

Communication Refinement for a Network-on-Chip Platform

Axel Jantsch and Zhonghai Lu
Royal Institute of Technology, Stockholm

April 2003

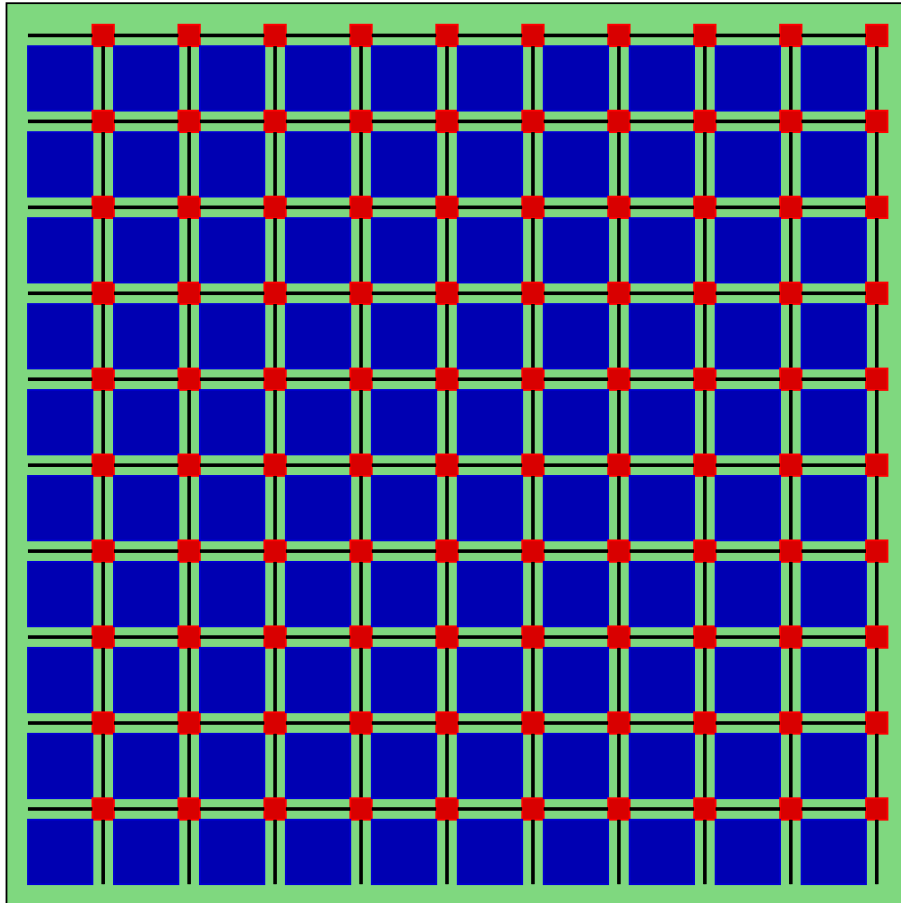


Overview

Nostrum Architecture and Platform
Communication Stack
Communication Patterns
Refinement
Summary



Nostrum Topology: Mesh



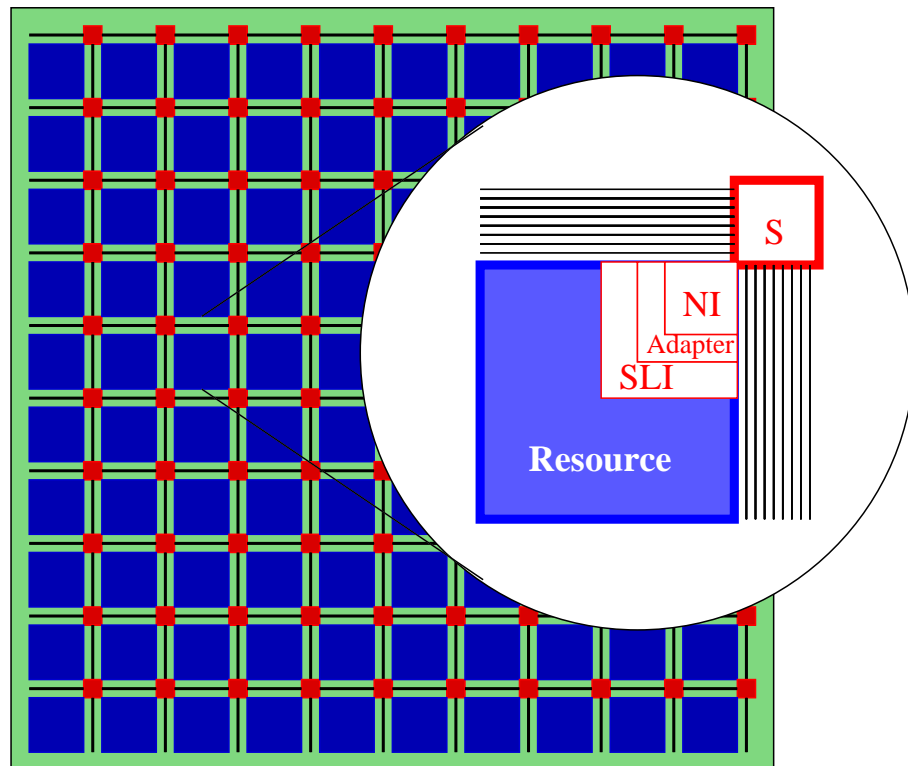
Characteristics:

- 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;

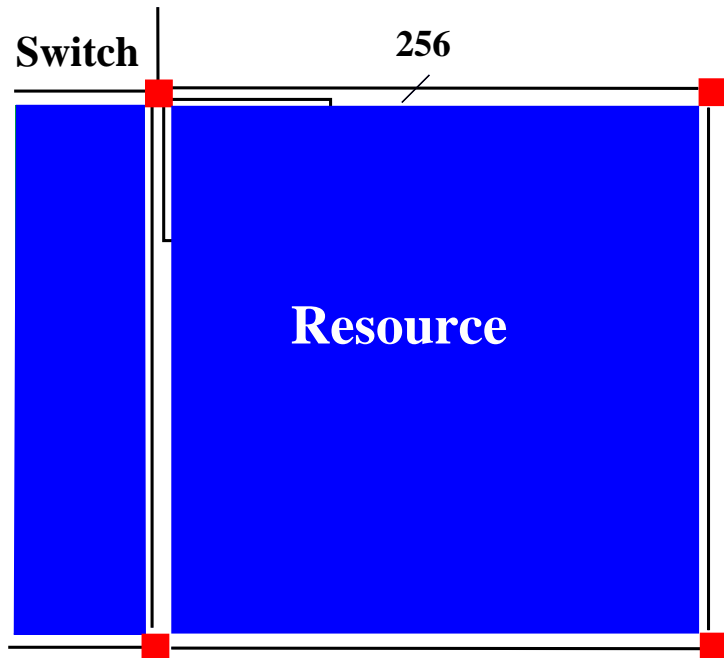
SLI: Session Layer Interface:

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

Node Geometry

Scenario:

- 60nm CMOS
- 22mm × 22mm chip size
- 300nm minimal wire pitch
- 2mm × 2mm resource
- 100μm × 100μm switch
- ⇒ 1333 wires on four metal layers
- switch-to-switch connection: 256 shielded and differential data signals;
- switch-to-resource connection: 256 data signals

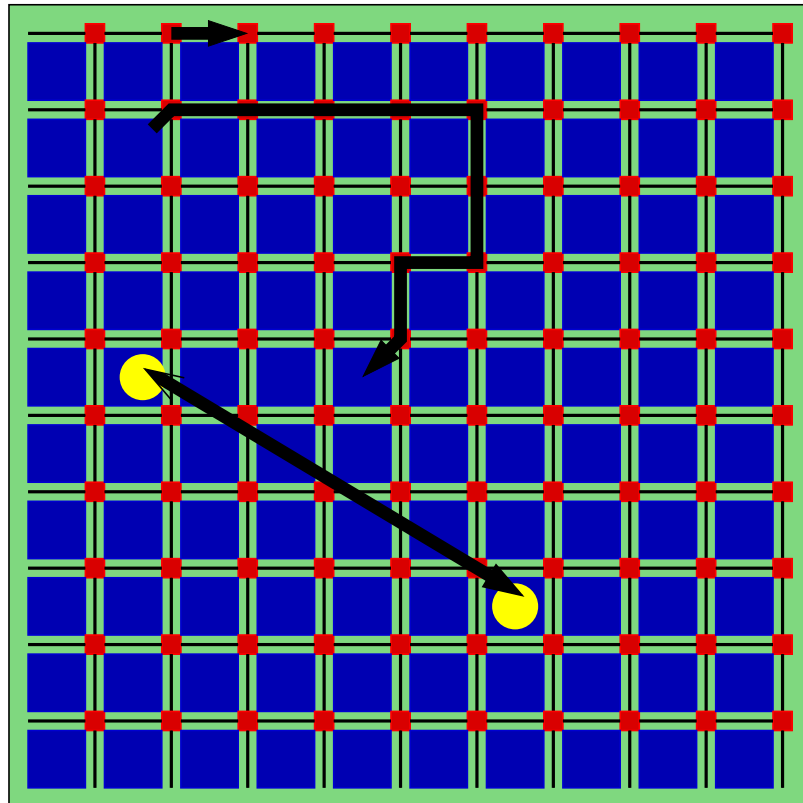


Nostrum Platform

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



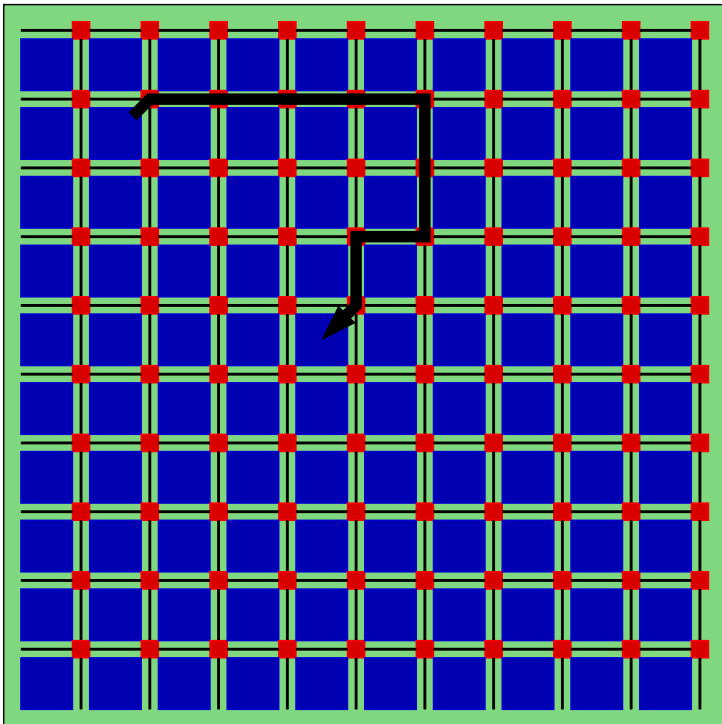
Nostrum Protocol Stack



Communication Layers:

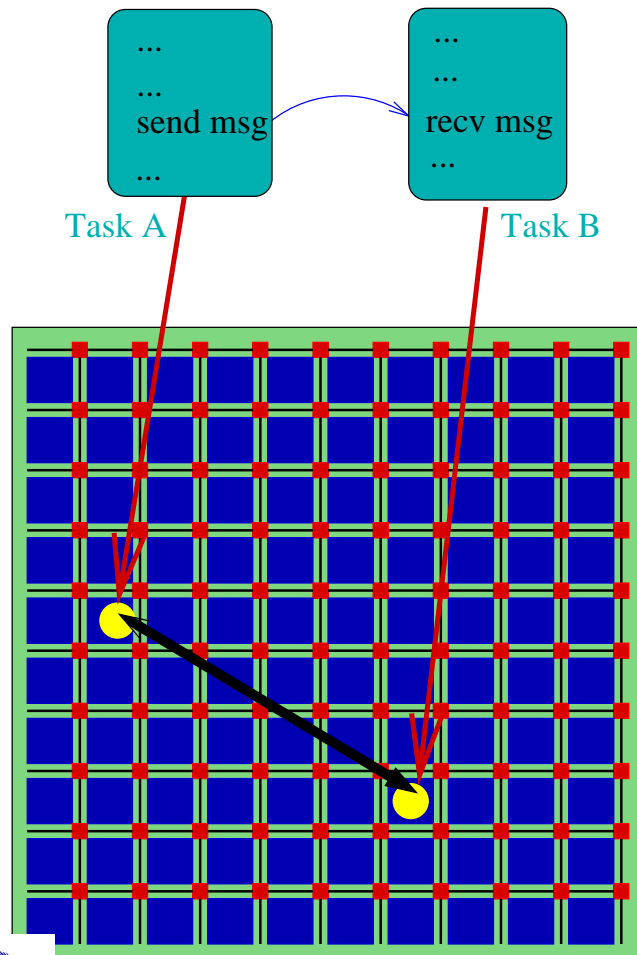
- Data link layer: switch-to-switch and switch-to-resource
- Network layer: resource-to-resource
- Session layer: process-to-process
- Application layer: application-to-application

Network Layer



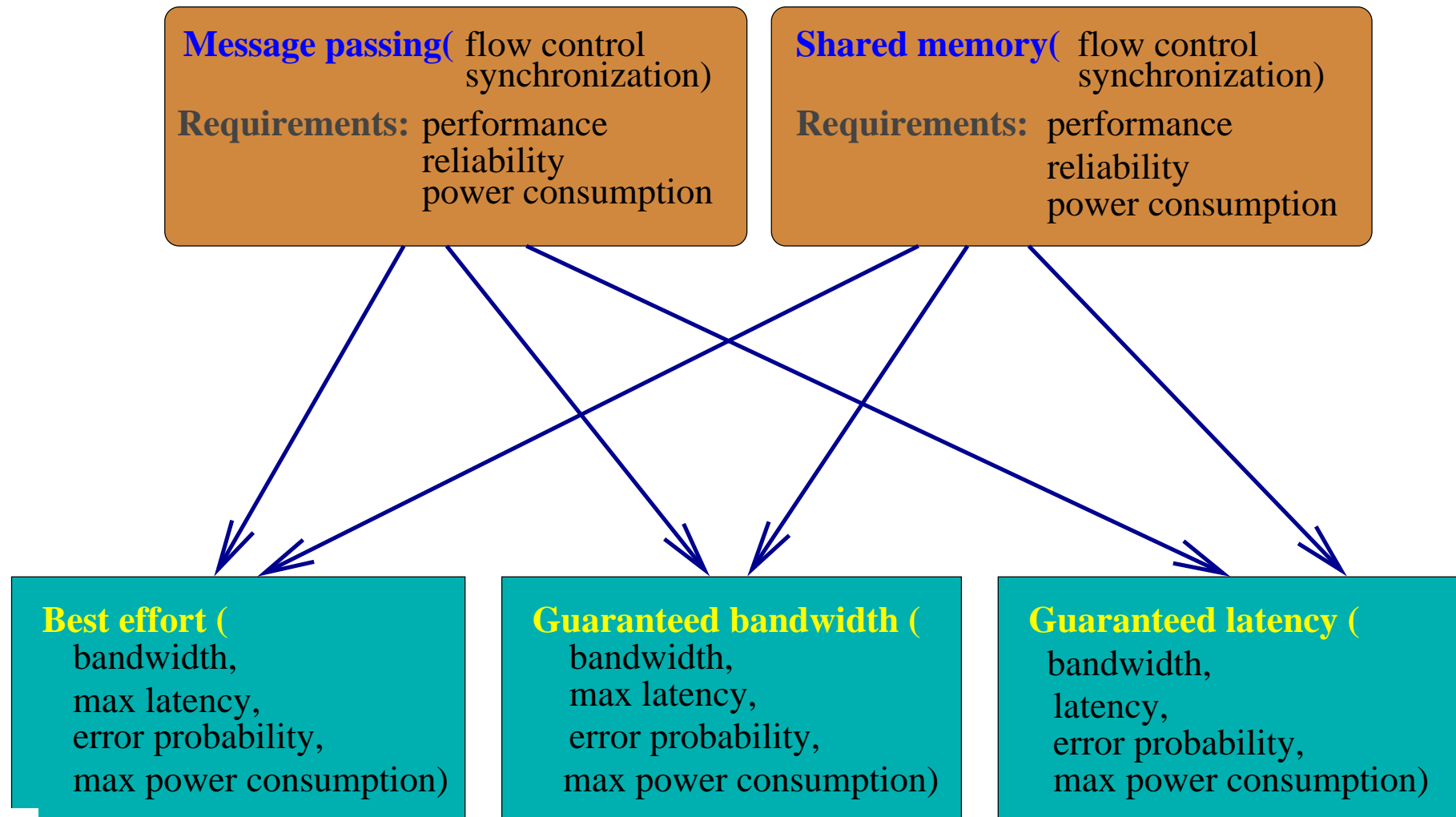
- Link layer frame = network layer packet
- Lossless communication
- Best effort service:
 - ★ Relative x-y addressing
 - ★ Out-of-order packet arrival
 - ★ Deflective routing with no buffers and no routing tables
- Virtual circuits with guaranteed bandwidth; varying latency
- Virtual circuits with guaranteed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

Session Layer

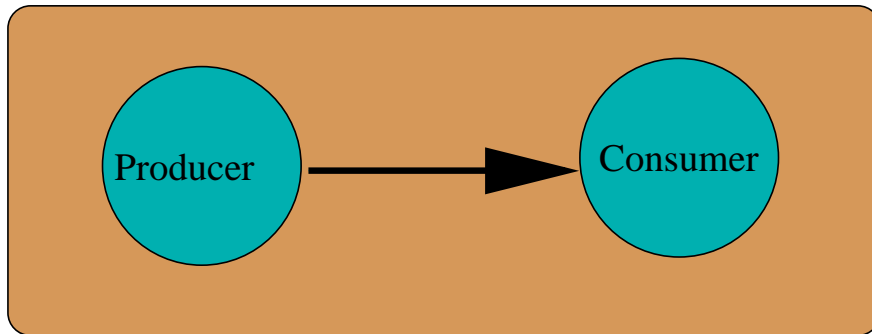


- Message passing communication:
 - ★ open/listen/accept/bind primitives to open a channel
 - ★ send/receive to communicate
 - ★ close to tear down the channel
 - ★ blocking/non-blocking send/receive
- Shared memory communication:
 - ★ allocation
 - ★ read/write
 - ★ read/modify/write
 - ★ free
- User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

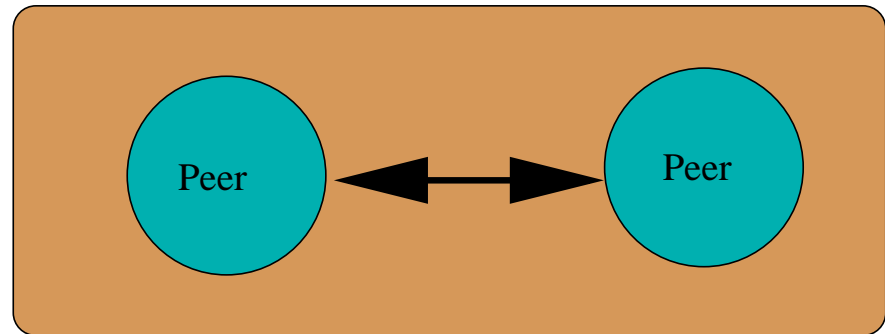
Refinement from Session Layer to Network Services



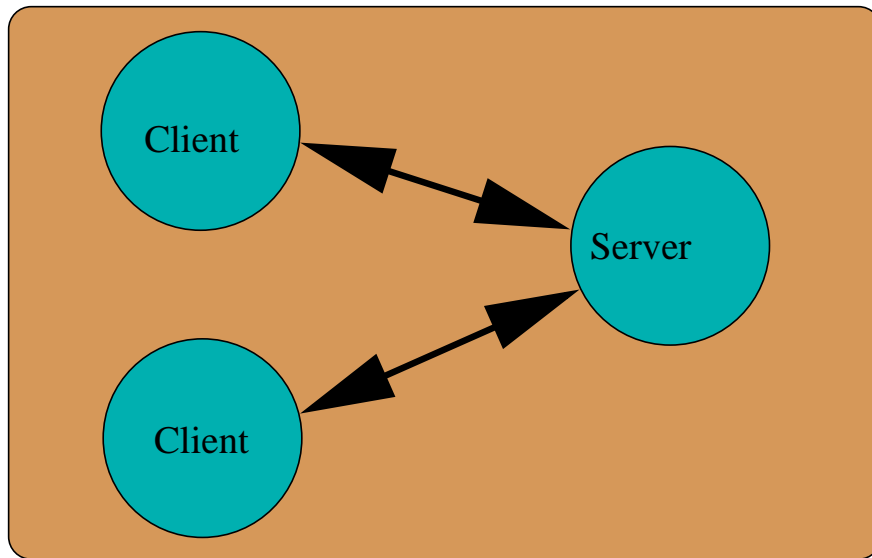
Communication Patterns



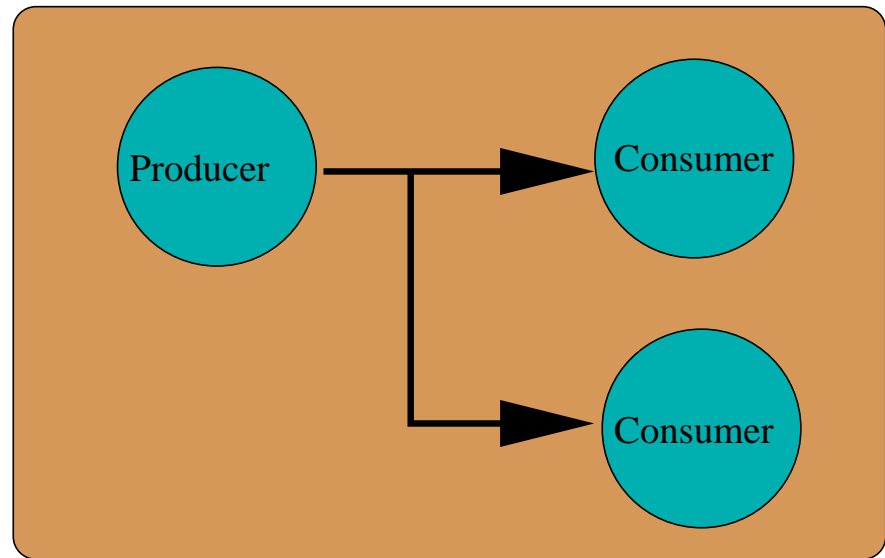
Producer-Consumer



Peer-to-Peer



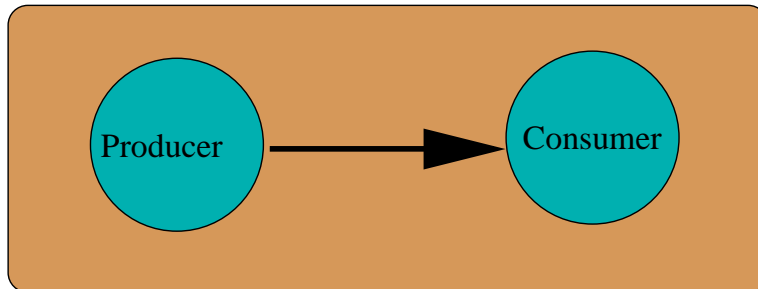
Client-Server



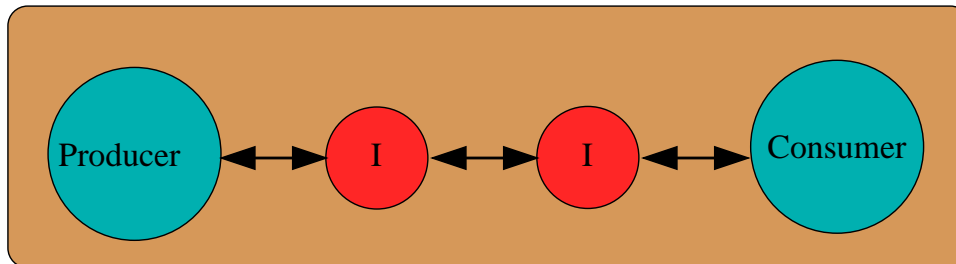
Multicast



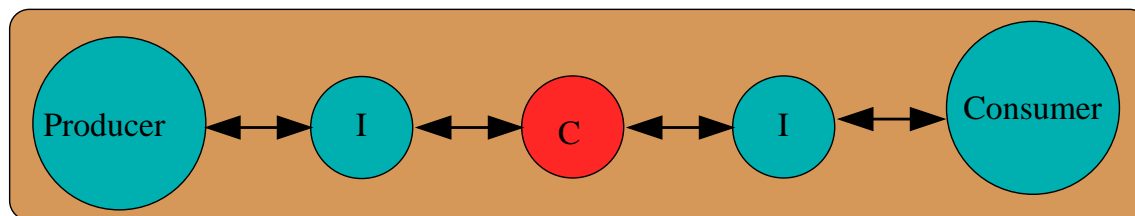
Channel Refinement



Ideal Communication

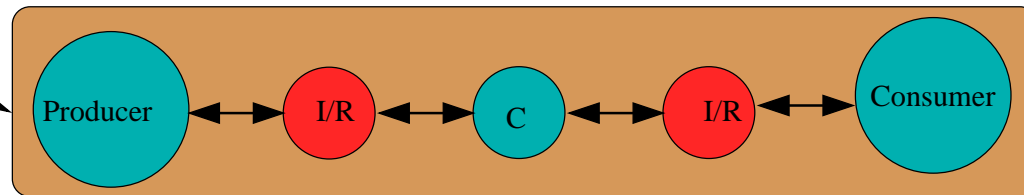


Asynchronous Interface
(Session layer communication)

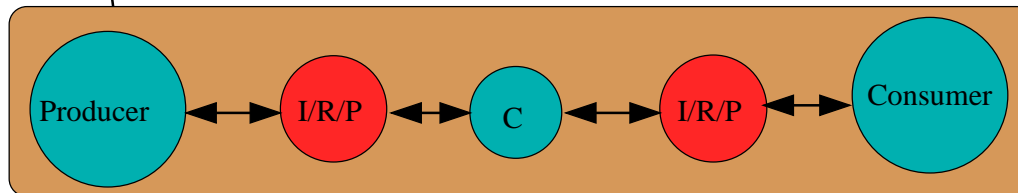


Channel performance model

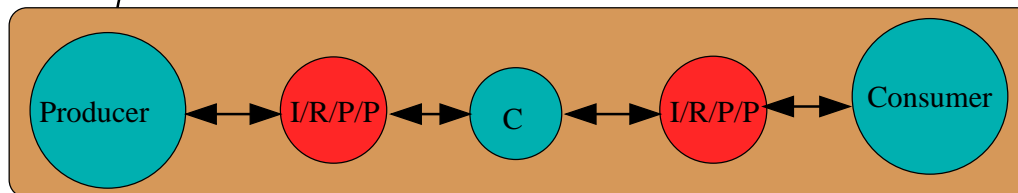
Channel Refinement - cont'd



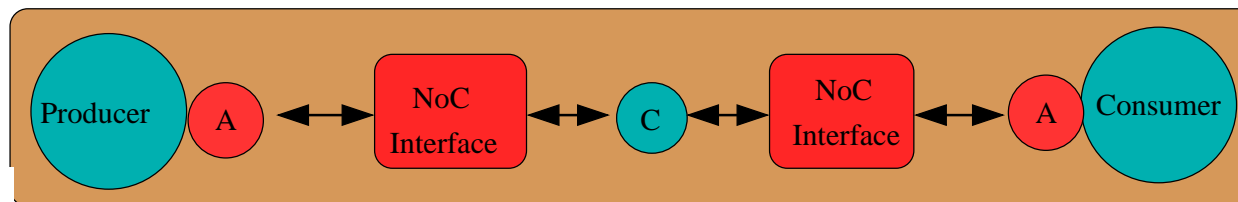
Refinement for Reliability



Refinement for Performance



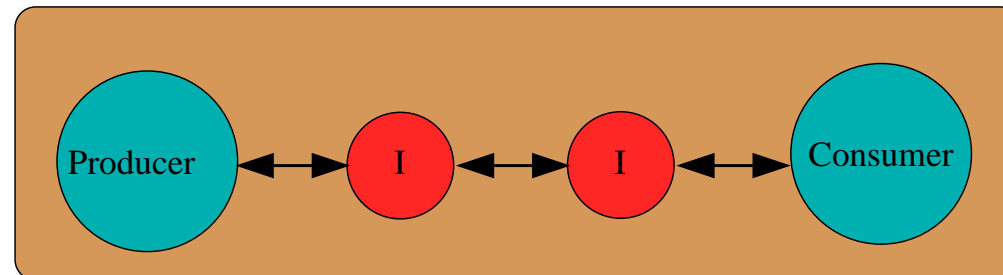
Refinement for Power



Mapping to NoC Services

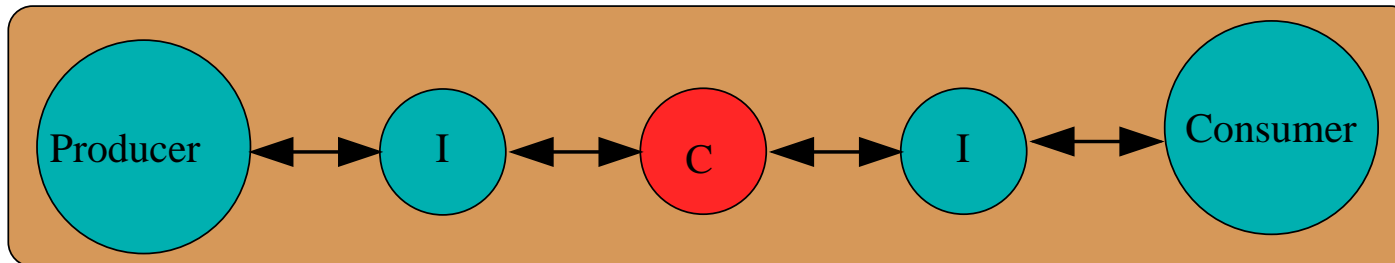


Asynchronous Interfaces



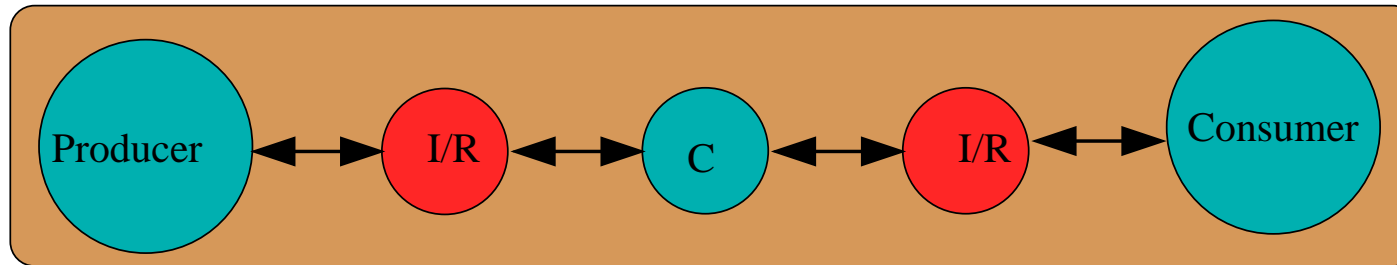
- Selection of session layer communication
 - ★ Open; send/receive; close
 - ★ Assembling and disassembling of messages
 - ★ Buffering
- Introduction of flow control
 - ★ for opening a connection
 - ★ for sending/receiving messages
- Specification of requirements on
 - ★ performance
 - ★ power
 - ★ reliability

Modelling the Channel



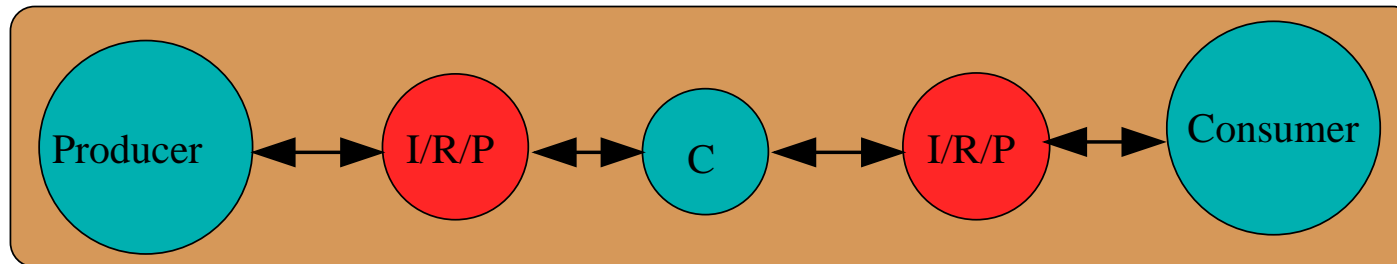
- Delay
- Jitter
- Reliability
- Deterministic or stochastic model

Refinement for Reliability



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

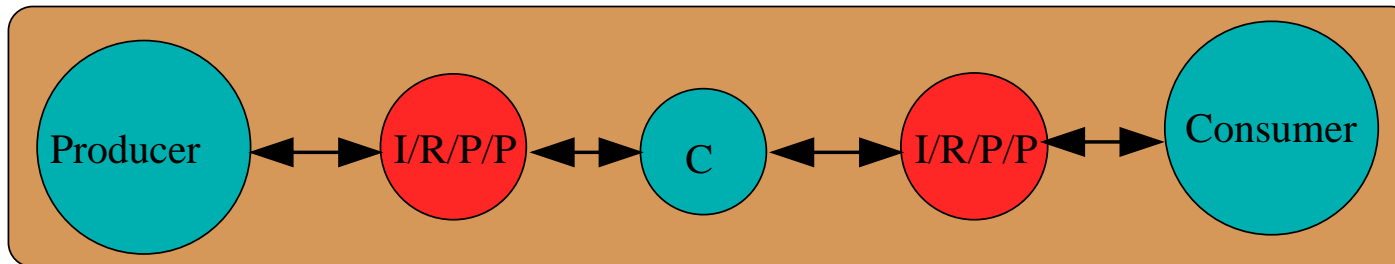
Refinement for Performance



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.

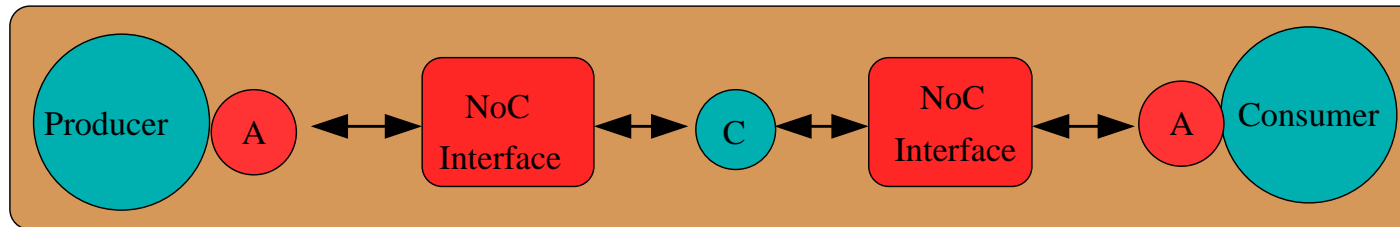
Refinement for Power



Options for power optimisations:

- Mapping to lower level services
- Minimizing traffic

Mapping onto NoC Services



- Selecting best effort/guaranteed bandwidth/guaranteed latency service
- Static/Dynamic allocation of virtual circuits
- Merging several channels into a single virtual circuit
- Validating performance and reliability
 - Static validation possible if
 - ★ Network services come with well defined characteristics
 - ★ Each communication channel has well specified requirements
- Instantiating adapter to the NoC service

Summary

- Refinement of task-to-task communication to NoC services
- Functionality, performance, power consumption and reliability are first class design objectives
- Static validation possible if
 - ★ Network services come with well defined characteristics
 - ★ Each communication channel has well specified requirements
- Communication design is composable
- Seven step refinement procedure

