

PreMaDoNA

Predictable Matching of Demands on Networked Architectures

Resource management and Quality of Service

J. Van Meerbergen, P. Poplavko, M. Pastrnak

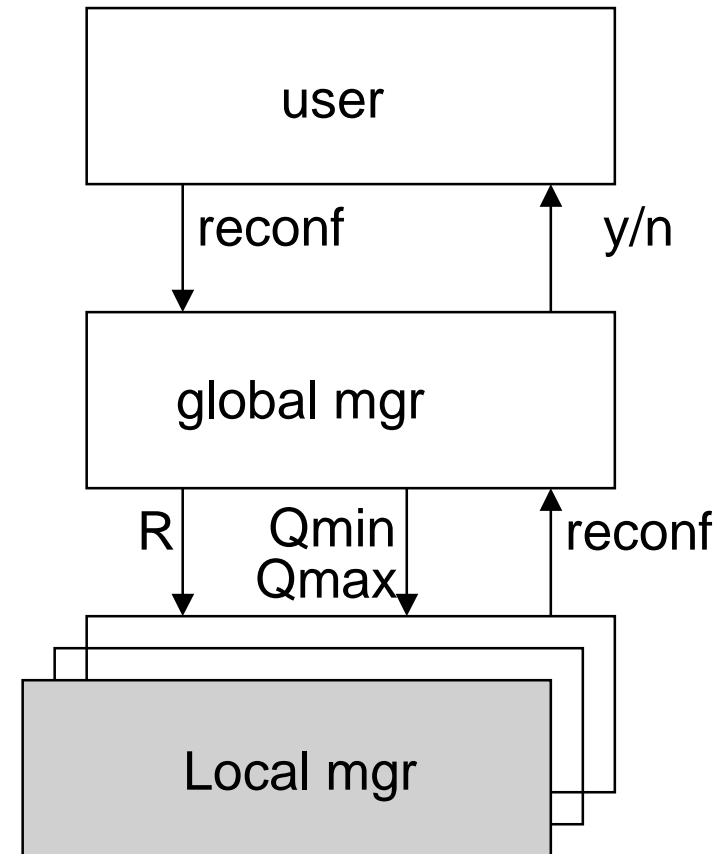
2 Domain analysis: summary

- Level 1: Task
 - Compute intensive kernels
- Level 2: Application/job
 - an activity (set of tasks) started/stopped by the user
- Level 3: System
 - Multiple jobs active simultaneously
 - Too many combinations to analyze at design time

- 2 step approach
 - Global level = application
 - Definition of coarse level operating points
 - local level = job
 - Fine level control

3 architecture

- Defines Resource budgets
- Estimate Quality levels
- Maximize the quality within a given Resource budget
- Generate reconfiguration request



4 Local manager

F_process_frame (in, Q)

...

(for I = 0..Q) ...

...

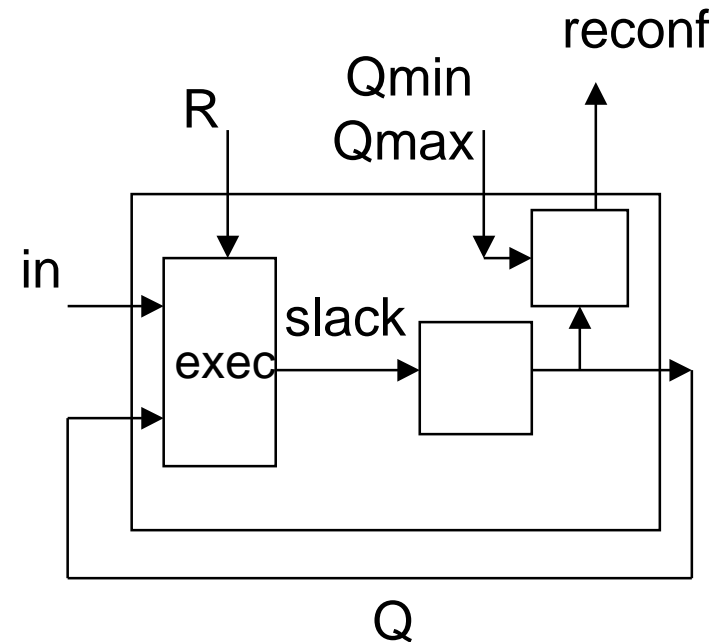
slack =

size = ...

local_mgr(par: Qmin, Qmax, R

in: slack, size

out: Q, reconf)



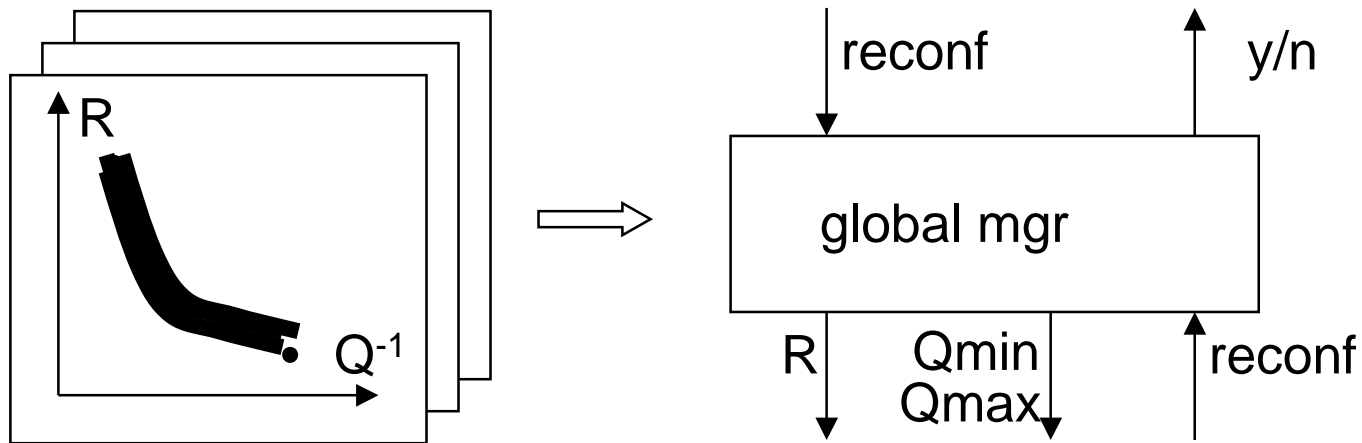
Q, slack, size ... are
just examples

5 Global manager

Given - a set of mappings for a new job to be started

- a number of jobs that are currently active

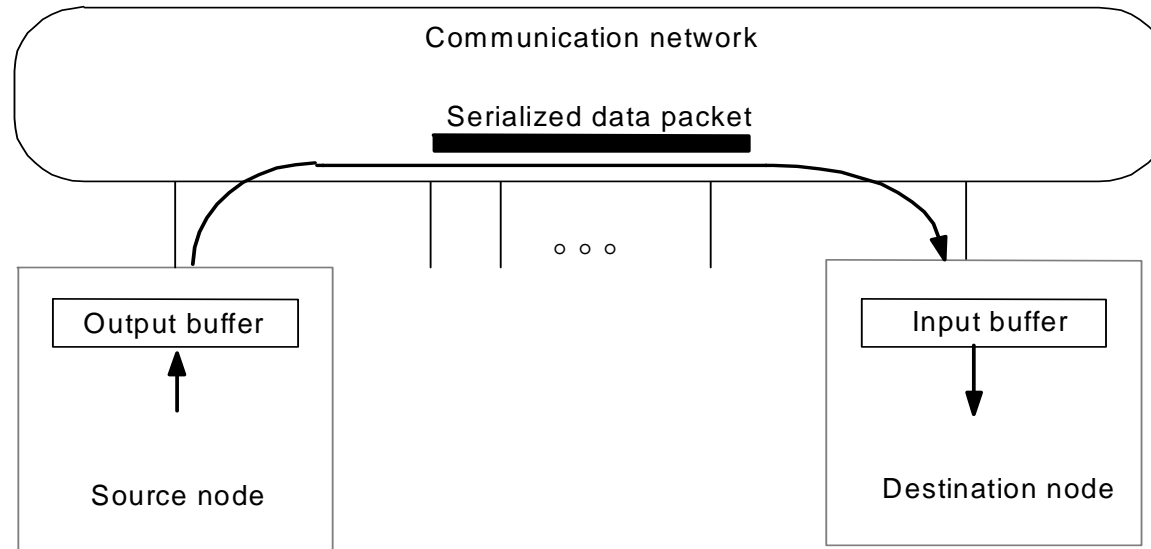
Find operation points for all jobs such that the quality is optimized



- Compile time
- Per job
- R is a vector

- Run time

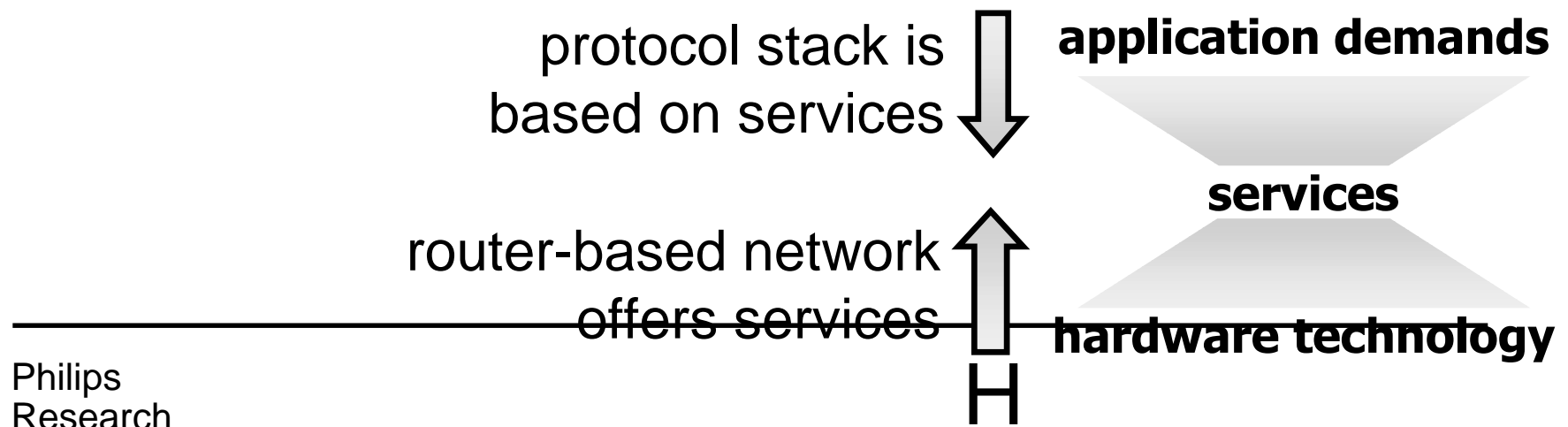
Communication: different from busses



- one-way transfer of information from a source output buffer to a dest. input buffer
 - causes some action at the destination
 - occurrence is not directly visible at source

8 networks on chip: protocol stack

- application
 - presentation
 - session
 - transport
-
- network
 - link
 - physical
- application diversity
 - network independent
 - peer to peer
- network on chip
 - network dependent

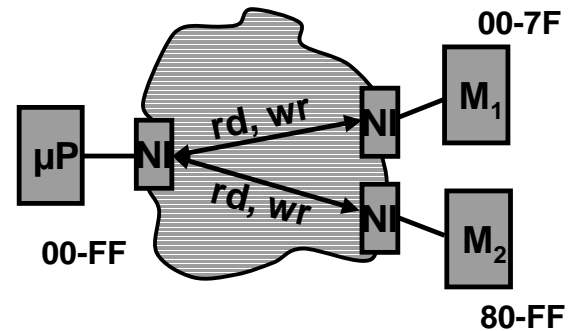
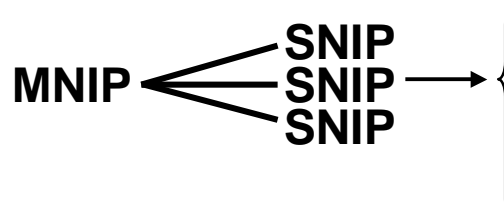


9 connection types

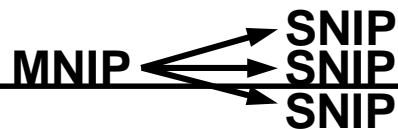
- simple
 - bidirectional

MNIP — SNIP

- narrowcast
 - bidirectional
 - for distributed shared memory

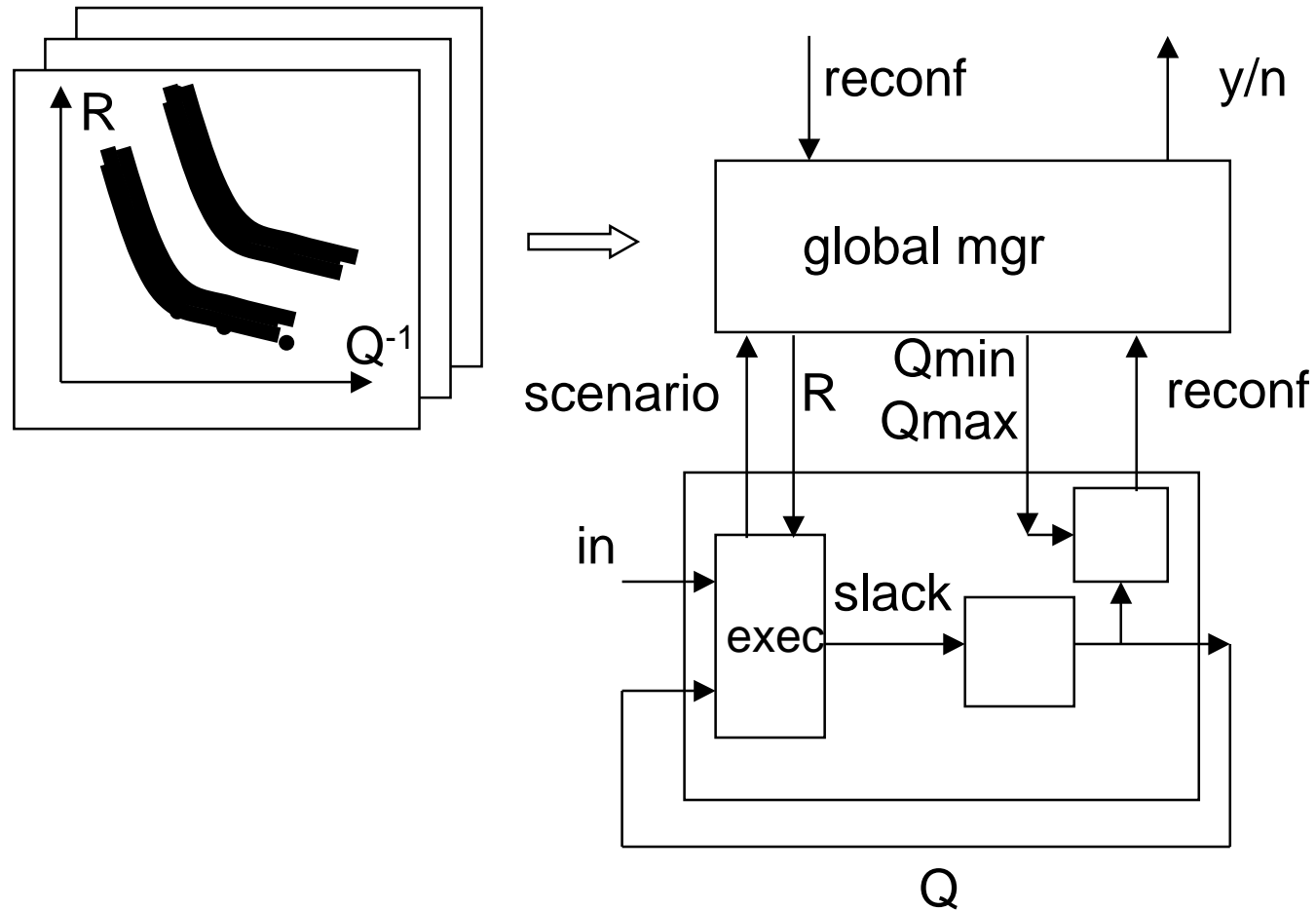


- multicast
 - unidirectional



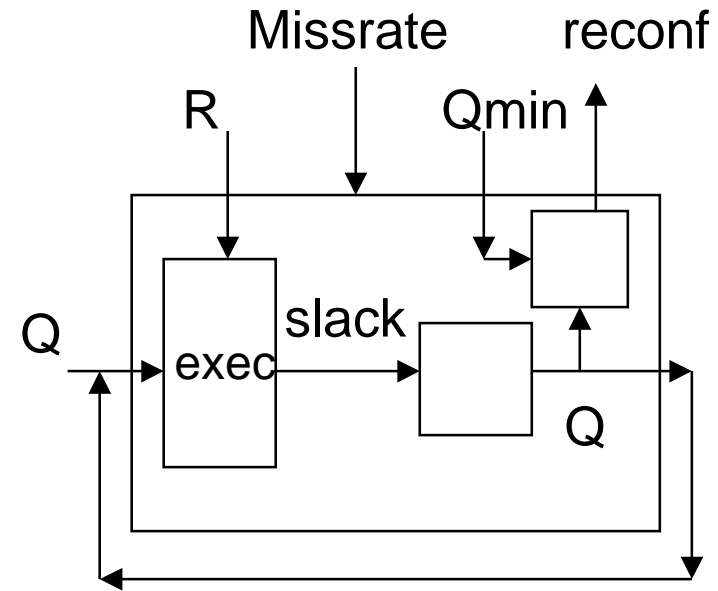
NI: network interface
 NIP: NI port
 MNIP: master NIP
 SNIP: slave NIP

10 Scenarios: input defined modes

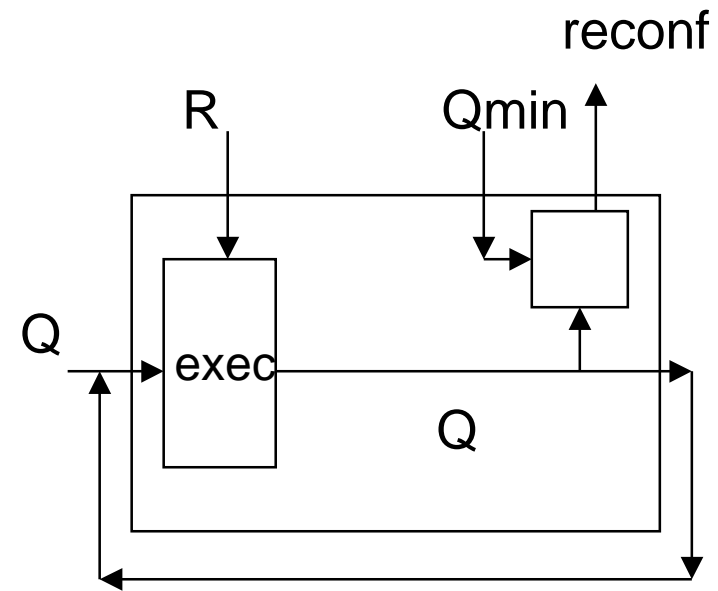


Scenario change is sent to the GM

- Case 1
 - Max Q for given R + missrate
 - Measure slack + adapt Q
 - Generate reconf req
 - If $Q < Q_{min}$



- Case 2: wavelet



- Case 3

- Obj function e.g. $Q+Mr$
- Max obj for given R + missrate
 - Compare with previous obj
- Generate reconf req
 - If $Q < Q_{min}$
 - If obj ??
- Char job
 - Mr
 - Q
 - OBJ

