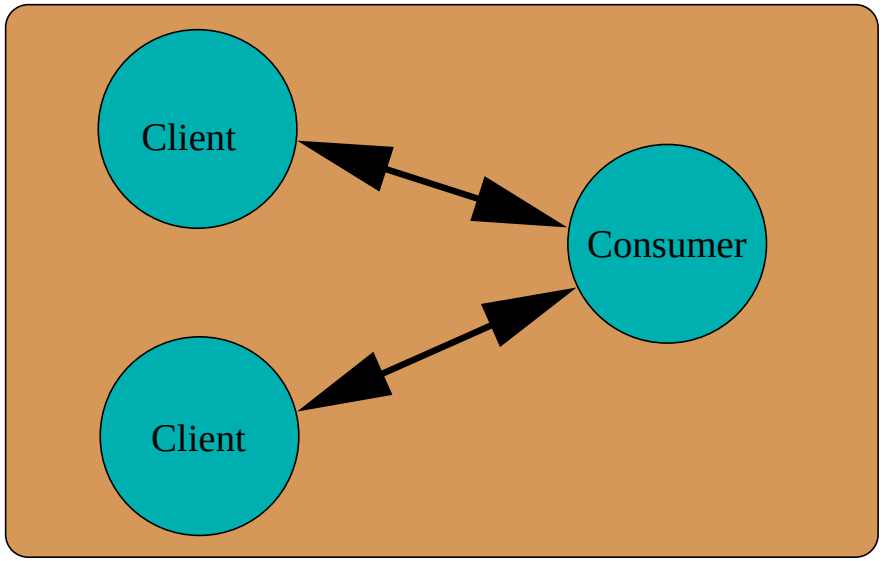
Communication Patterns



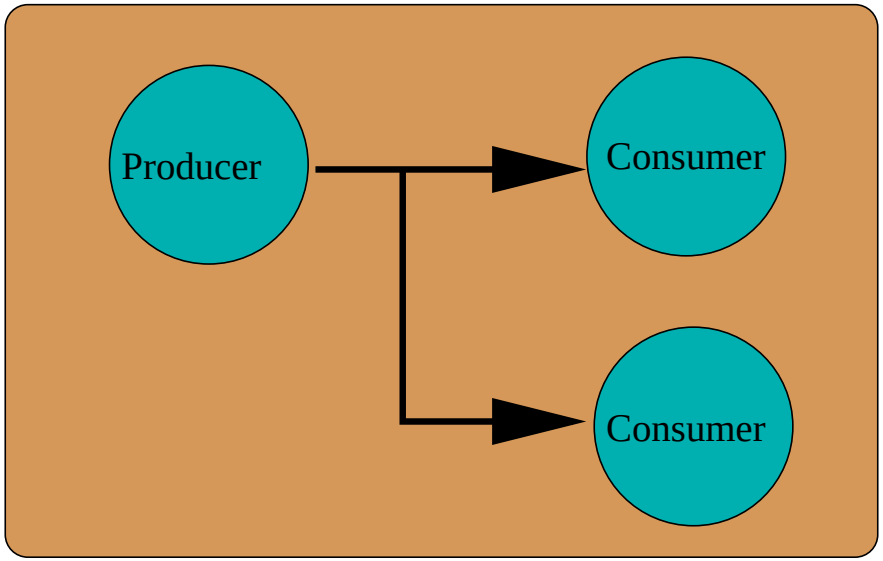
Producer−Consumer



Peer−to−Peer

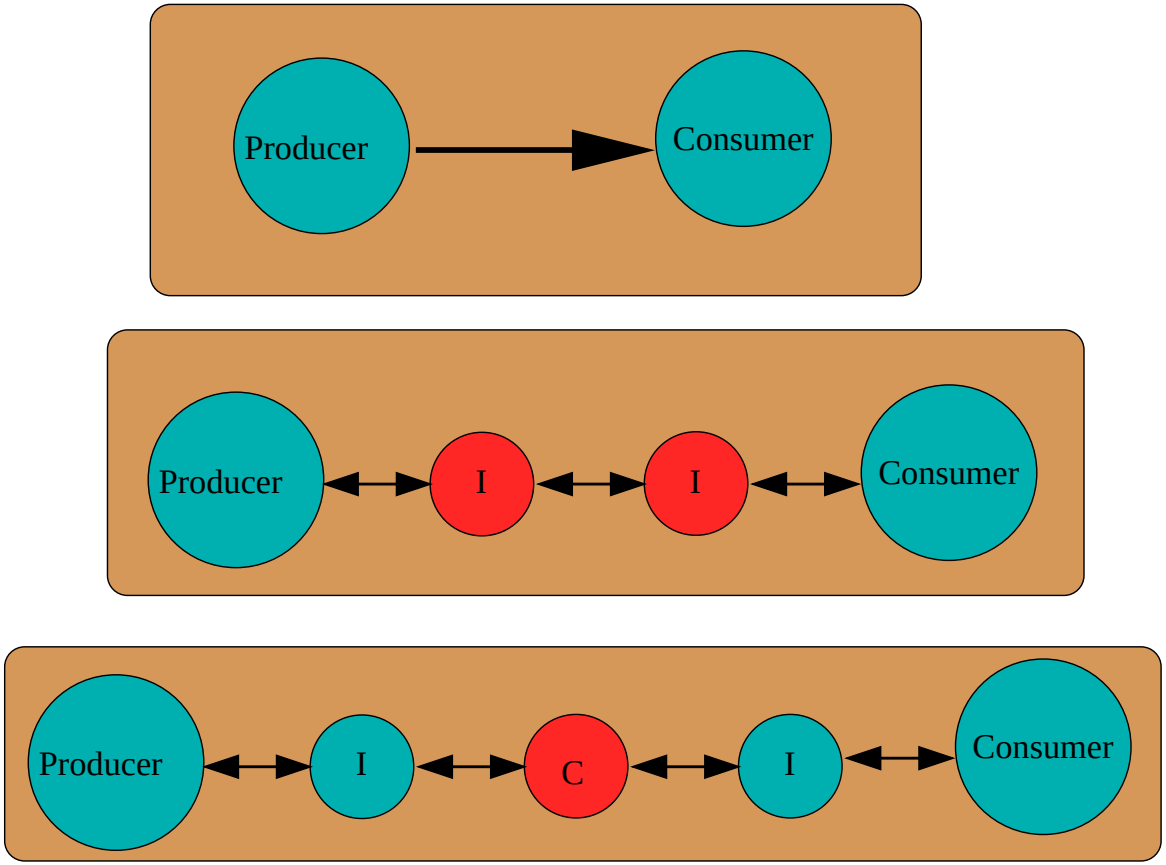


Client−Server



Multicast

Channel Refinement

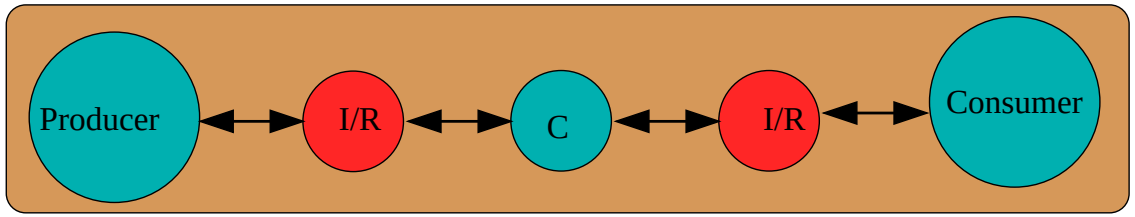


Synchronous Communication

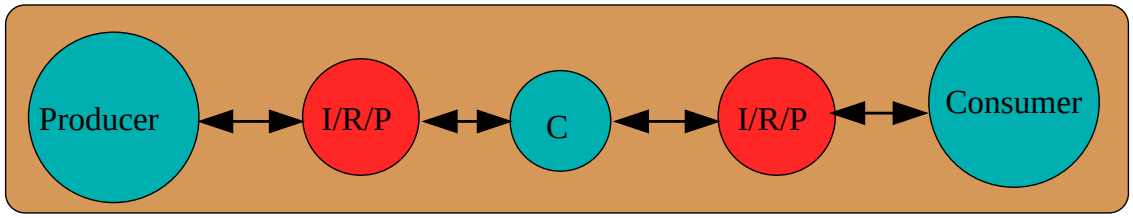
Asynchronous Interface

Channel performance model

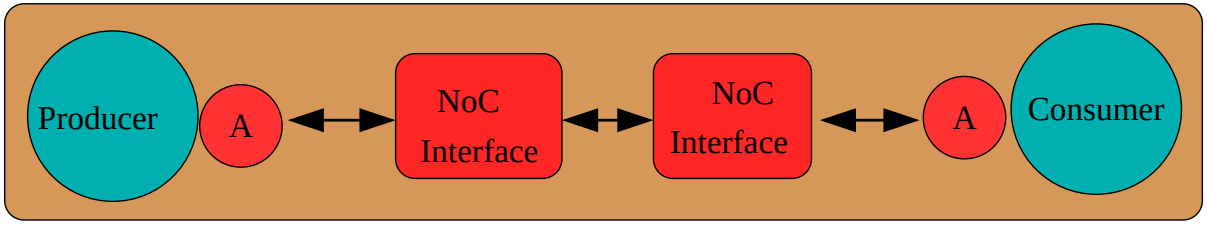
Channel Refinement - cont'd



Refinement for Reliability

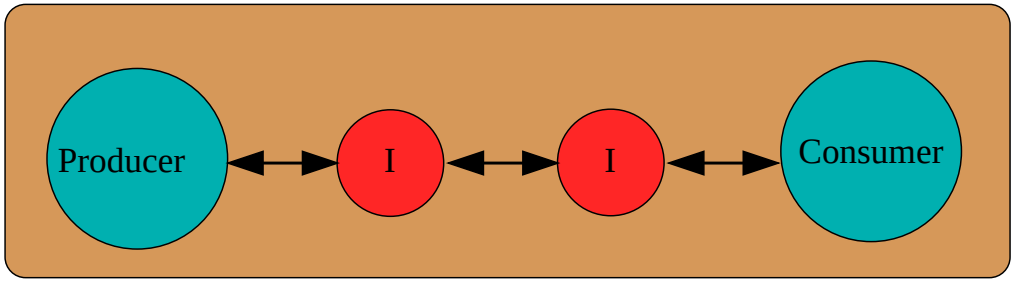


Refinement for Performance



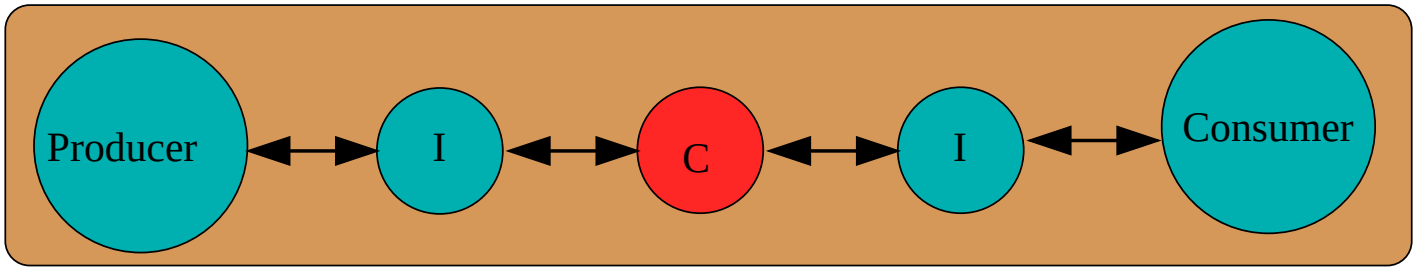
Mapping to NoC Services

Asynchronous Interfaces



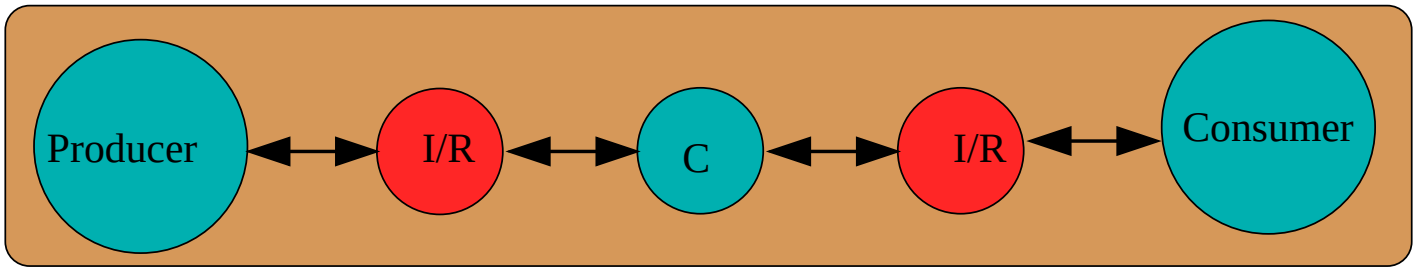
- Reflect the major steps of the targeted services
  - ⋆ Open; send/receive; close
  - ⋆ Assembling and disassembling of messages
  - ⋆ Buffering
- Introduction of flow control
  - ⋆ for opening a connection
  - ⋆ for sending/receiving messages

Modelling the Channel



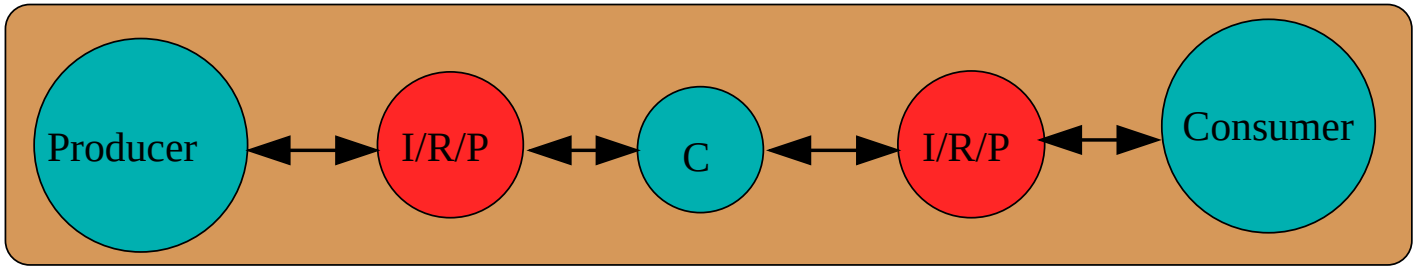
- Delay
- Jitter
- Reliability
- Deterministic or stochastic model

Refinement for Reliability



- Design for a fault model!
- Possible faults:
  - ⋆ Lost package
  - ⋆ Faulty data in arriving packet
  - ⋆ Spurious packet
  - ⋆ Faulty sender/receiver
  - ⋆ etc.
- Example measure: Acknowledgement for every n packets

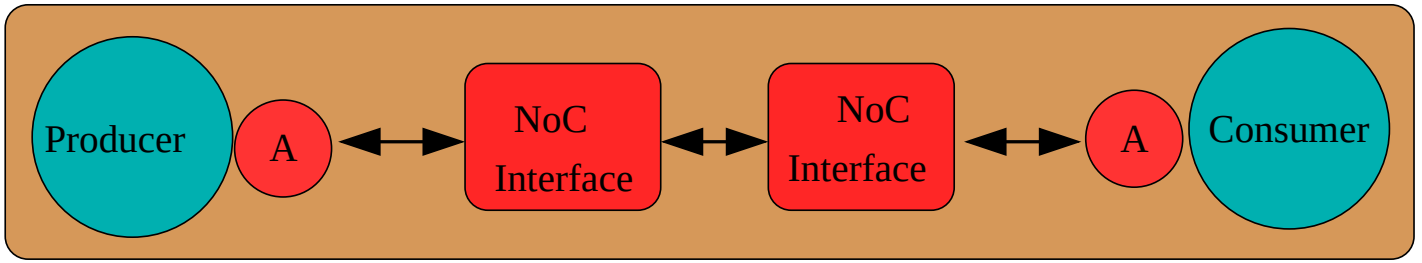
Refinement for Performance



Options for performance optimisations:

- Mapping to lower level services
- Buffer dimensioning for hiding jitter and delays due to flow control
- Overlapping acknowledgement with sending data
- etc.

Mapping onto NoC Services



- Selecting best effort/guaranteed bandwidth/guaranteed latency service
- Validating performance and reliability
- Merging several channels into a single virtual circuit
- Instantiating adapter to the NoC service
- Static/Dynamic allocation of virtual circuits

Summary

- Refinement of task-to-task communication to NoC services
- Functionality, performance, area, power consumption and reliability are first class design objectives
- Six step refinement procedure