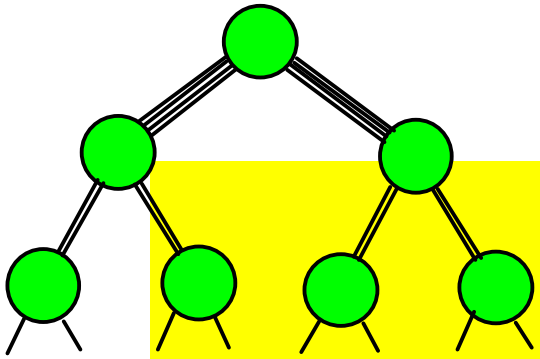
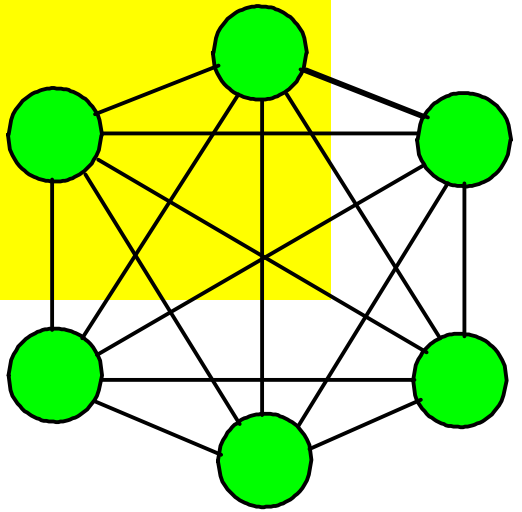
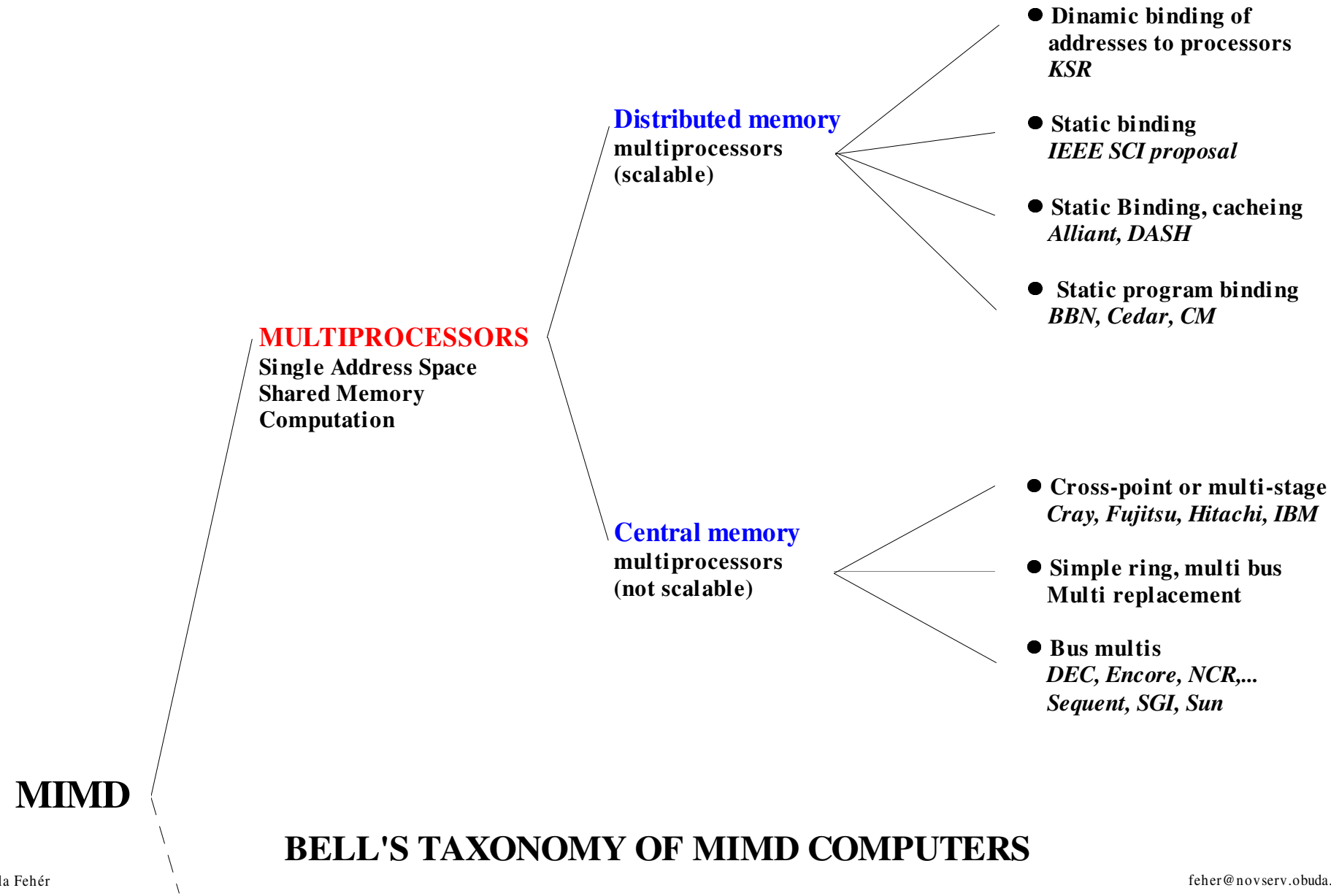


MULTICOMPUTER
Architectures

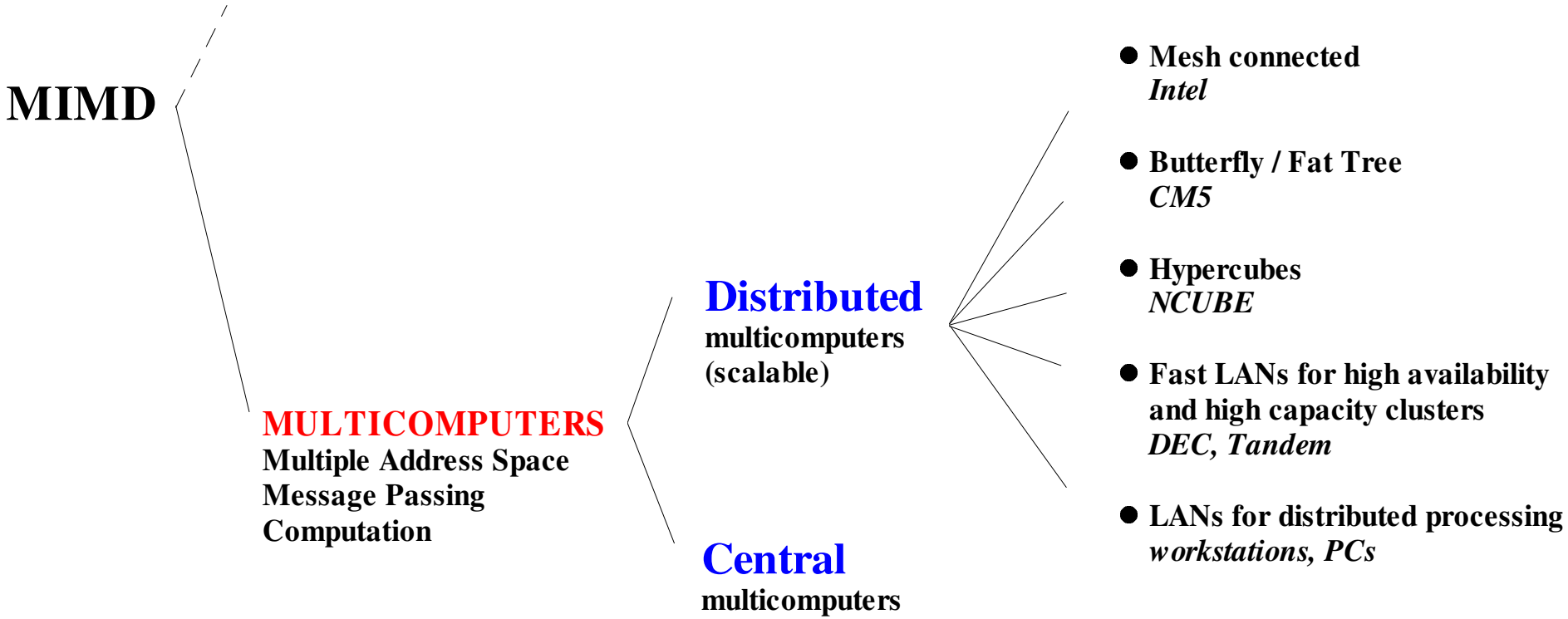


Gyula Fehér feher@novserv.obuda.kando.hu



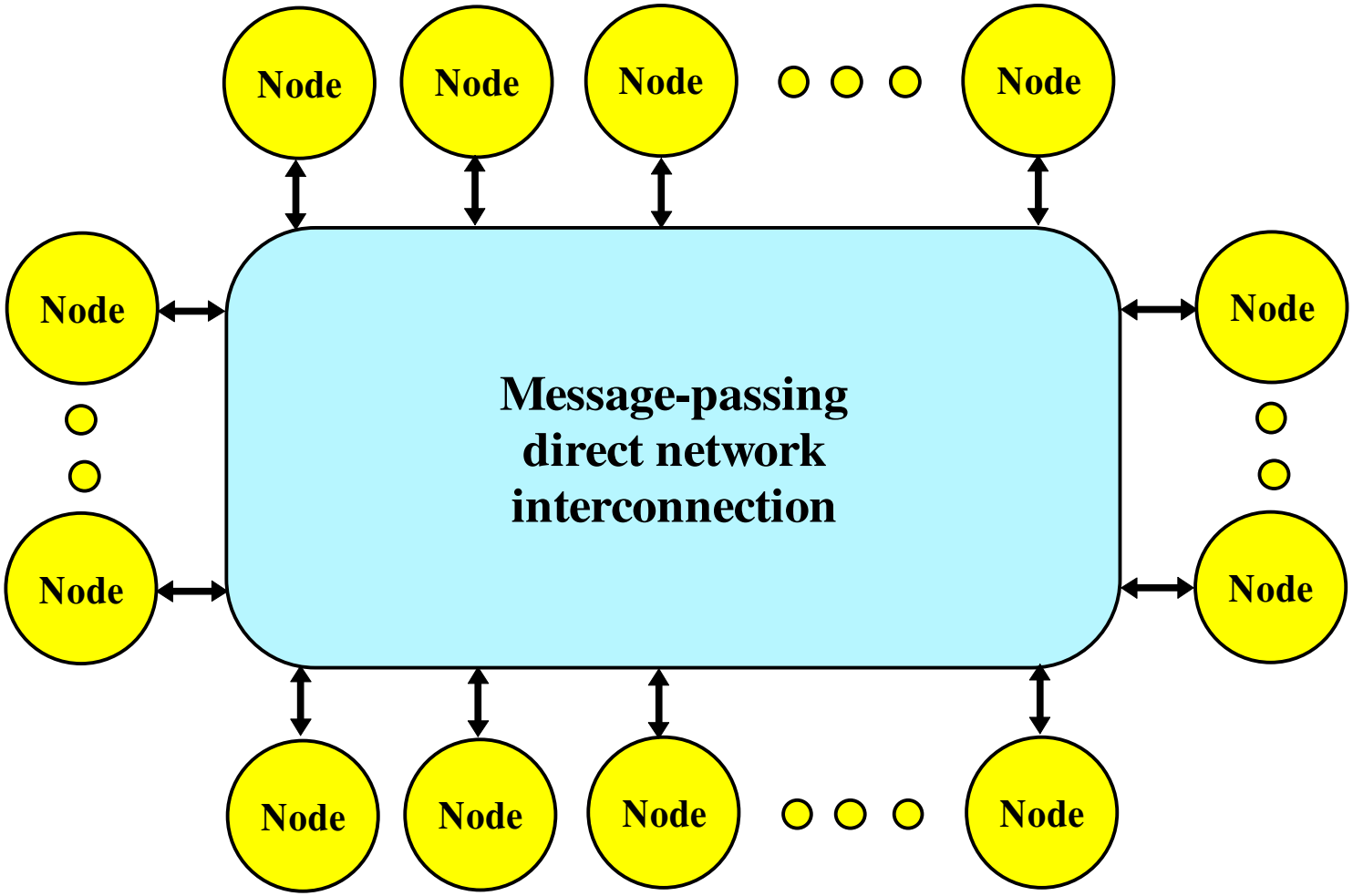
BELL'S TAXONOMY OF MIMD COMPUTERS

Gyula Fehér feher@novserv.obuda.kando.hu



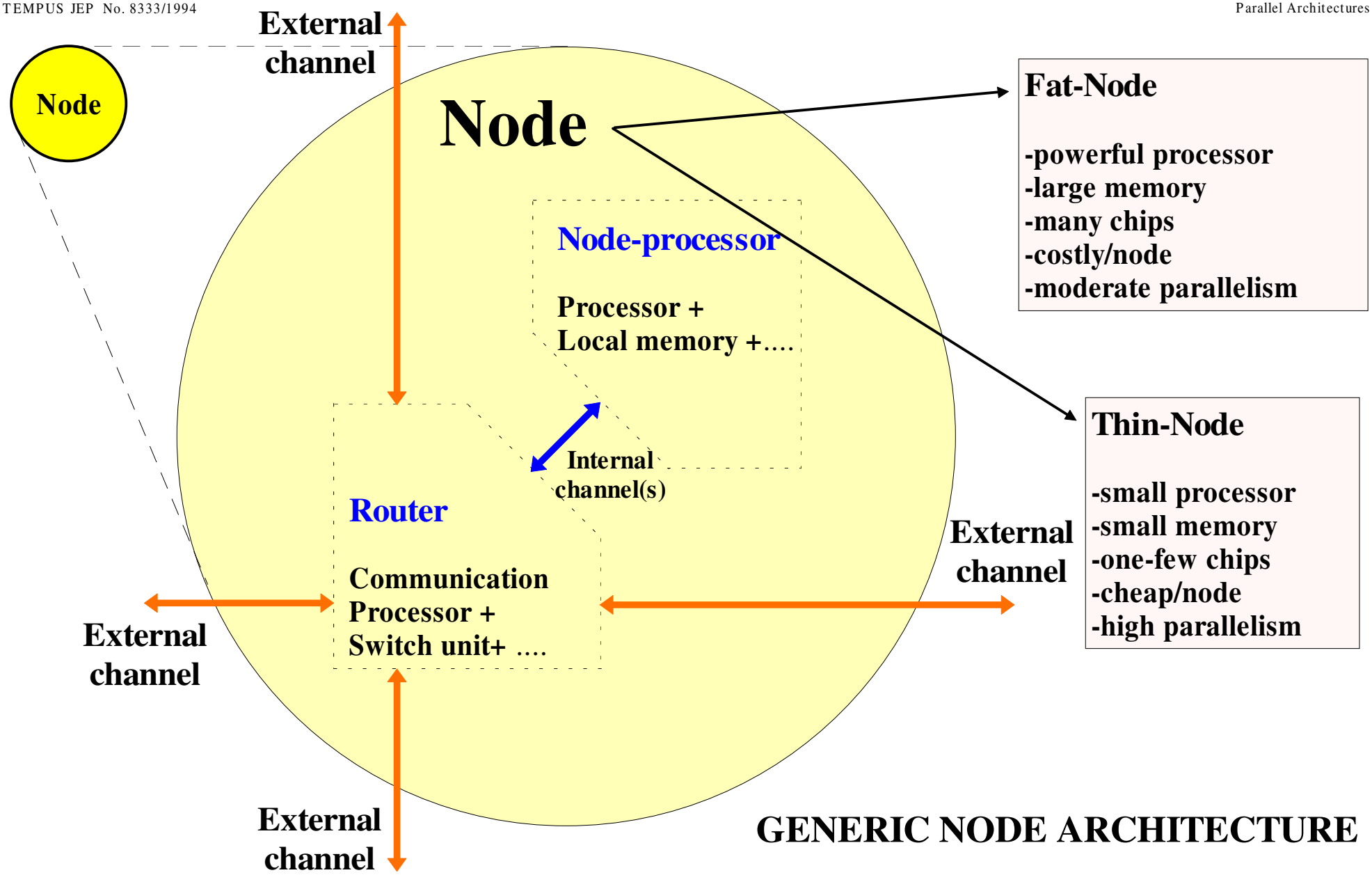
BELL'S TAXONOMY OF MIMD COMPUTERS

Gyula Fehér feher@novserv.obuda.kando.hu



GENERIC MODEL OF A MESSAGE-PASSING MULTICOMPUTER

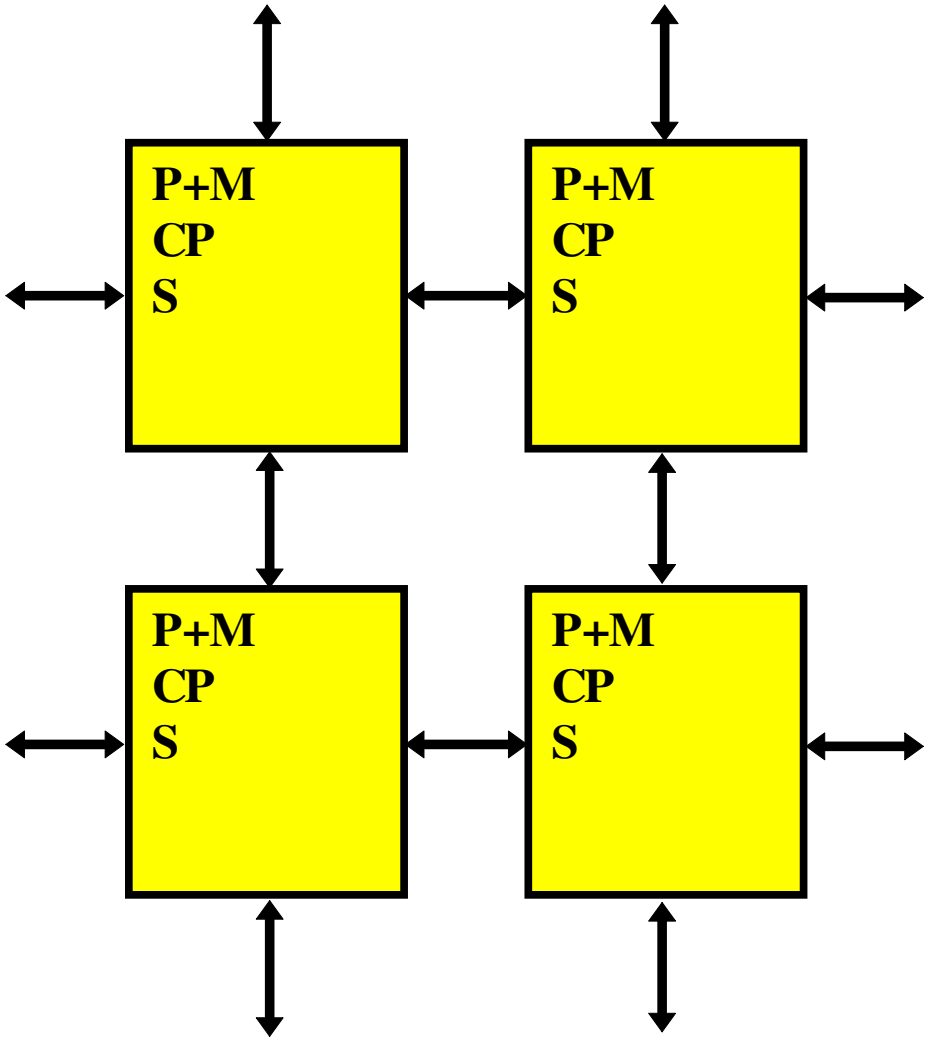
Gyula Fehér feher@novserv.obuda.kando.hu



GENERIC NODE ARCHITECTURE

Gyula Fehér

feher@novserv.obuda.kando.hu



Legends:

P = Processor

M = Memory

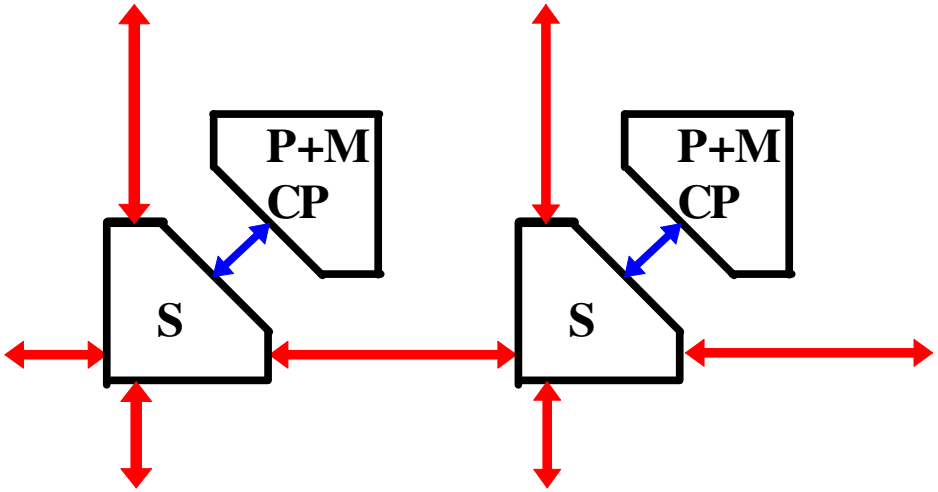
CP = Communication processor

S = Switch unit

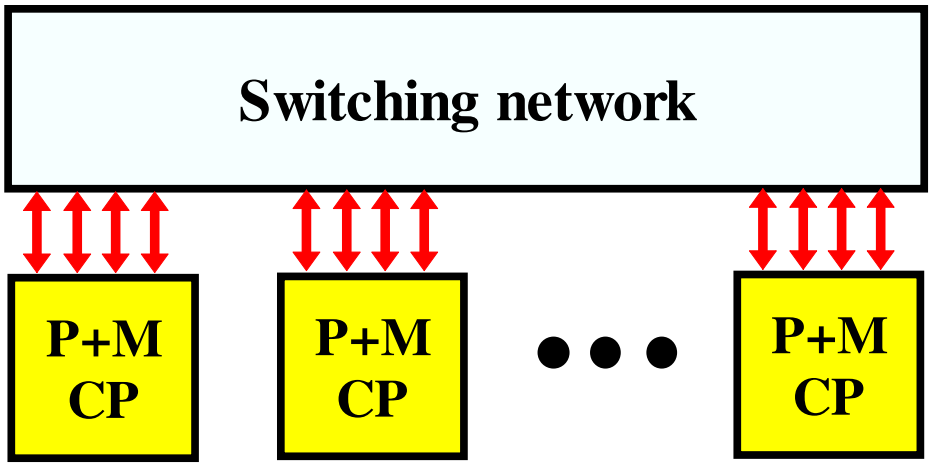
(a) First generation

GENERIC ORGANIZATION MODEL OF THE MESSAGE-PASSING MULTICOMPUTERS

Gyula Fehér feher@novserv.obuda.kando.hu



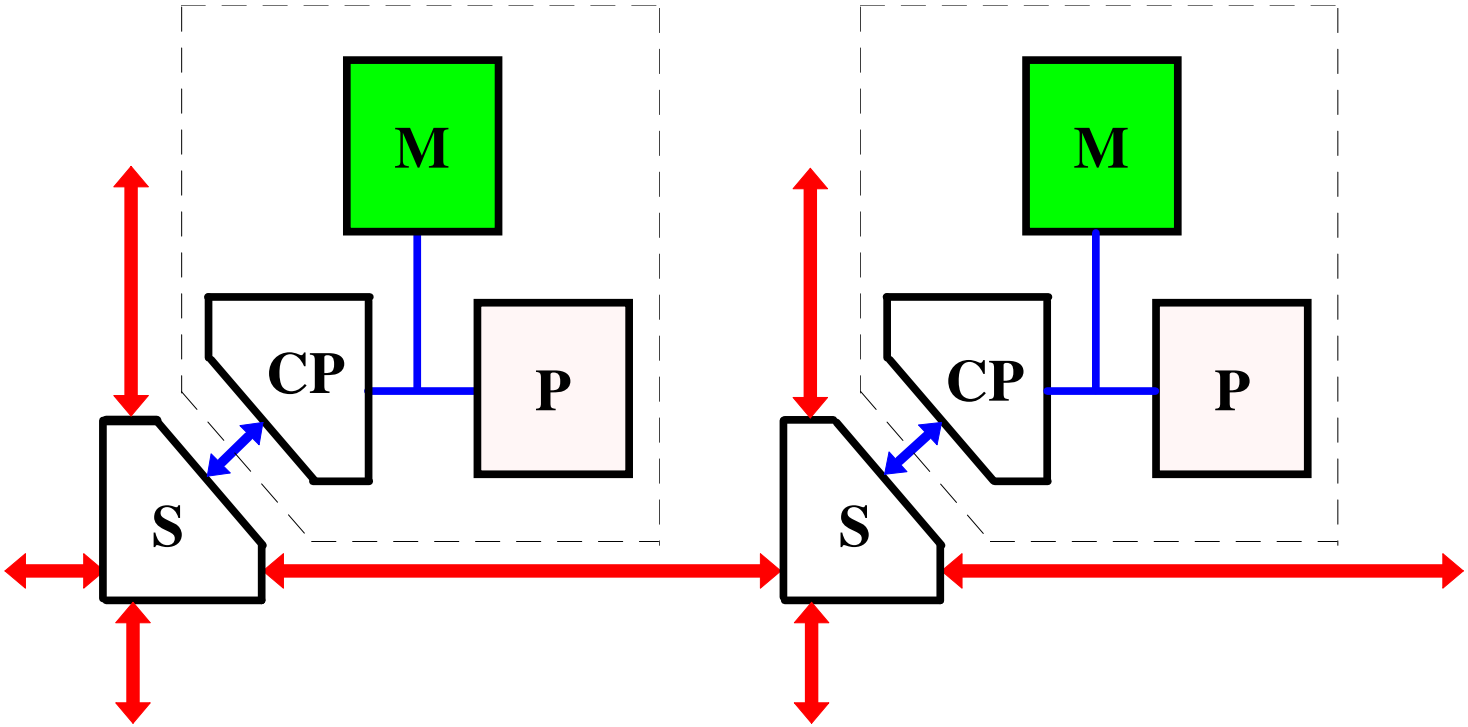
(b) Decentralized 2nd generation



(c) Centralized 2nd generation

GENERIC ORGANIZATION MODEL OF MESSAGE-PASSING MULTICOMPUTERS

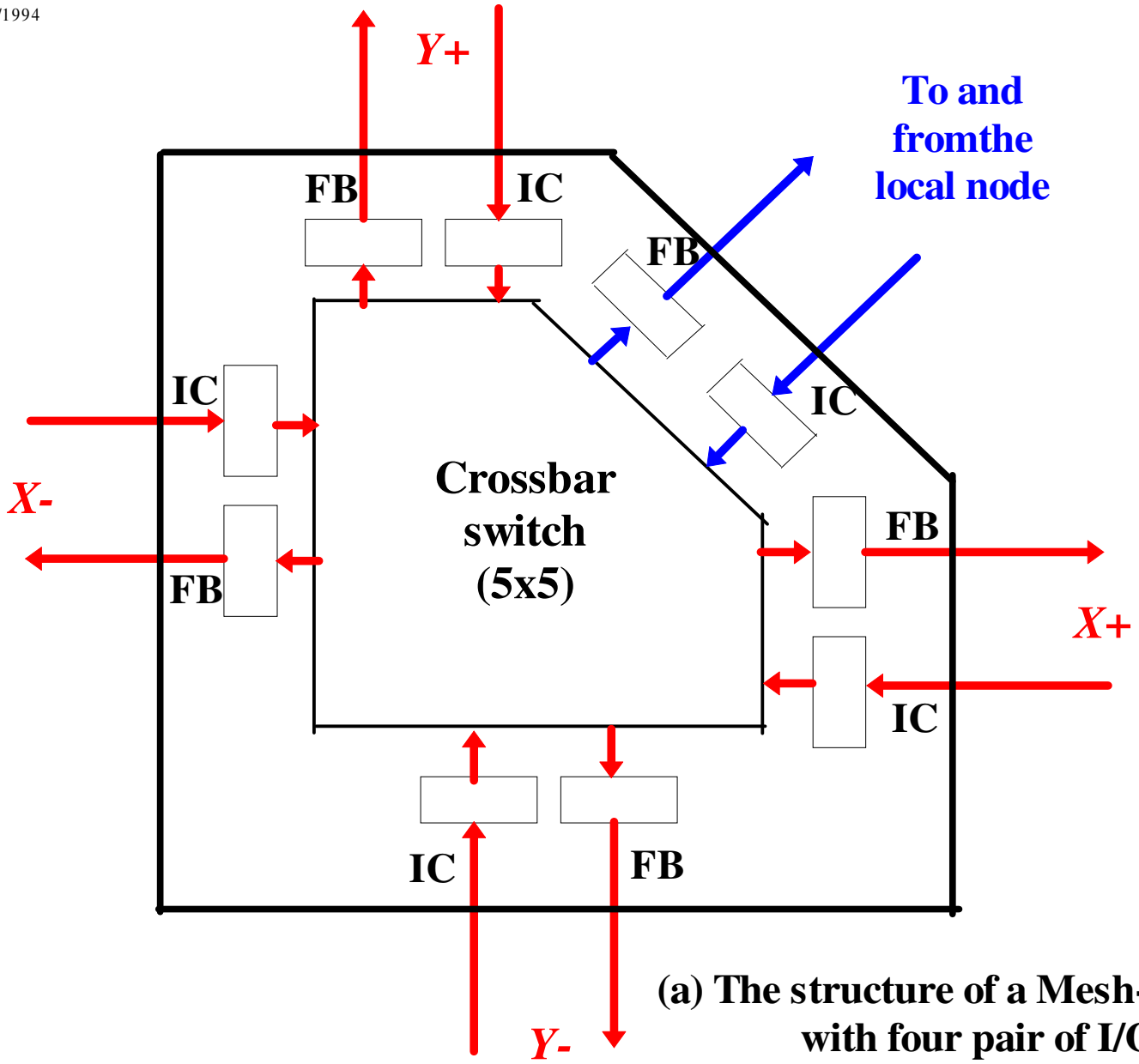
Gyula Fehér feher@novserv.obuda.kando.hu



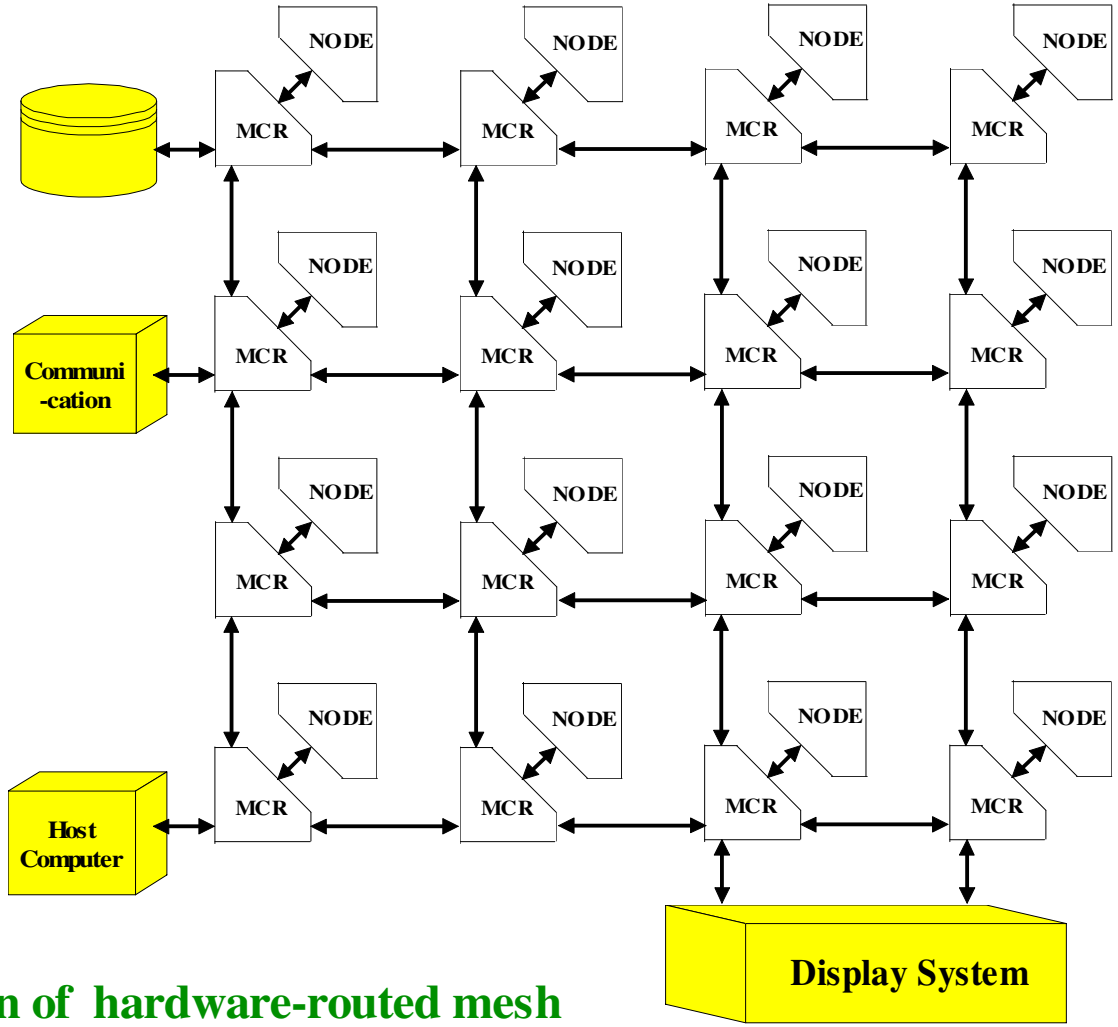
(d) 3rd generation

GENERIC ORGANIZATION MODEL OF THE MESSAGE-PASSING MULTICOMPUTERS

Gyula Fehér feher@novserv.obuda.kando.hu



(a) The structure of a Mesh-Connected Router with four pair of I/O channels

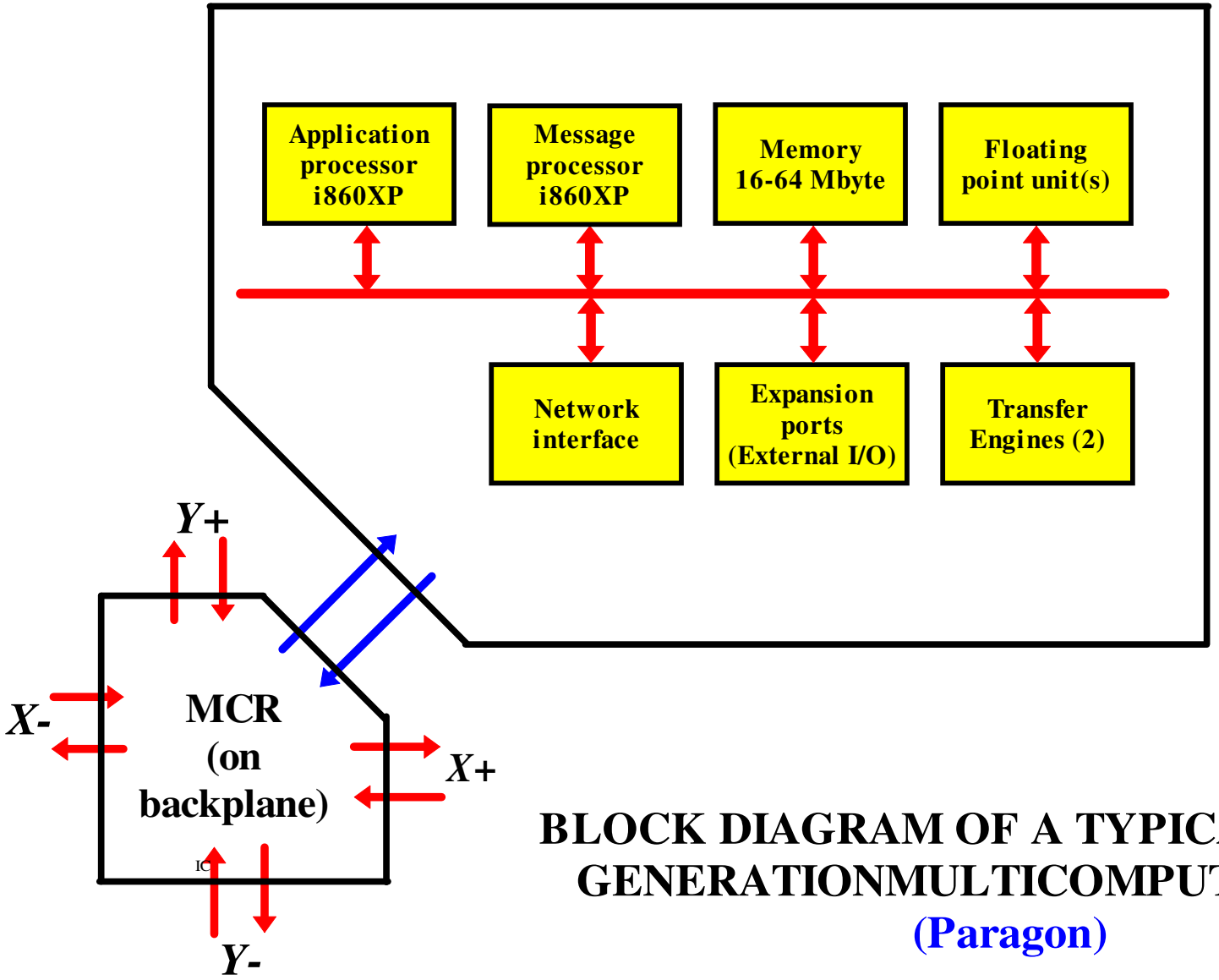


(b) Application of hardware-routed mesh interconnect. system.

ARCHITECTURE OF A SECOND GERERATION MULTICOMPUTER

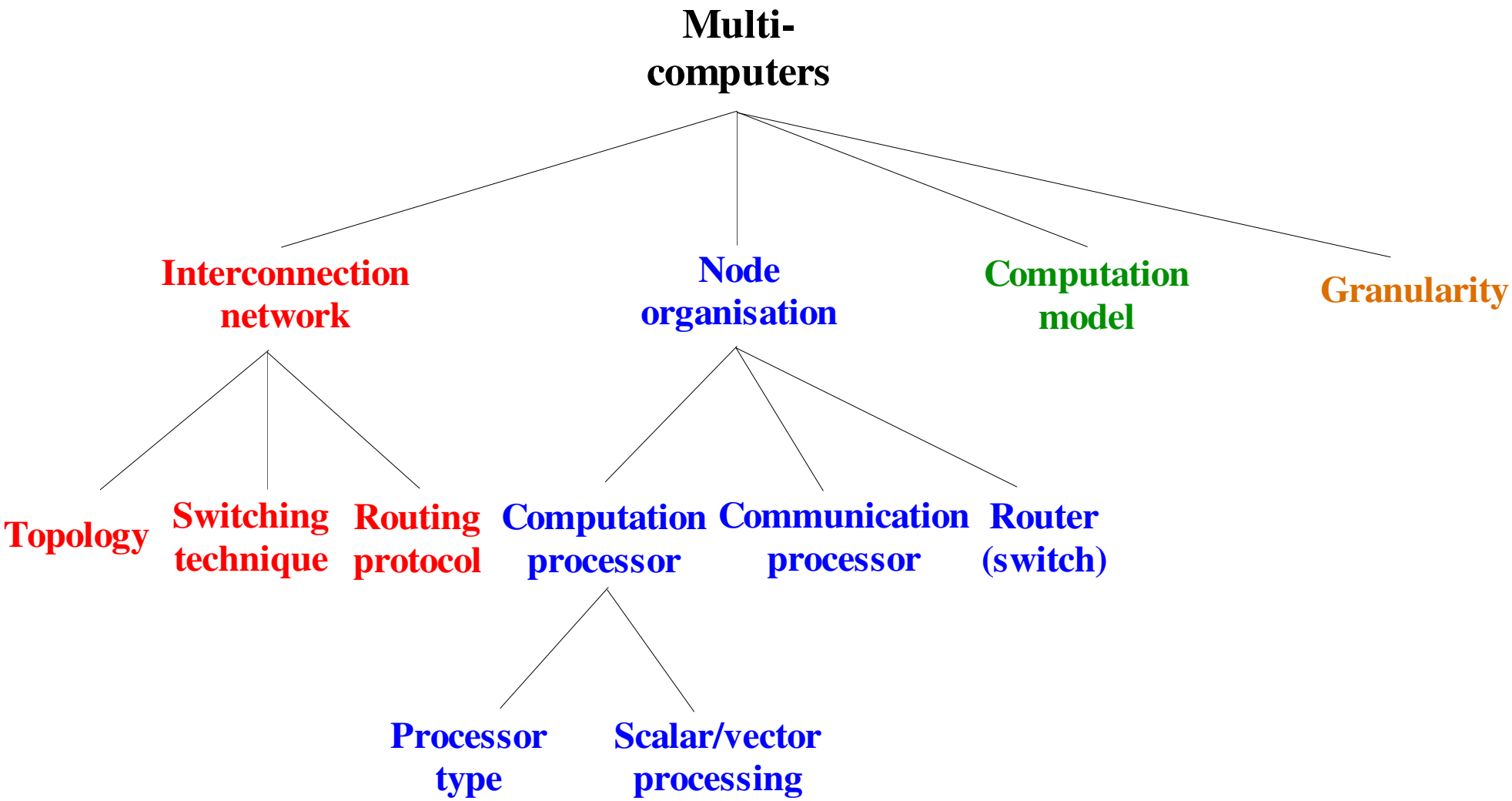
Gyula Fehér

feher@novserv.obuda.kando.hu



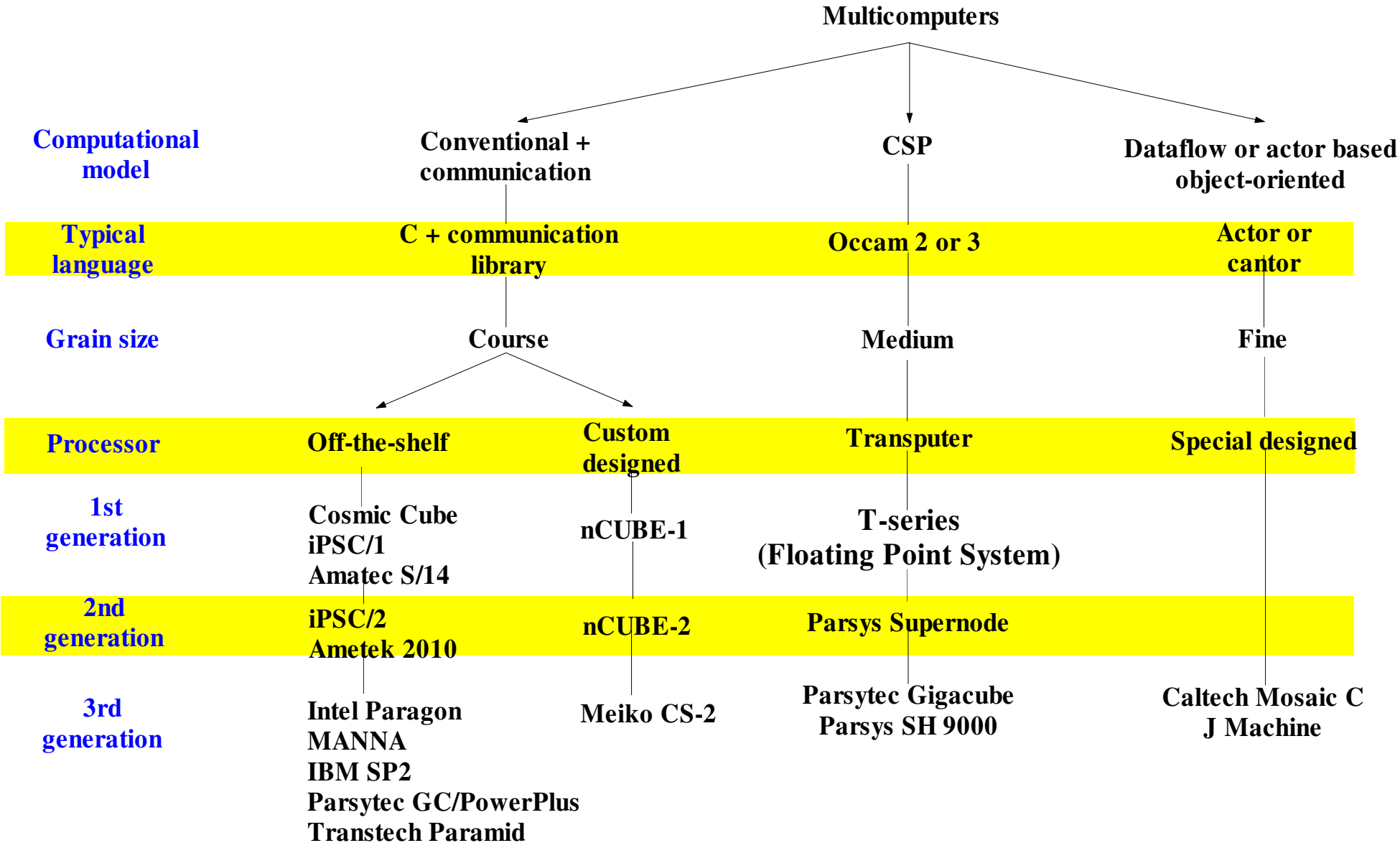
BLOCK DIAGRAM OF A TYPICAL SECOND GENERATIONMULTICOMPUTER NODE (Paragon)

Gyula Fehér feher@novserv.obuda.kando.hu



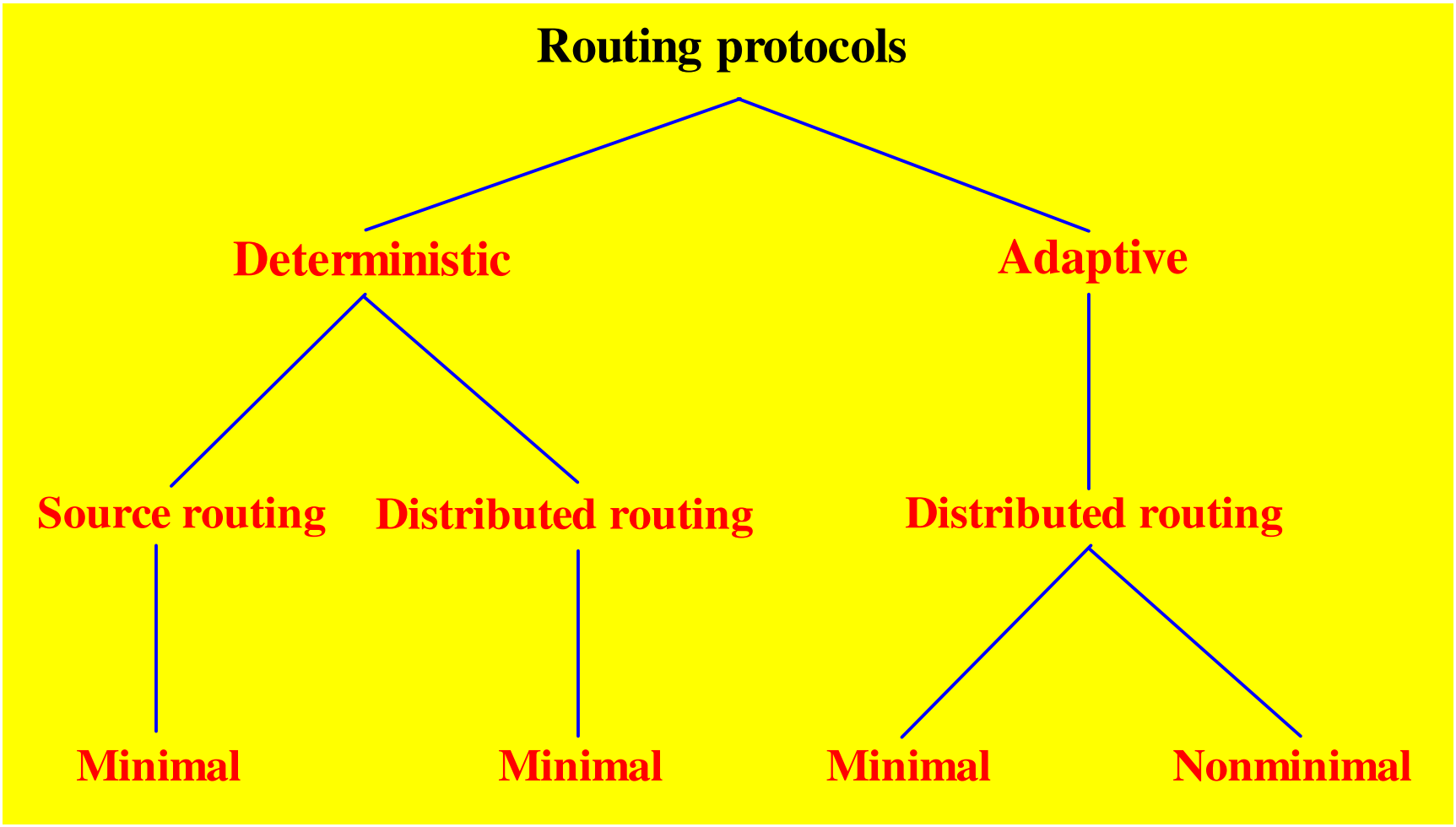
DESIGN SPACE OF MULTICOMPUTERS

Gyula Fehér feher@novserv.obuda.kando.hu



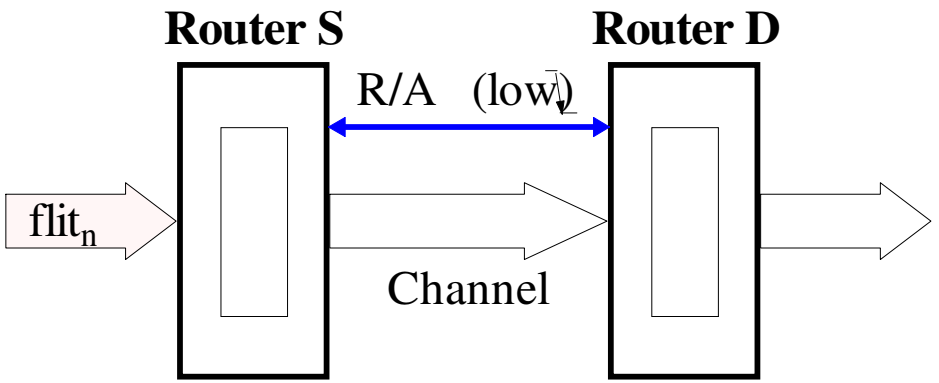
CLASSIFICATION OF MULTICOMPUTERS

Gyula Fehér feher@novserv.obuda.kando.hu

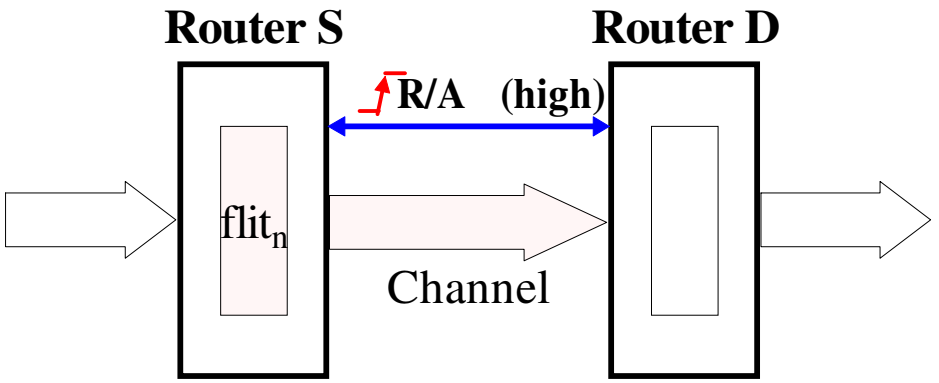


CATEGORIZATION OF ROUTING PROTOCOLS

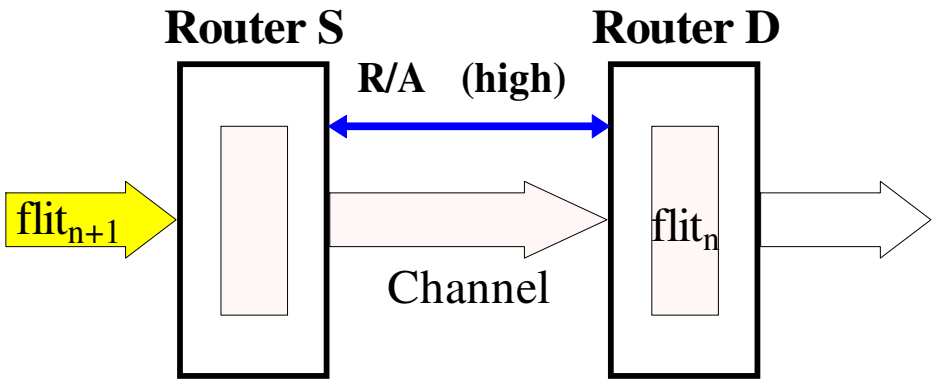
Gyula Fehér feher@novserv.obuda.kando.hu



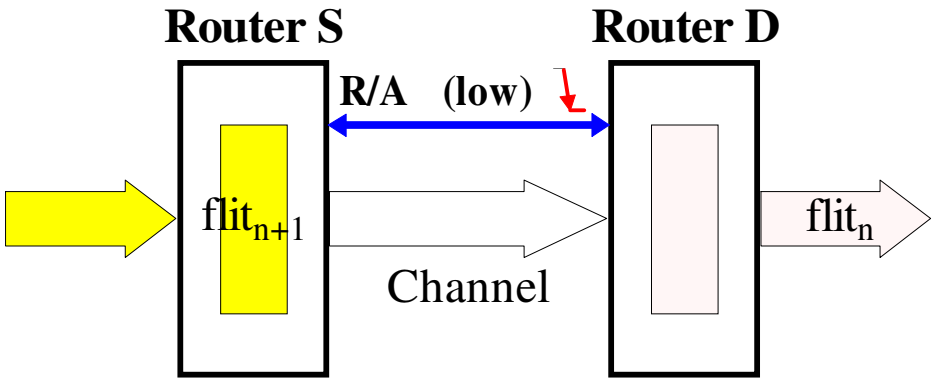
(a) D is ready to receive the next flit



(b) S is ready to send the next flit (flit$_n$)



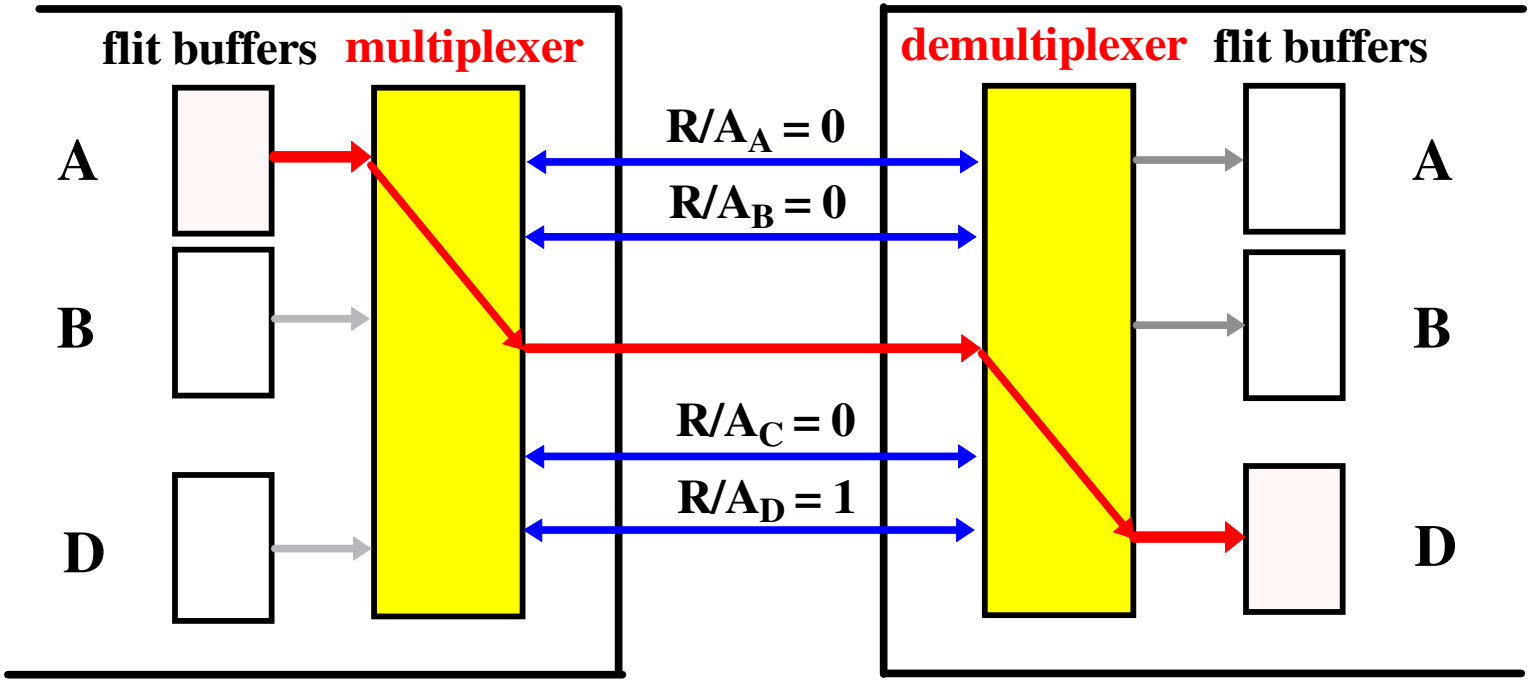
(c) Flit$_n$ is latched in D's flit buffer



(d) Flit$_n$ is removed from D's buffer and
D sets R/A low, flit$_{n+1}$ arrives at S's buffer

ASYNCHRONOUS PIPELINED HANDSHAKING PROTOCOL BETWEEN TWO ROUTERS THROUGH A REQUEST/ACKNOWLADGE LINE

Gyula Fehér feher@novserv.obuda.kando.hu

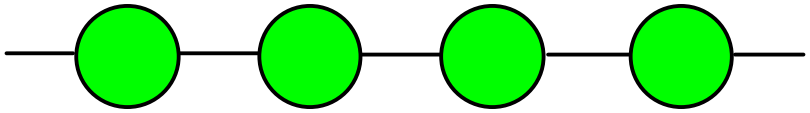


Four virtual channels share a unidirectional physical channel

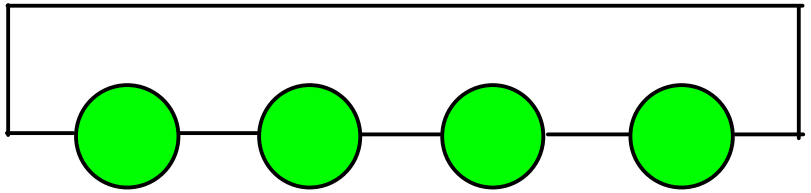
A dedicated single-bit Request/Acknowladge wire exists between an input virtual channel and an output virtual channel of two adjacent nodes. The multiplexer multiplexes data from the virtual channels over the physical channel.

ORGANIZATION OF VIRTUAL CANNELS

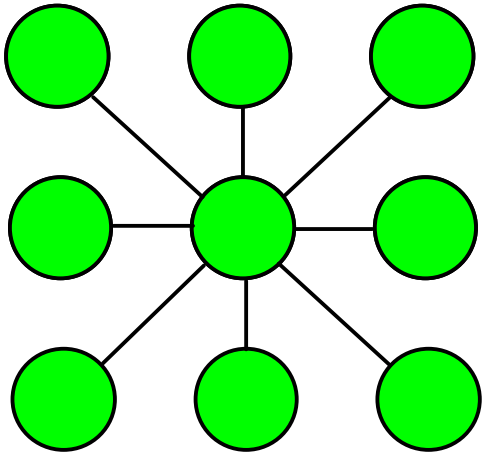
Gyula Fehér feher@novserv.obuda.kando.hu



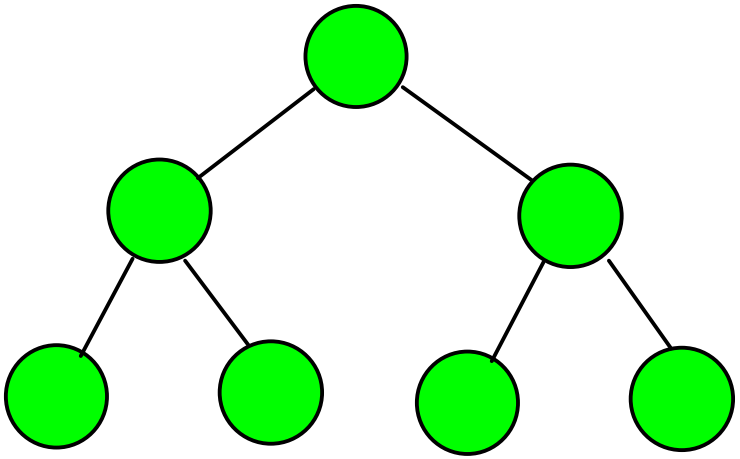
Linear array



Ring



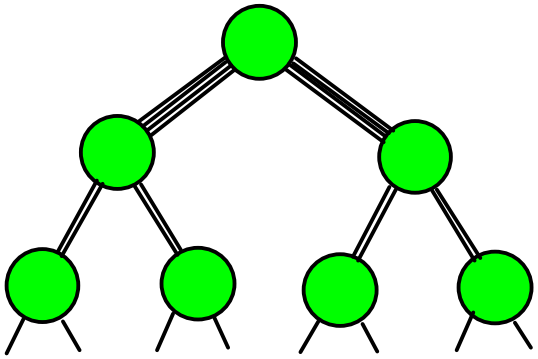
Star



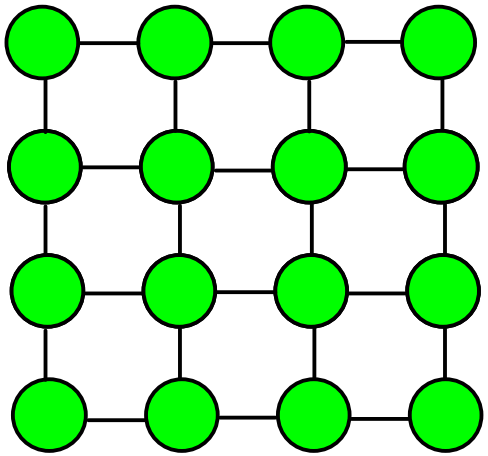
Binary tree

MAIN NETWORK TOPOLOGIES (a)

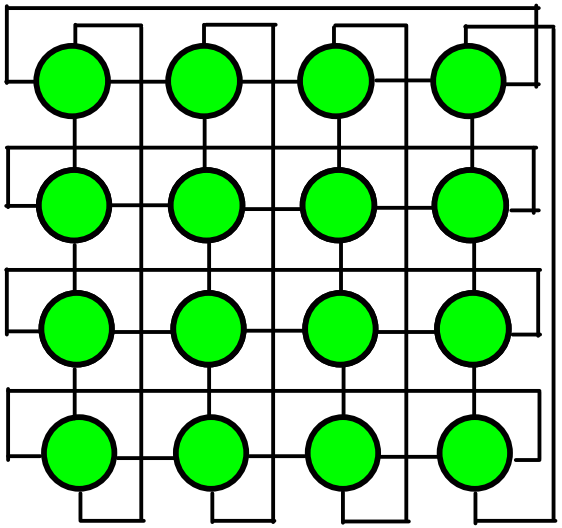
Gyula Fehér feher@novserv.obuda.kando.hu



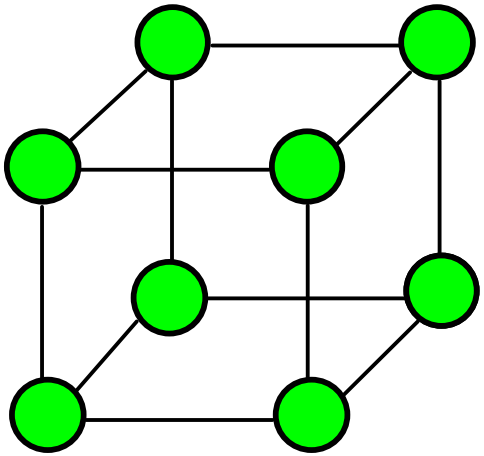
Fat tree



2-D mesh



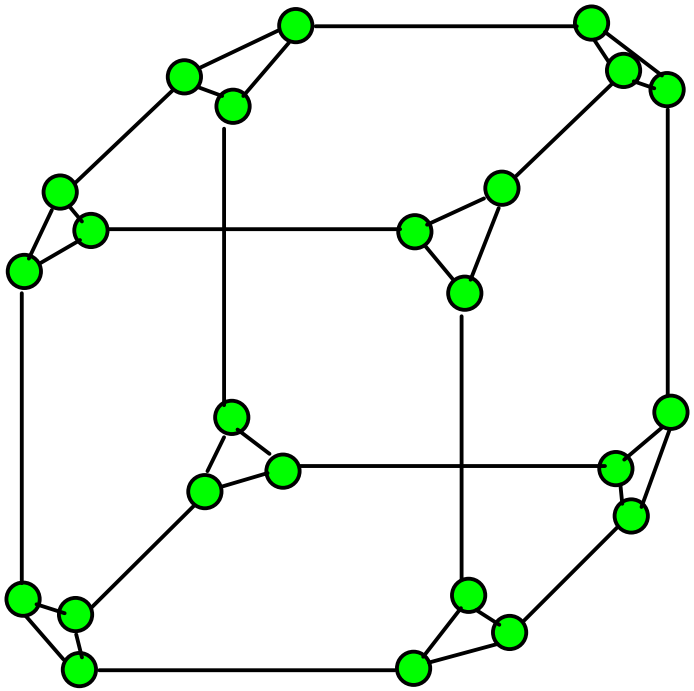
2-D wraparound mesh



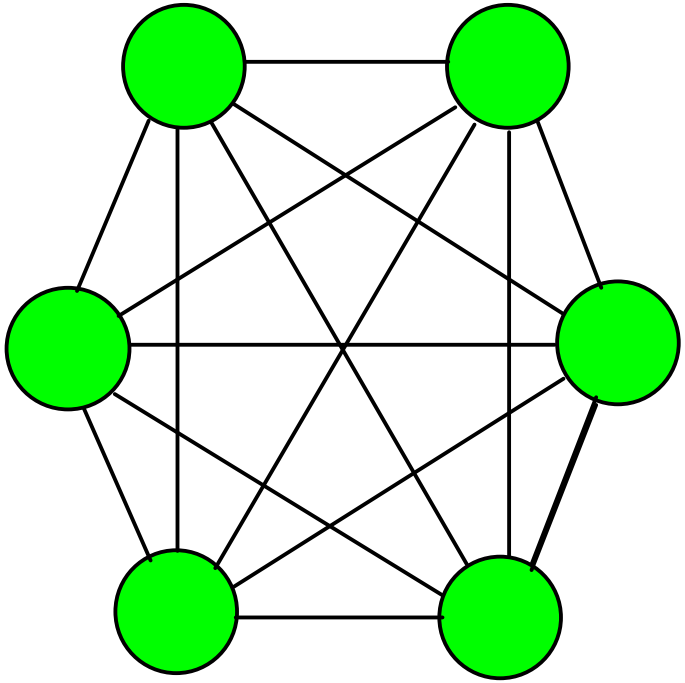
2-D mesh

MAIN NETWORK TOPOLOGIES (b)

Gyula Fehér feher@novserv.obuda.kando.hu



3-cube connected cycle



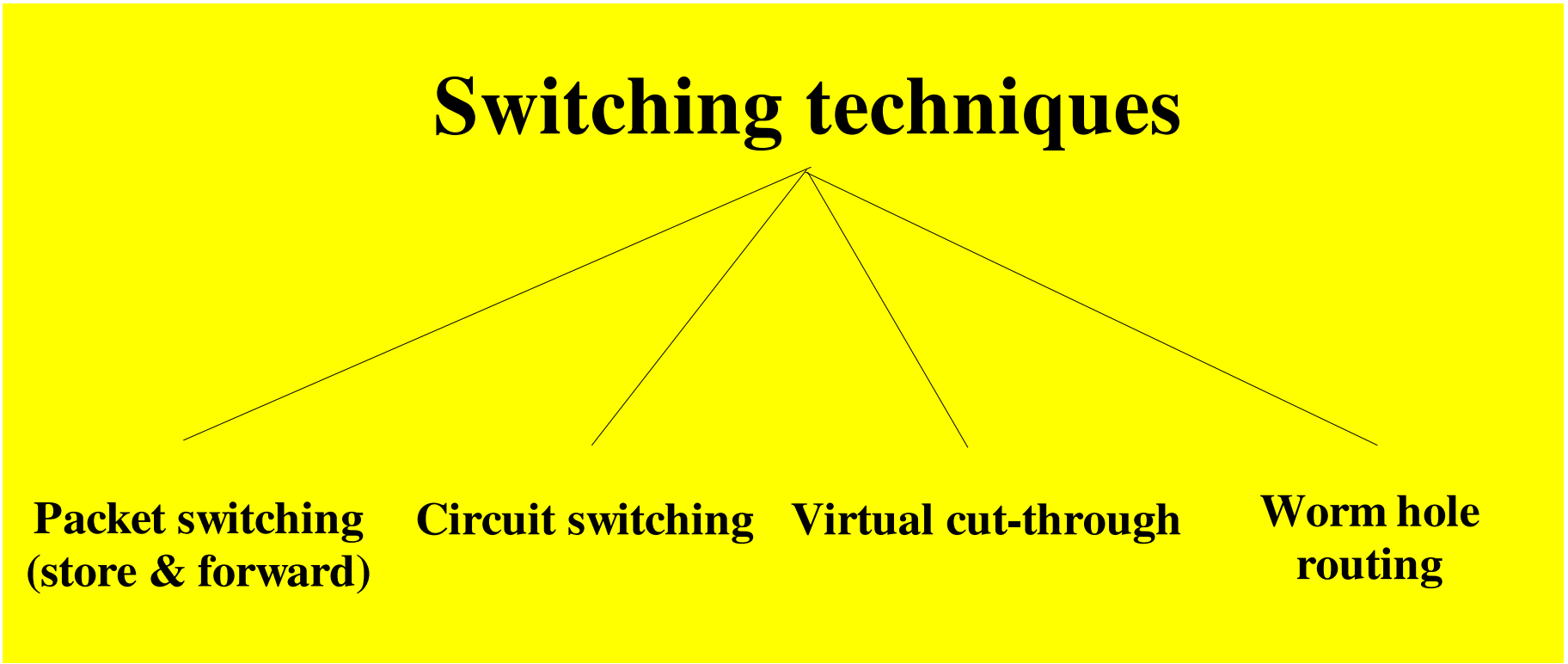
Completely connected

MAIN NETWORK TOPOLOGIES (c)

Gyula Fehér feher@novserv.obuda.kando.hu

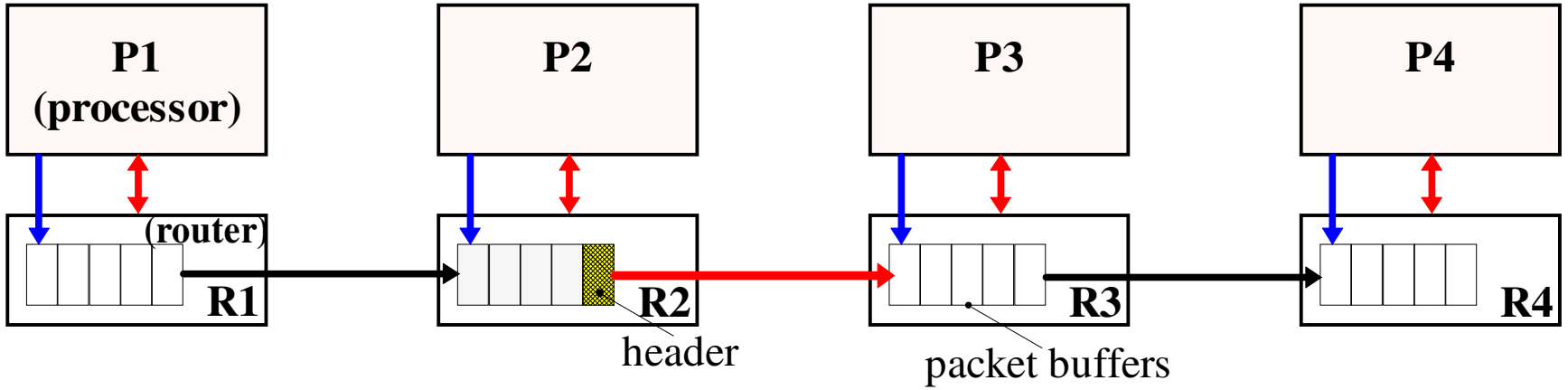
Topology	*Node degree*	*Diameter*	*Bisection width*	*Arc connectivity*	*Cost*
Linear array	1 or 2	N-1	1	1	N-1
Ring	2	N/2	2	2	N
Star	1 or N-1	2	1	1	N-1
Binary tree	1, 2 or 3	2log((N+1)/2)	1	1	N-1
2-D mesh	2, 3 or 4	$2(N^{1/2}-1)$	$N^{1/2}$	2	$2(N-N^{1/2})$
2-D wraparound mesh	4	$N^{1/2}$	$2N^{1/2}$	4	2N
3-D cube	3, 4, 5 or 6	$3(N^{1/3}-1)$	$N^{2/3}$	3	$2(N-N^{2/3})$
Hypercube	logN	logN	N/2	logN	(NlogN)/2
Completely connected	N-1	1	$N^2/4$	N-1	N(N-1)/2

Summary of static network parameters for the main topologies

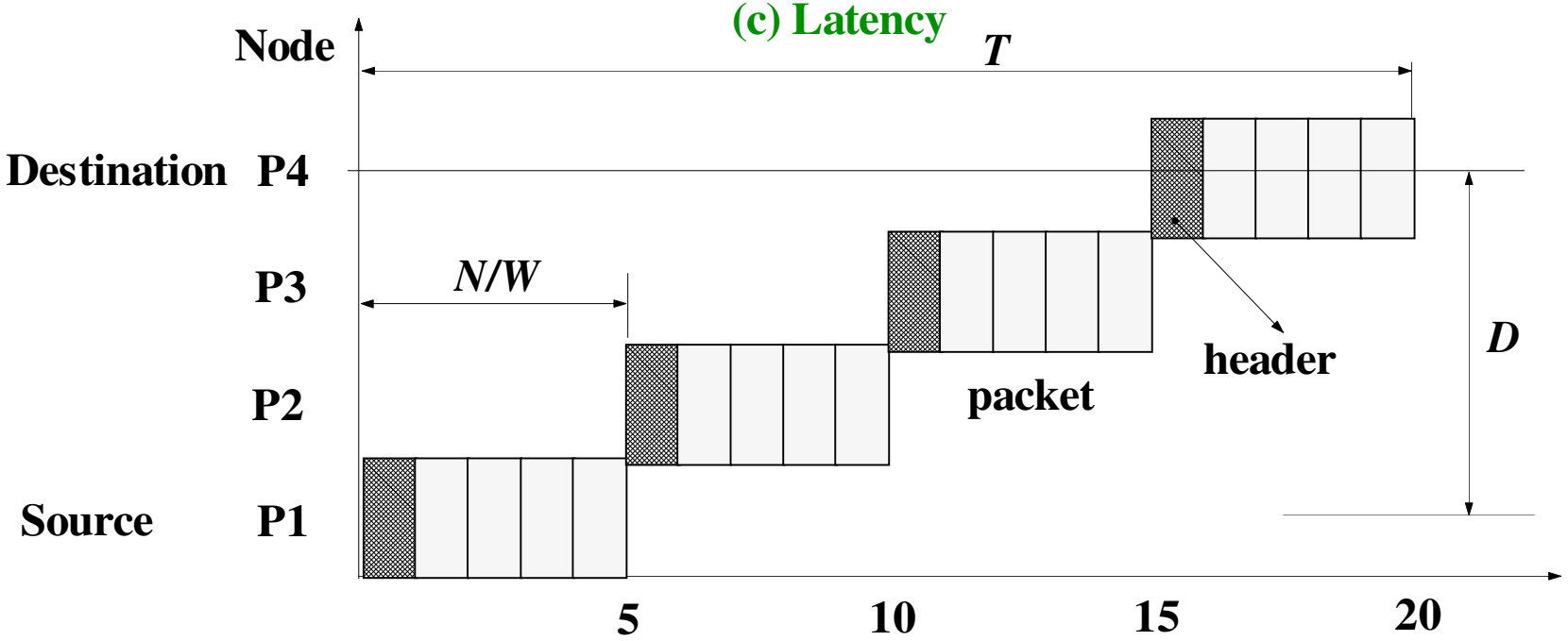


DESIGN SPACE OF SWITCHING TECHNIQUES

(a) Arrangement



(c) Latency



PACKET SWITCHING (a)

Gyula Fehér

feher@novserv.obuda.kando.hu

Packet switching message-passing mechanism

- All the messages are divided into *packets* which are sent independently via the communication network between the source and destination nodes.
- The mesages are transmitted in a *store-and-forward* fashion (each byte contained in a message had to be stored at each node along a route and forwarded to the next hope.)
- A packet consists of a header and the data .A *header* contains the necessary routing information based on that the switching unit decides where to forward the packet.
- When a packet arrives at an intermediate node, the whole packet is stored in a packet buffer.
- Latency:

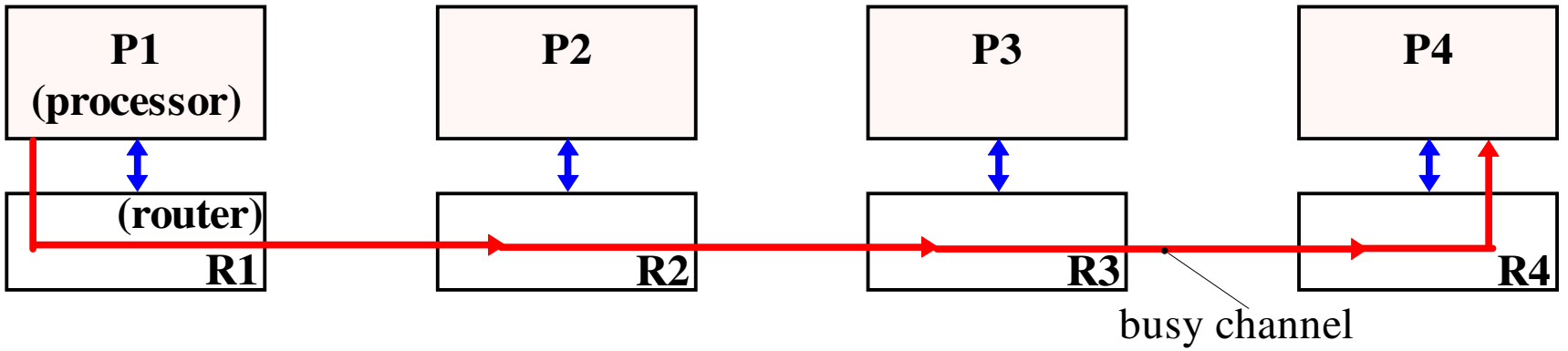
$$L = \frac{P}{B} D$$

- Where P is the packet lenght, B is the channel bandwidth and D is the distance (number of nodes).
- Main drawback: latency is proportional to the message path lenght. (This was the reason why the diameter was the most important parameter in the first generation multicomputers, and why the hypercube was so popular.

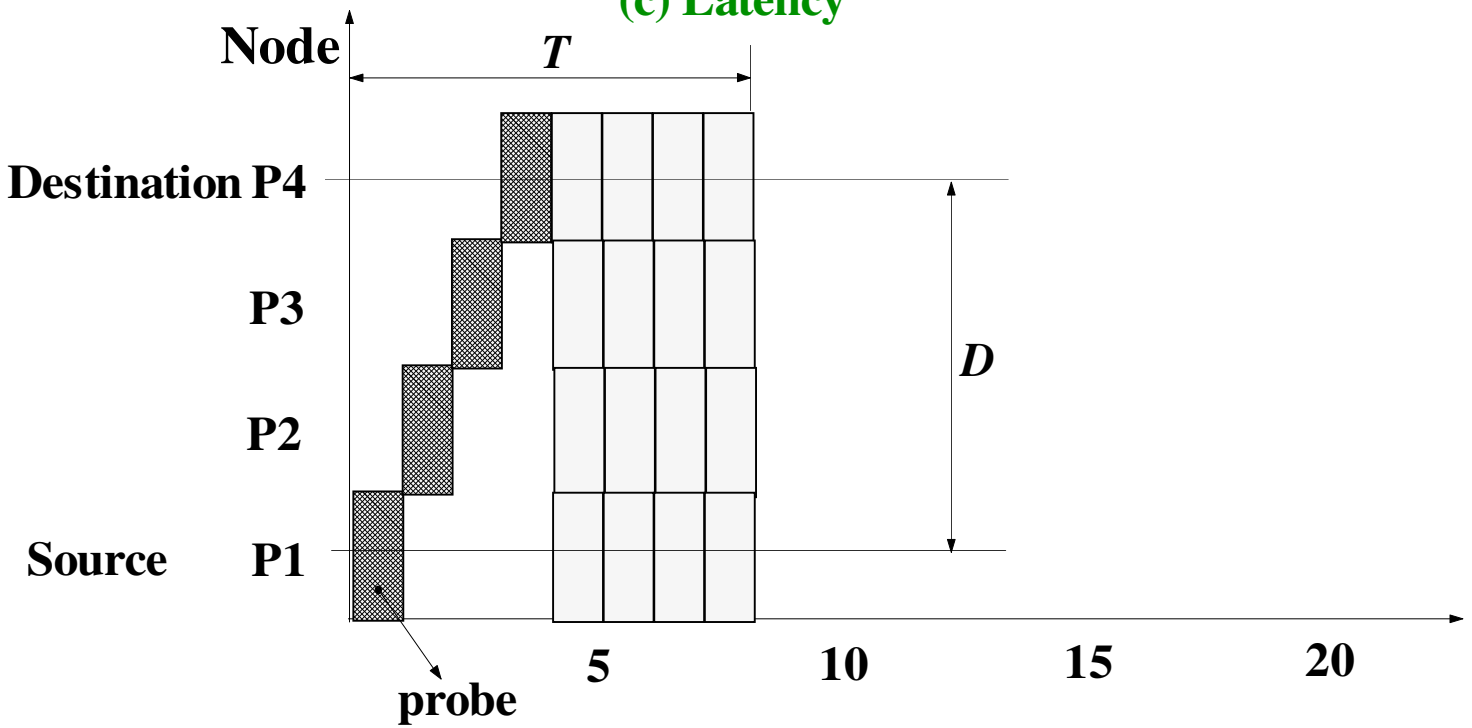
PACKET SWITCHING (b)

Gyula Fehér feher@novserv.obuda.kando.hu

(a) Arrangement



(c) Latency



CIRCUIT SWITCHING (a)

Gyula Fehér

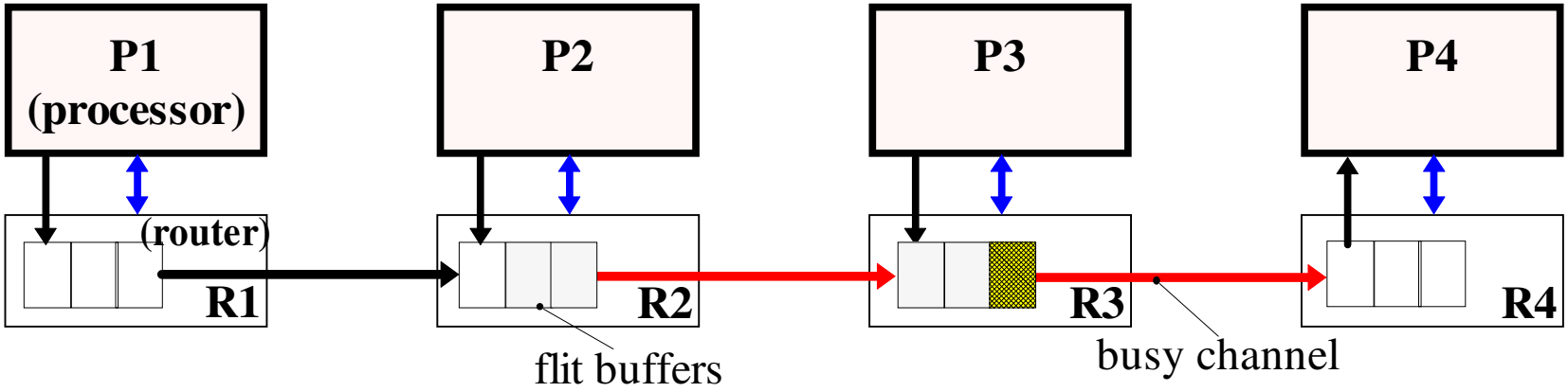
feher@novserv.obuda.kando.hu

Circuit swithing

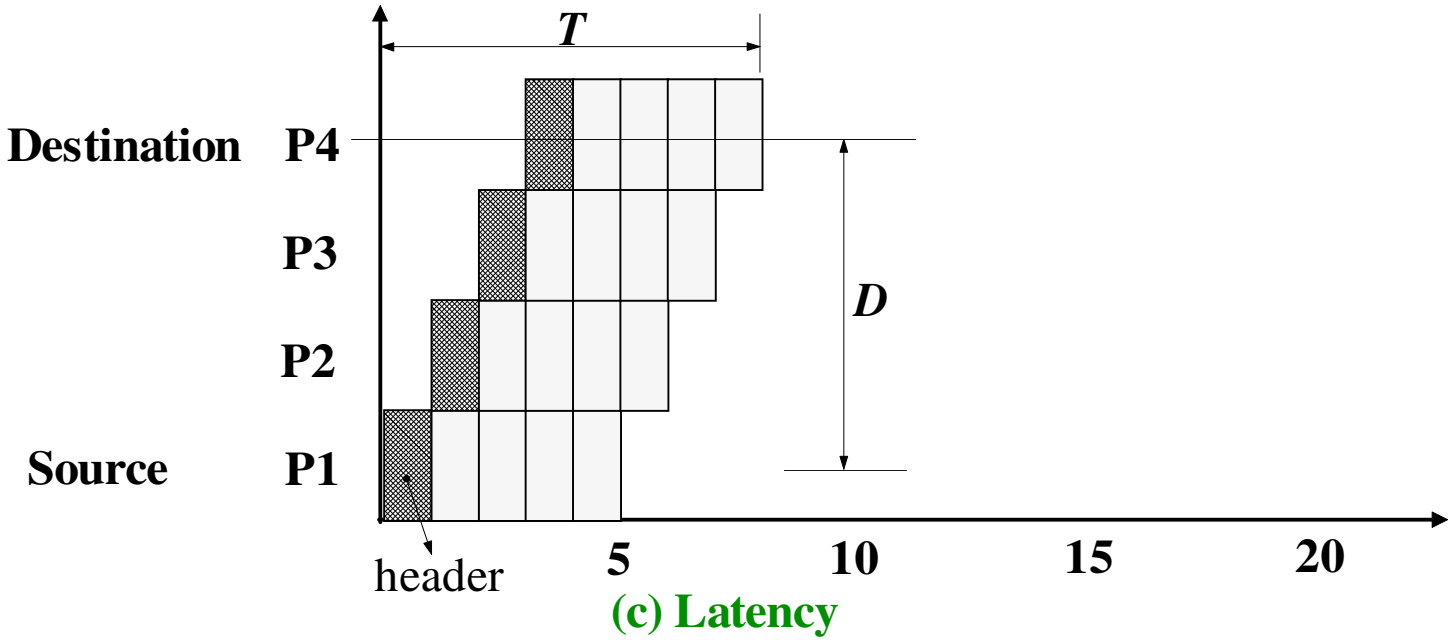
- In the first phase of the communication a path (a *circuit*) is built up between the source and destination by sending a special short message (called probe).
- The *probe* has similar function as the header of packets in the packet swiching system.
- The circuit is held until the entire message is transmitted
- During the communication the channels constituing the circuit are reserved exclusively, no other messages can be transmitted by them.
- In the last phase the circuit is torn down either by the tail of transmitted message or by an acknowladgement message returned by the destination node.
- If a desired channel is used by another circuit in the circuit establishment phase, the partially built up circuit may be torn down.
- Circuit switching does not need packetizing. No matter what the message size is.
- There is no need for buffering.
- Latency:

$$L = \frac{P}{B} D + \frac{M}{B}$$

  Where P is the lenght of the probe and M is the lenght of the message.
- The most important benefit: If the lenght of the probe, P is much smaller then the lenght of the message, M then the latency becomes independent of the communication distance.



(a) Arrangement



(c) Latency

VIRTUAL CUT-THROUGH SWITCHING (a)

Gyula Fehér feher@novserv.obuda.kando.hu

Virtual cut-through switching

- Attemp to combine the benefits of packet switching and circuit switching
- The message is divided in to small units called flow control digits, or *flits*.
- As long as the required channels are free, the message is forwarded flit by flit among the nodes in a pipeline fashion.
- If the required channel is busy, flits are buffered at intermediate nodes.
- If the buffers are large enough, the entire message is buffered at the blocked intermediate node, resulting a behaviour similar to packet switching.
- If the buffers are not large enough, the message will be buffered across several nodes, holding the links between them.
- The network latency:

$$L = \frac{HF}{B} D + \frac{M}{B}$$

Where HF is the lenght of the header flit.

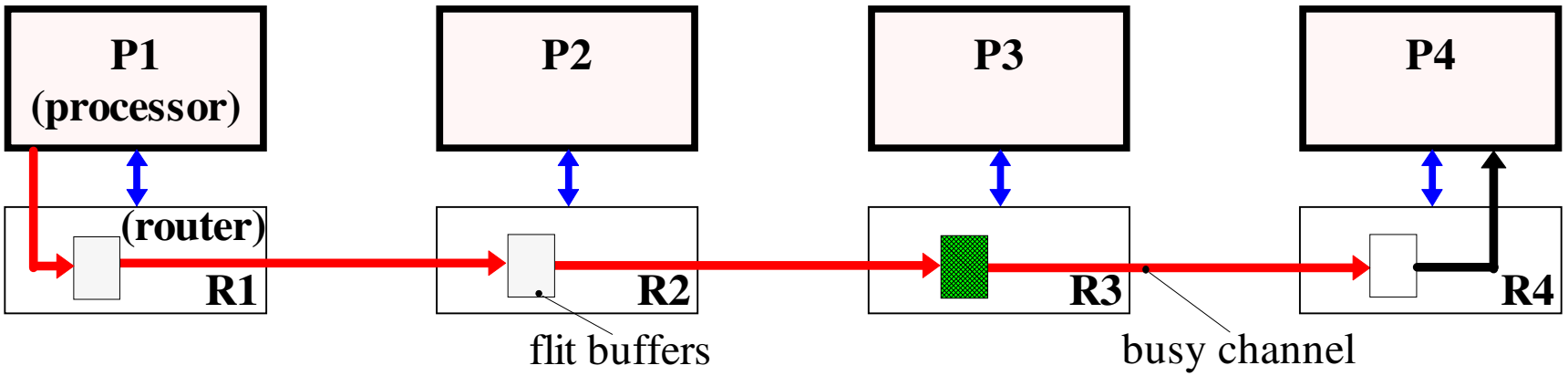
- Main benefit: If $HF << M$, the latency becomes independent of the distance, D.

VIRTUAL CUT-THROUGH SWITCHING

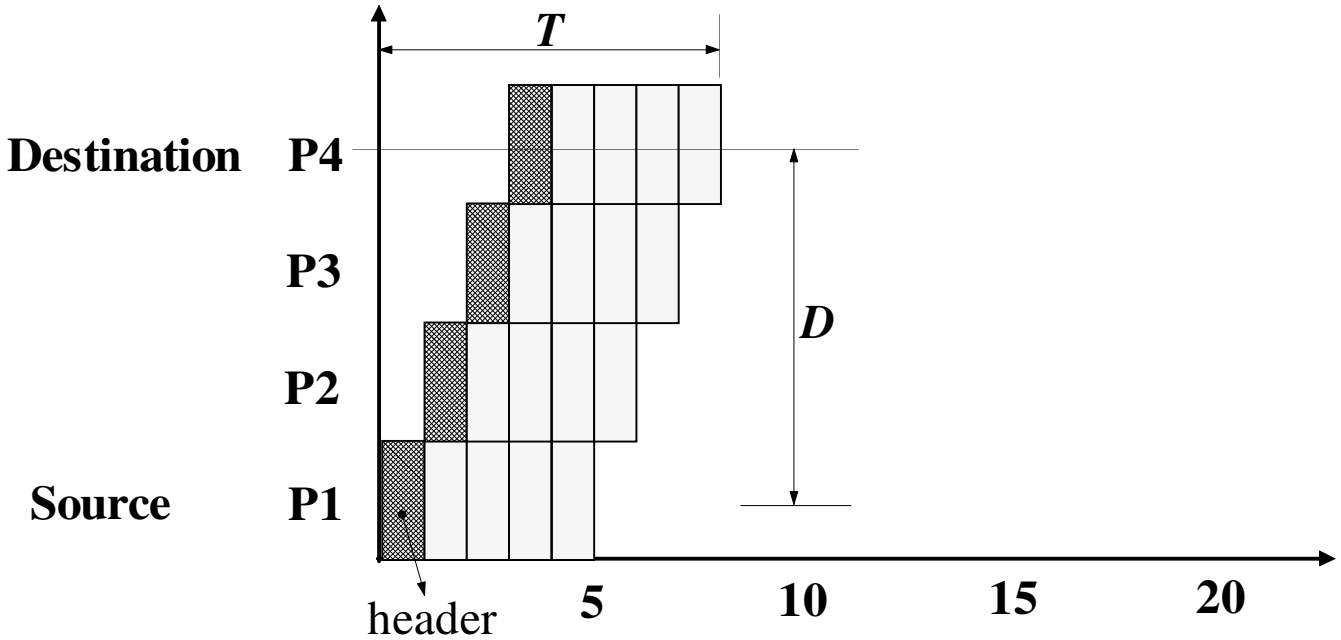
(b) Main features

Gyula Fehér feher@novserv.obuda.kando.hu

WORM-HOLE SWITCHING (a)



(a) Arrangement



(c) Latency

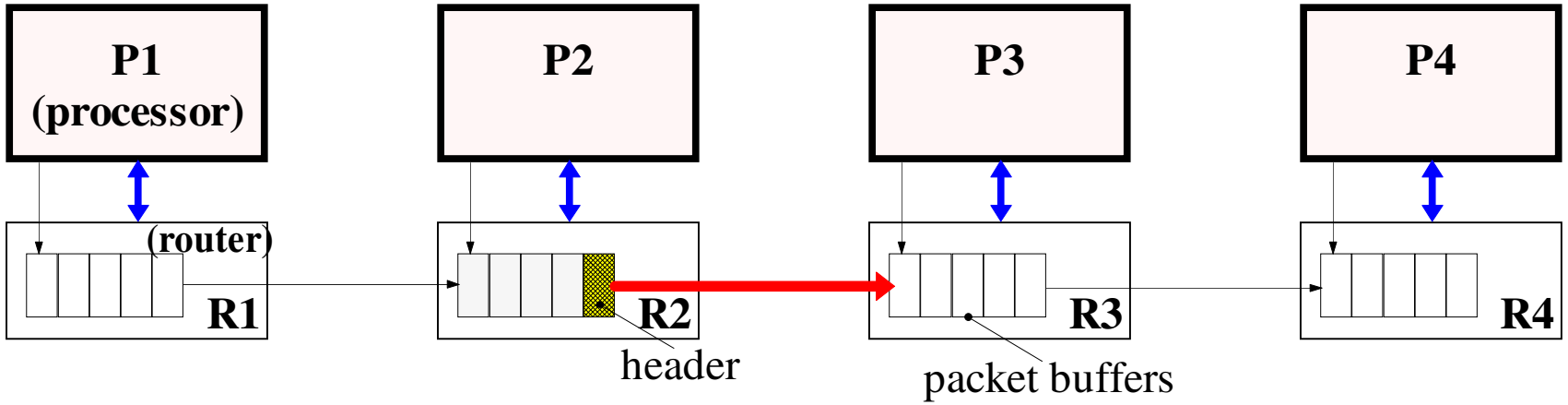
Worm-hole switching

- Do not create a circuit between sender and receiver. Instead, an initial control message at the start of the message establishes a path through the network and all subsequent data for that message are forwarded along that path.
- The message is broken into very small pieces (*flits*) and the network is pipelined. This is referred to as worm-hole routing due to the way that a message *worms* its way through the system.
- A special case of virtual cut-through, where the buffers at the intermediate nodes have the size of a flit.
- There is a start-up overhead related to distance, the entire message is not penalised due to the pipelining. Thus, a message transmission time for a worm-hole routed network is of the form:

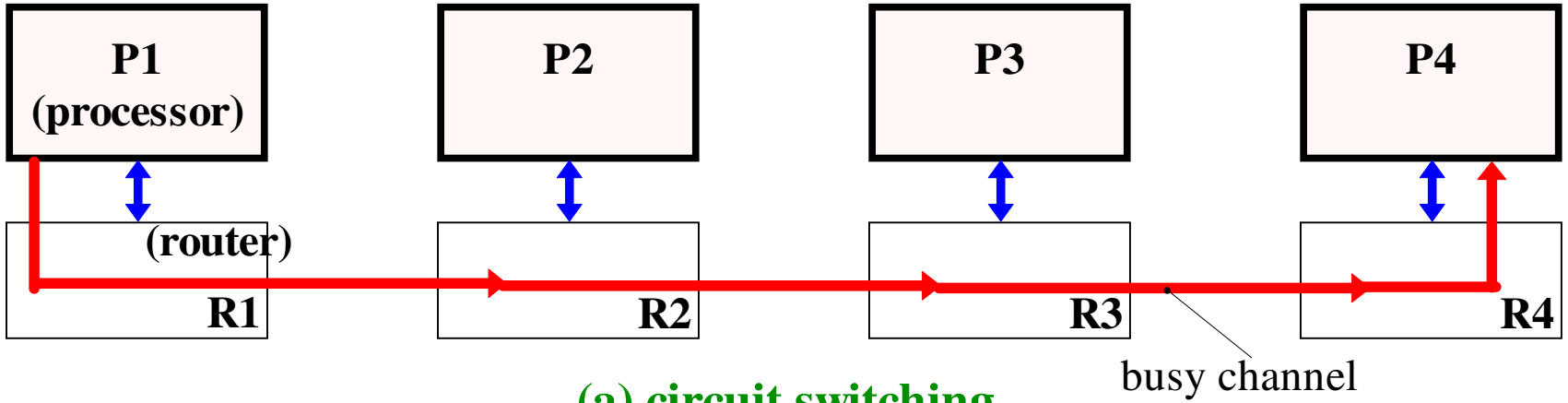
$$T = C_2 \cdot (D-1) \cdot P + C_2 \cdot N$$

where P is the packet size, N is the message lenght, D is the source to destination distance, and C_2 is a constant, representing the data rate of the network. If P is small relative to N, T will be similar to that for the circuit switching system in that T is not very dependent on D.

- The primary advantage of such a network is that links need not be blocked for the entire message duration, and (after introducing the *virtual channel* concept) it is possible to multiplex messages along individual links.



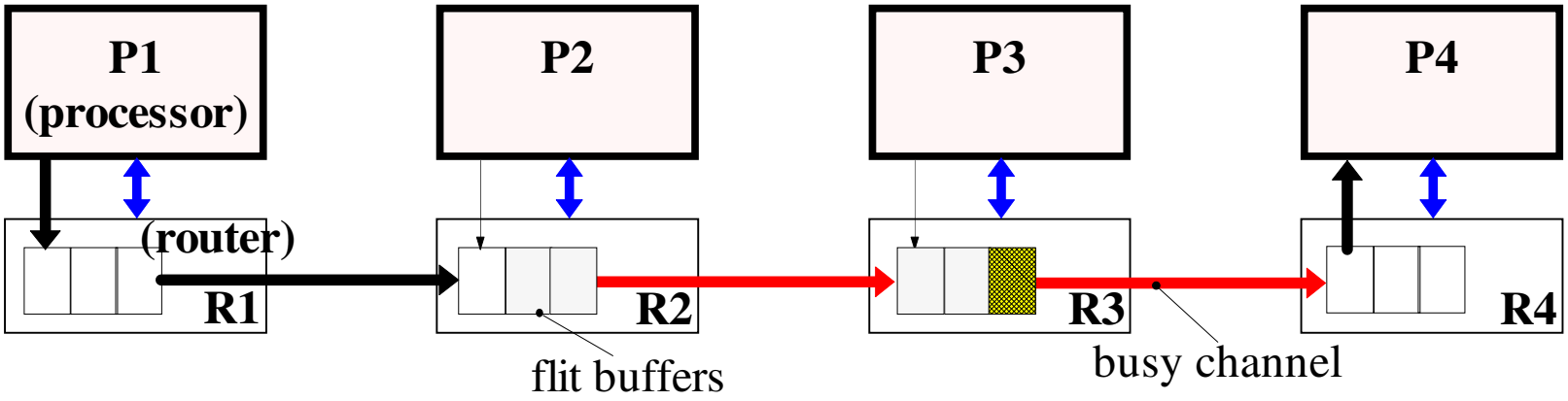
(a) packet switching



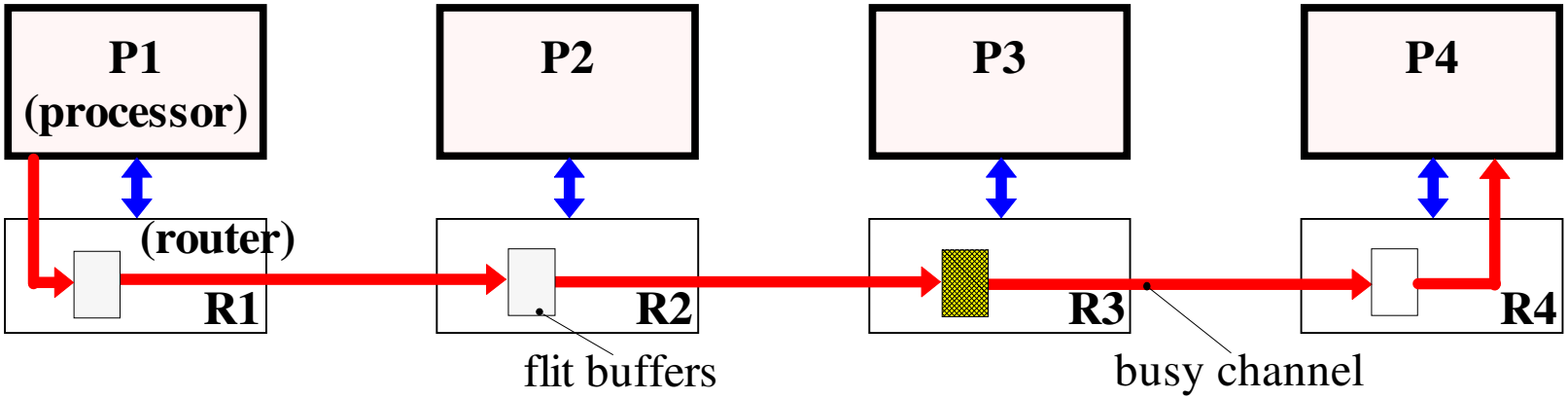
(a) circuit switching

COMPARING SWITCHING TECHNIQUES (a)

Gyula Fehér feher@novserv.obuda.kando.hu



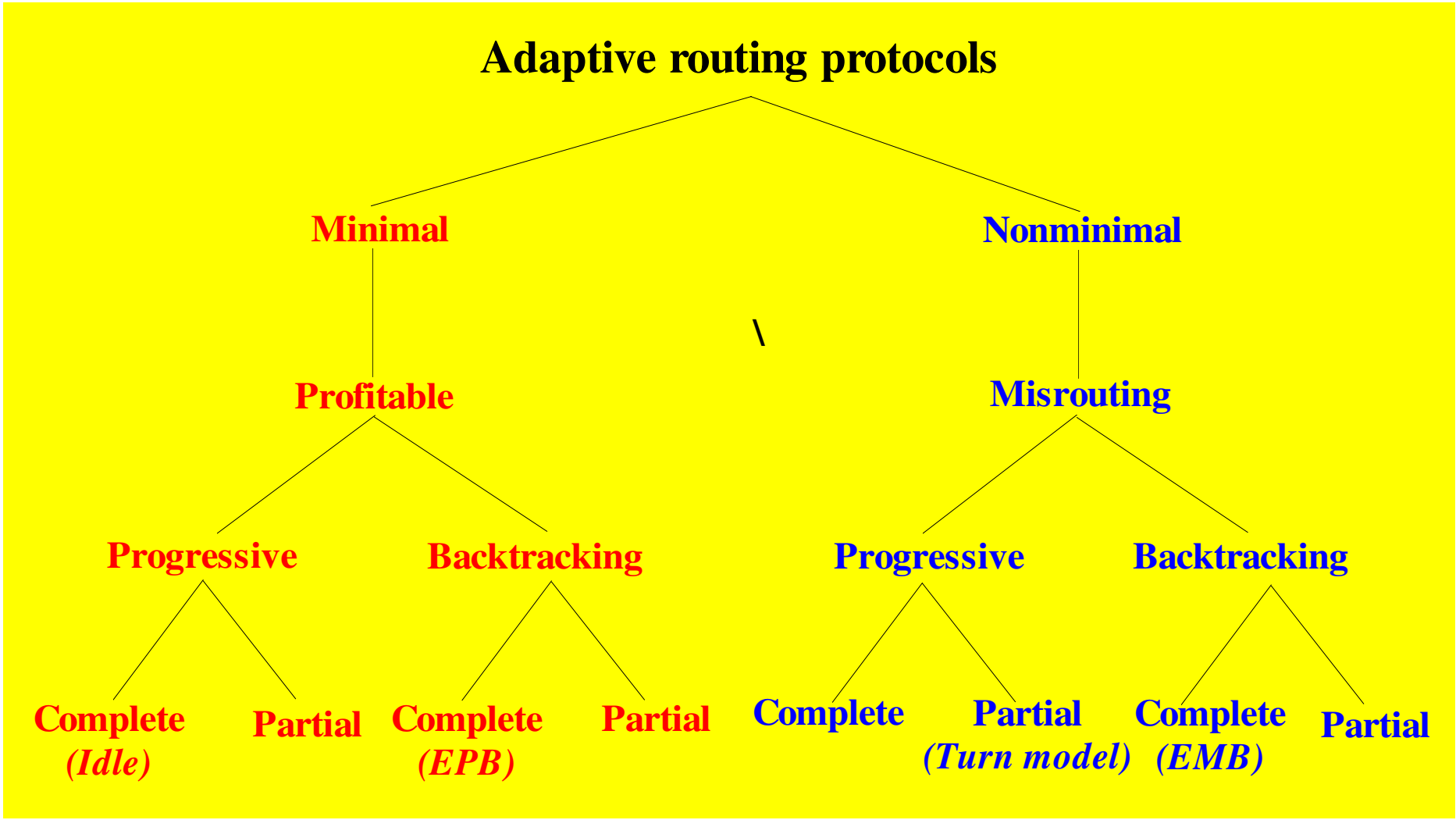
(b) virtual cut-through



(c) Worm hole routing

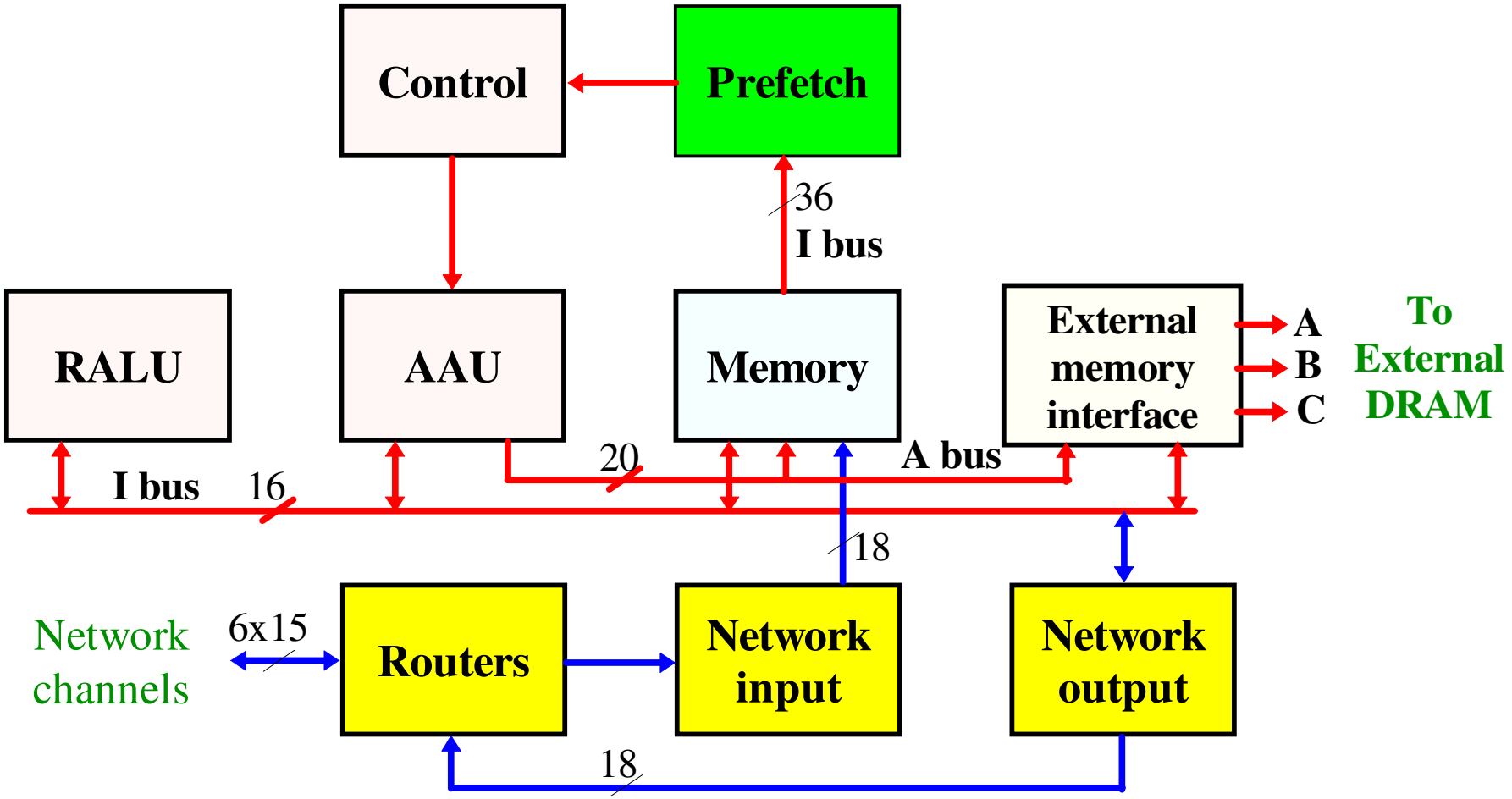
COMPARING SWITCHING TECHNIQUES (b)

Gyula Fehér feher@novserv.obuda.kando.hu



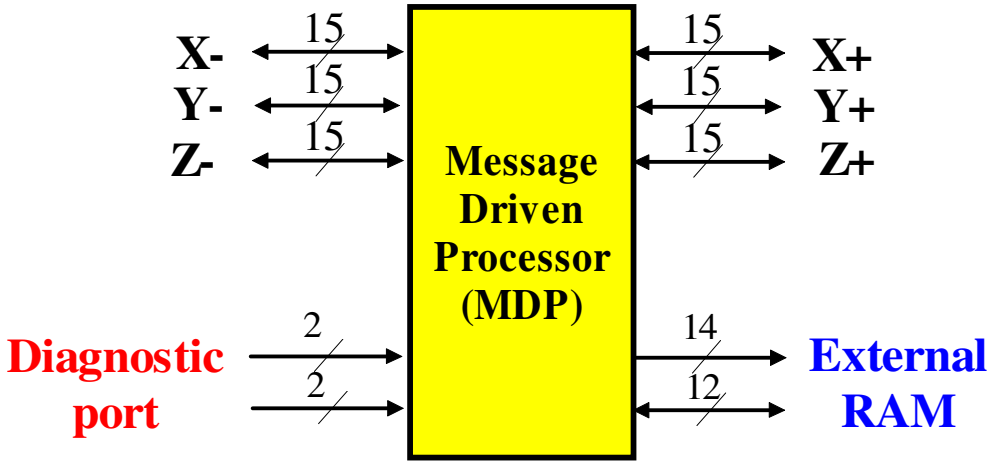
CATEGORIZATION OF ADAPTIVE ROUTING PROTOCOLS

Gyula Fehér feher@novserv.obuda.kando.hu



Internal structure of the MDP

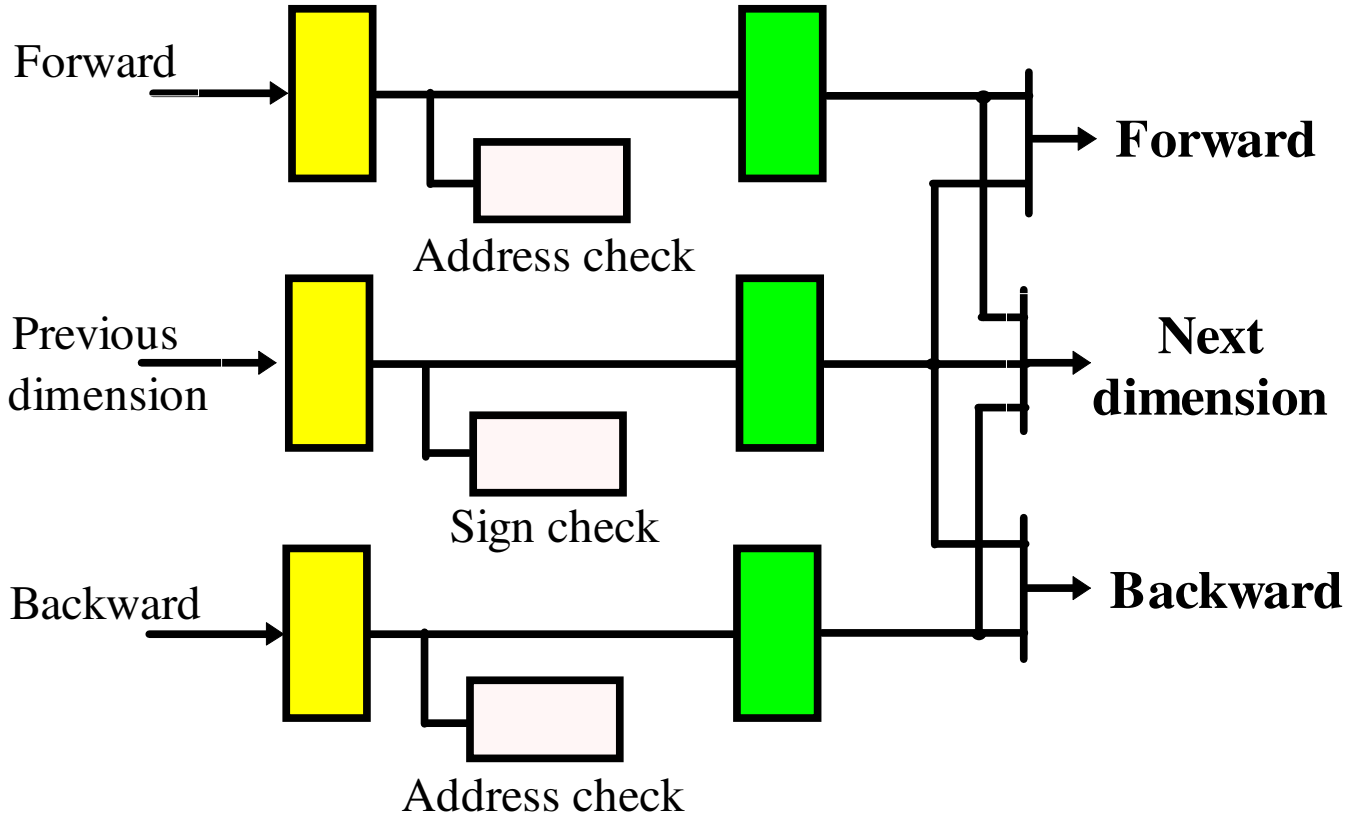
- J-Machine supports global virtual address space and
- efficient primitive mechanisms for communication, synchronization and naming.
- It attempts to be able to tolerate communication latency by implementing message-driven processors.
- These processors are message driven in the sense that *they begin execution in response to messages*, via the dispatche mechanism.
- No receive instruction is needed, and this eliminates some of the communication overhead.
- The sending of messages consumes no processing resources on intermediate nodes.
- Each node of it consists an MDP and 1 MByte of DRAM



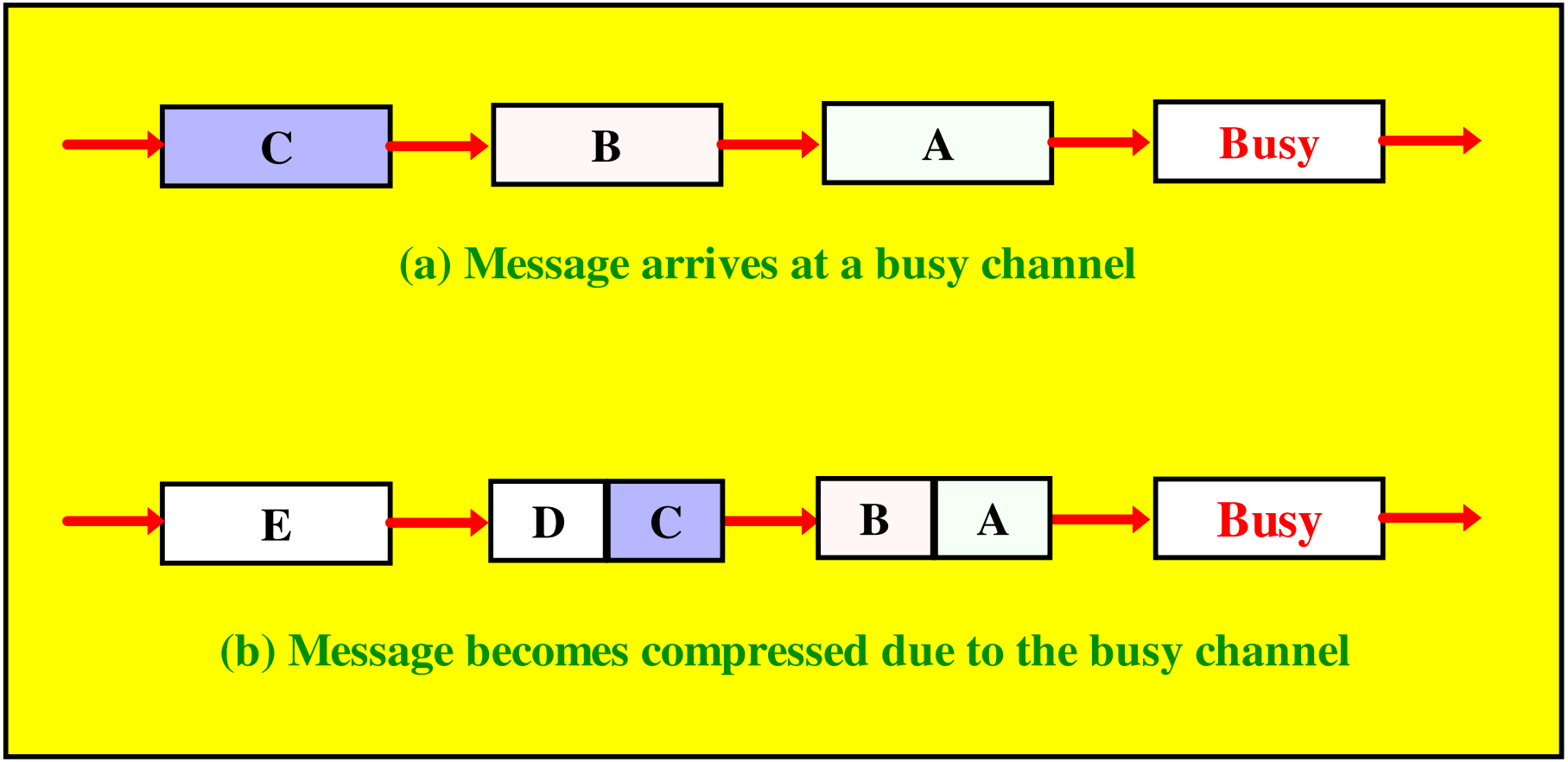
Pin layout of the MDP

The chip is called Message Driven Processor since, it processes in response to messages without having any receive instruction

Internal structure of the MDP



Structure of the routing unit for one directions



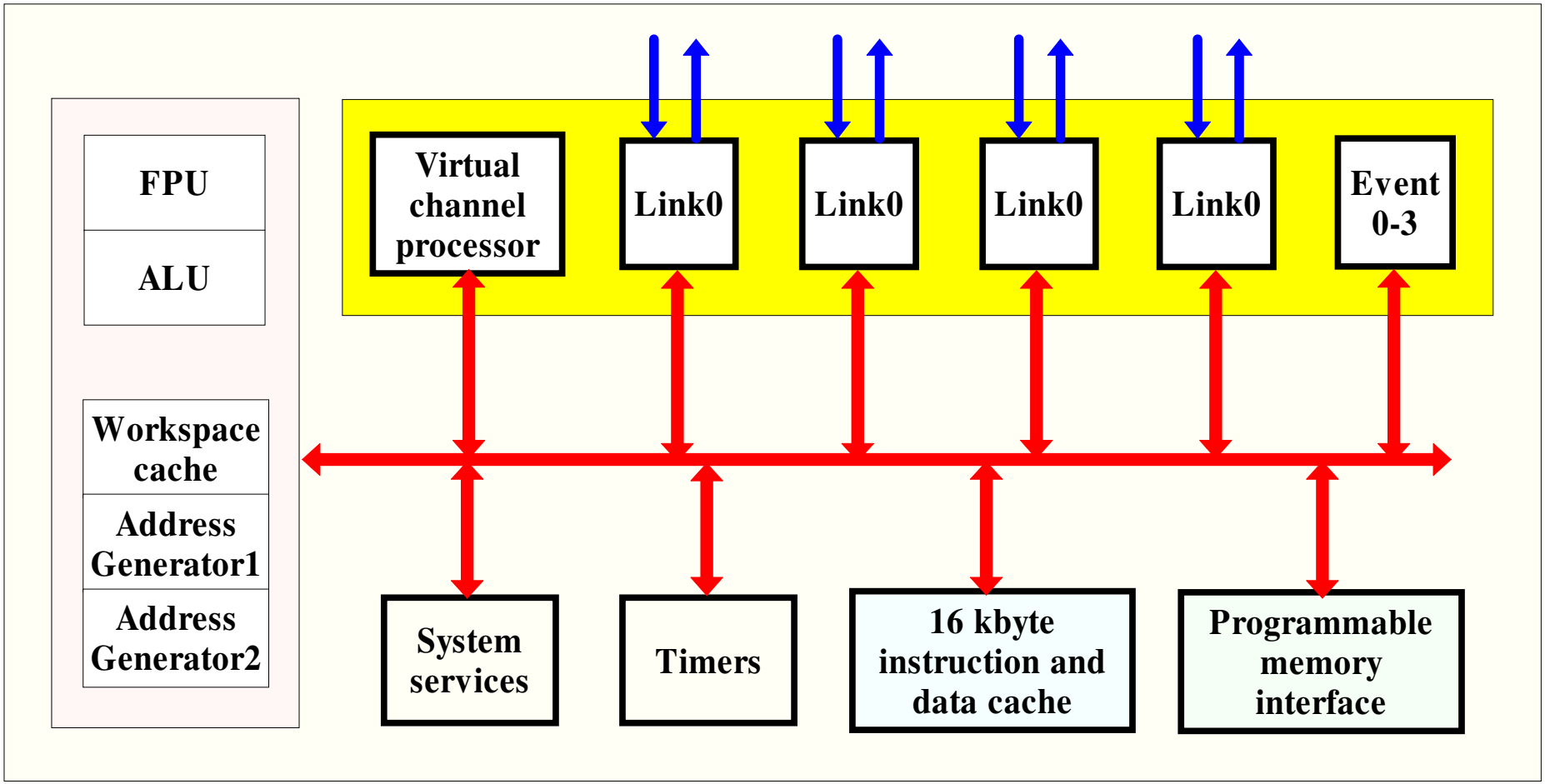
MESSAGE COMPRESSION DUE TO BUSY CHANNELS

Gyula Fehér feher@novserv.obuda.kando.hu

Machine	T1:μs/msg	T2:μs/byte	cycles/msg	cycles/byte
nCUBE/2 (Vendor)	160.0	0.45	3200	9
CM-5 (Vendor)	86.0	0.12	2838	4
nCUBE/2 (Active)	23.0	0.45	460	9
CM-5 (Active)	3.0	0.12	109	4
J-Machine	0.9	0.04	11	0.5

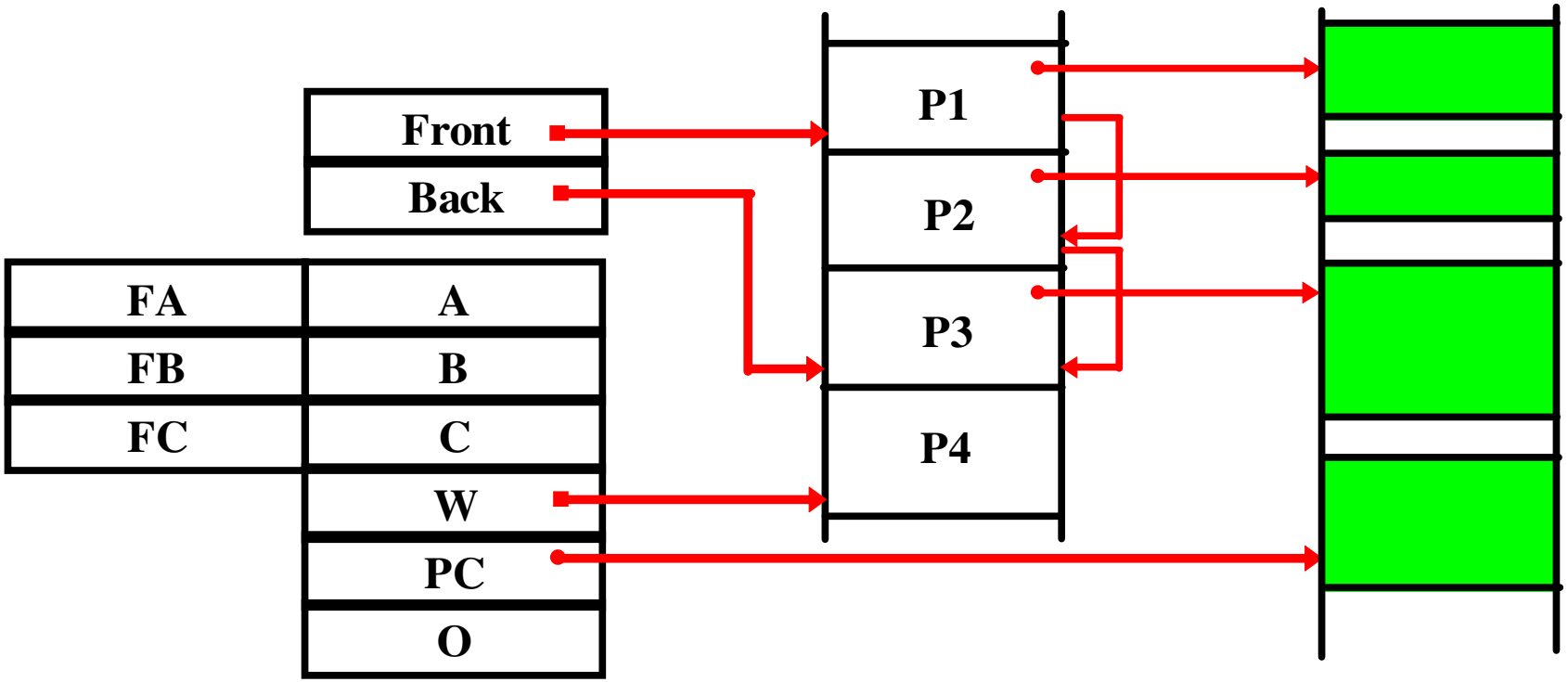
***Legend*: T1 is the sum of the fixed overheads of send and receive**
T2 is the injection overhead per byte

One way message owerhead



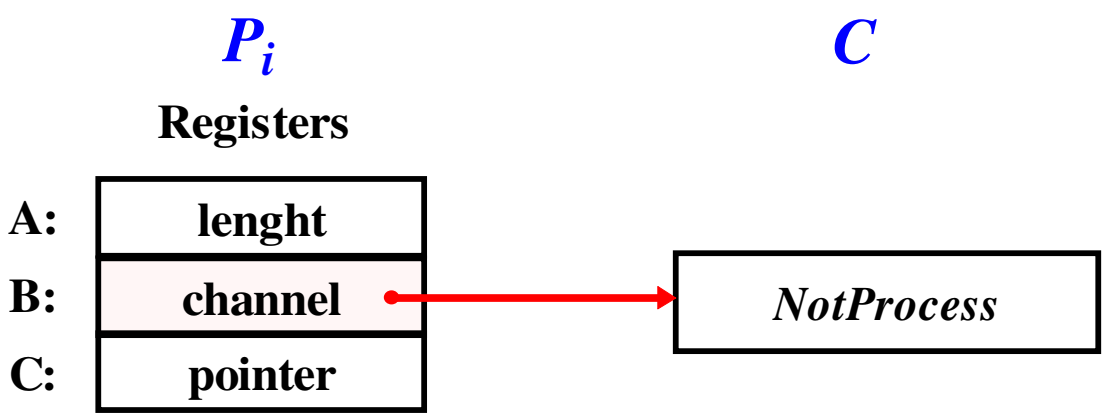
STRUCTURE OF THE T9000

Gyula Fehér feher@novserv.obuda.kando.hu

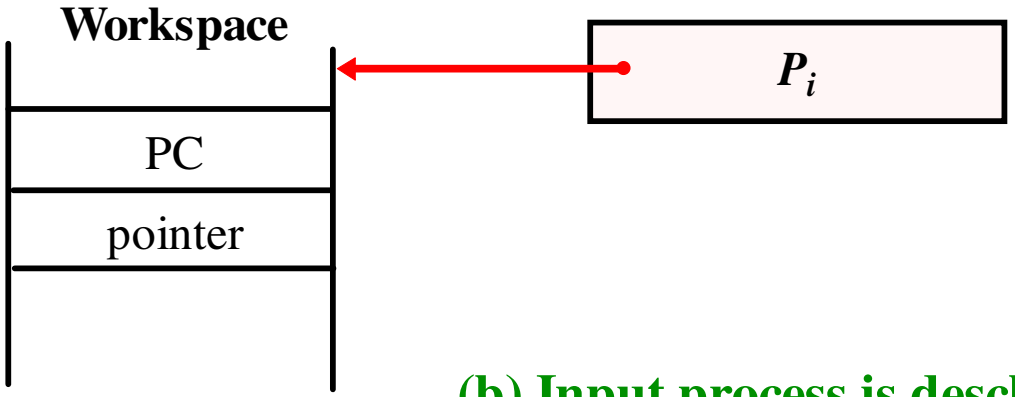


REGISTER & READY LIST STRUCTURE OF TRANSPUTERS

Gyula Fehér feher@novserv.obuda.kando.hu



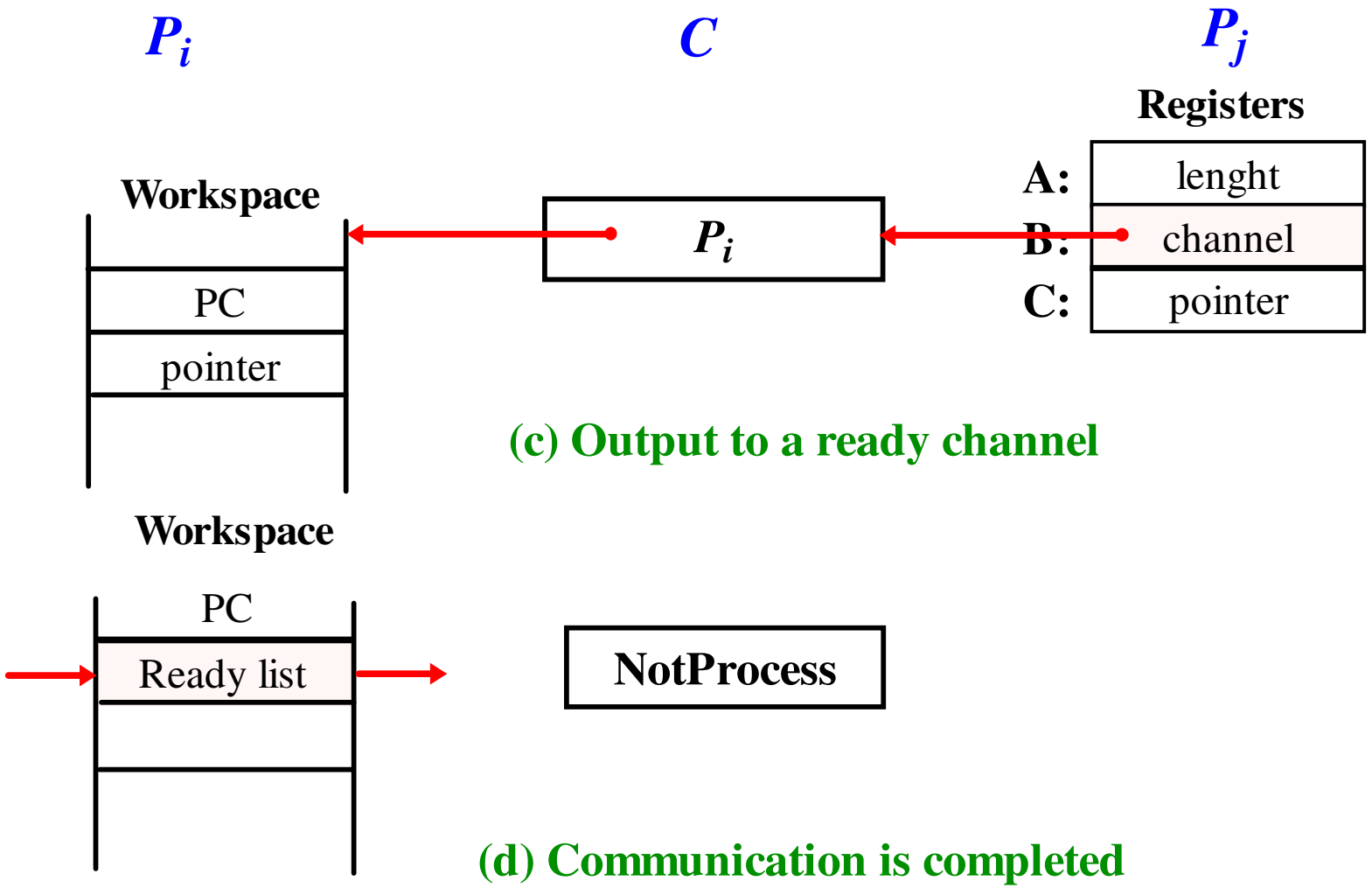
(a) Input on empty channel



(b) Input process is descheduled

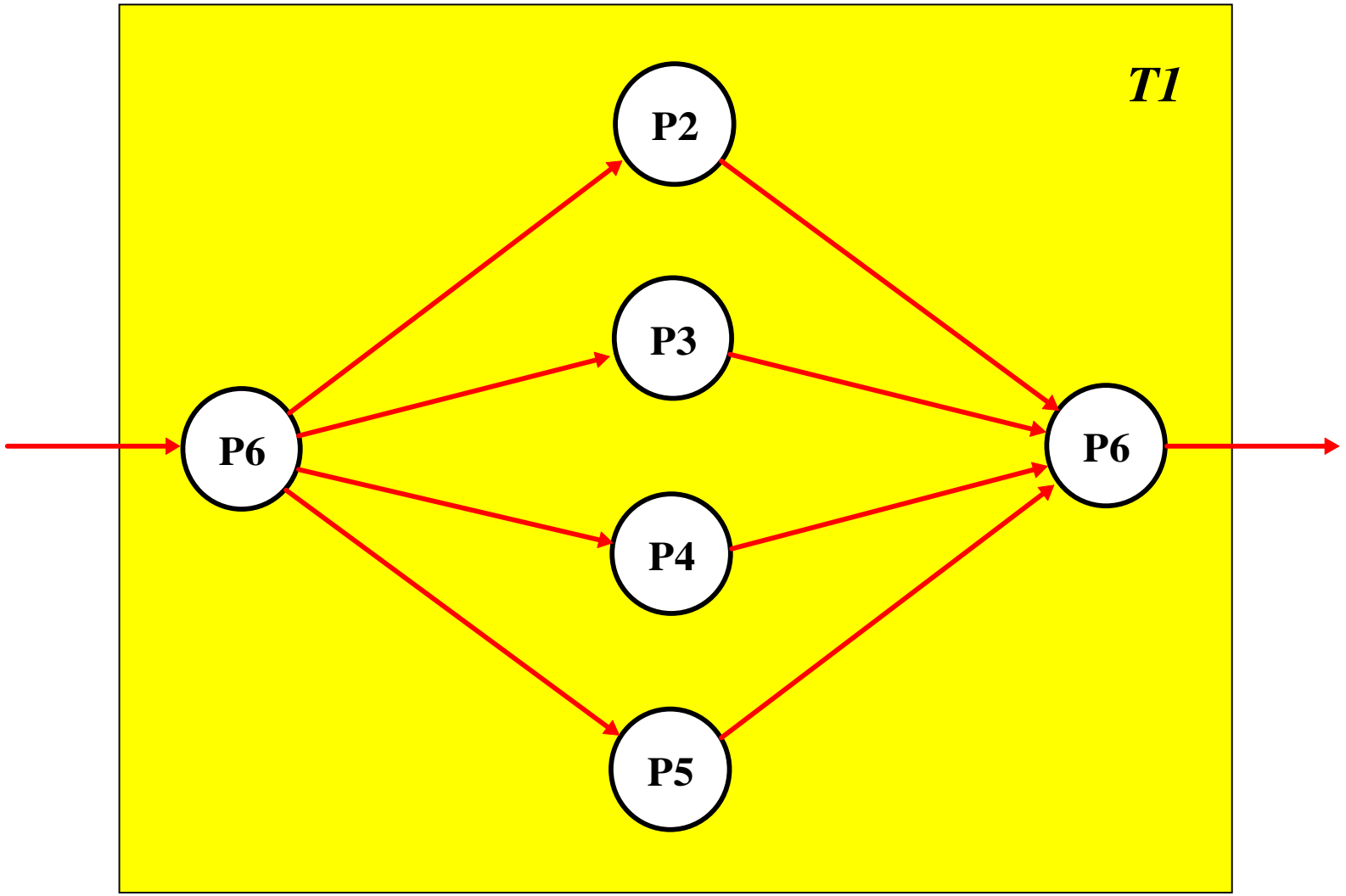
INTERNAL CHANNEL COMMNICATION IN TRANSPUTERS

Gyula Fehér feher@novserv.obuda.kando.hu



INTERNAL CHANNEL COMMNICATION IN TRANSPUTERS

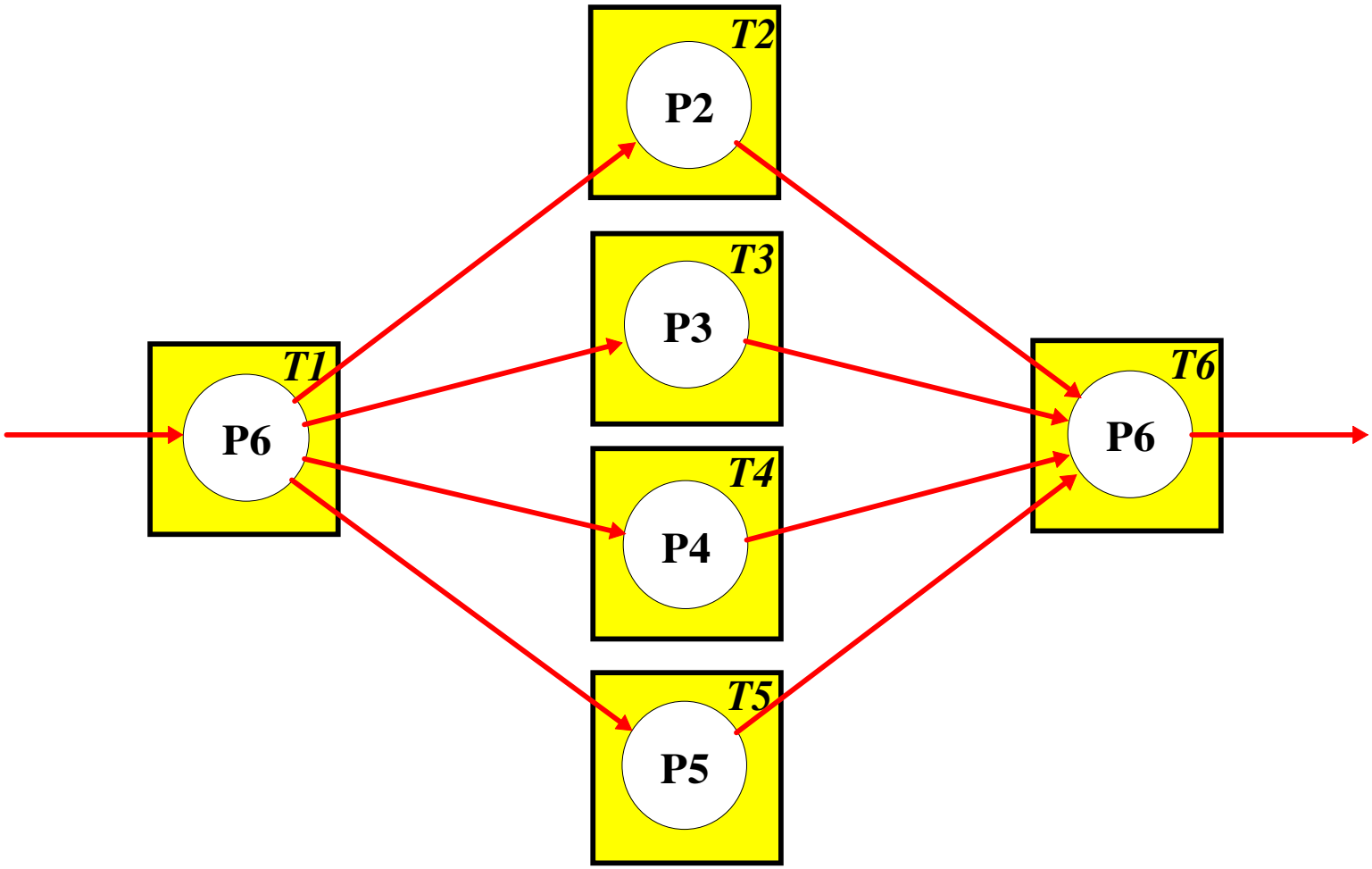
Gyula Fehér feher@novserv.obuda.kando.hu



(a) Processes mapped on 1 transputer

PROBLEM WITH SCHEDULING INVARIANCE

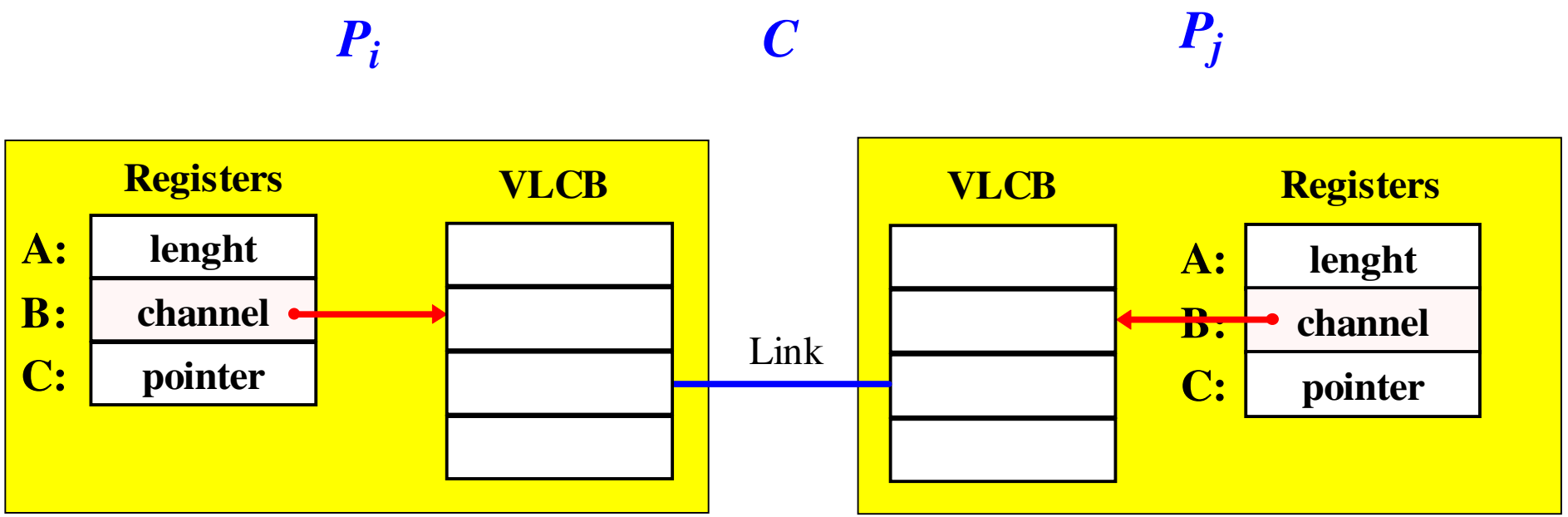
Gyula Fehér feher@novserv.obuda.kando.hu



(a) Processes mapped on 6 transputers
T1 and T6 are illegally require 5 links

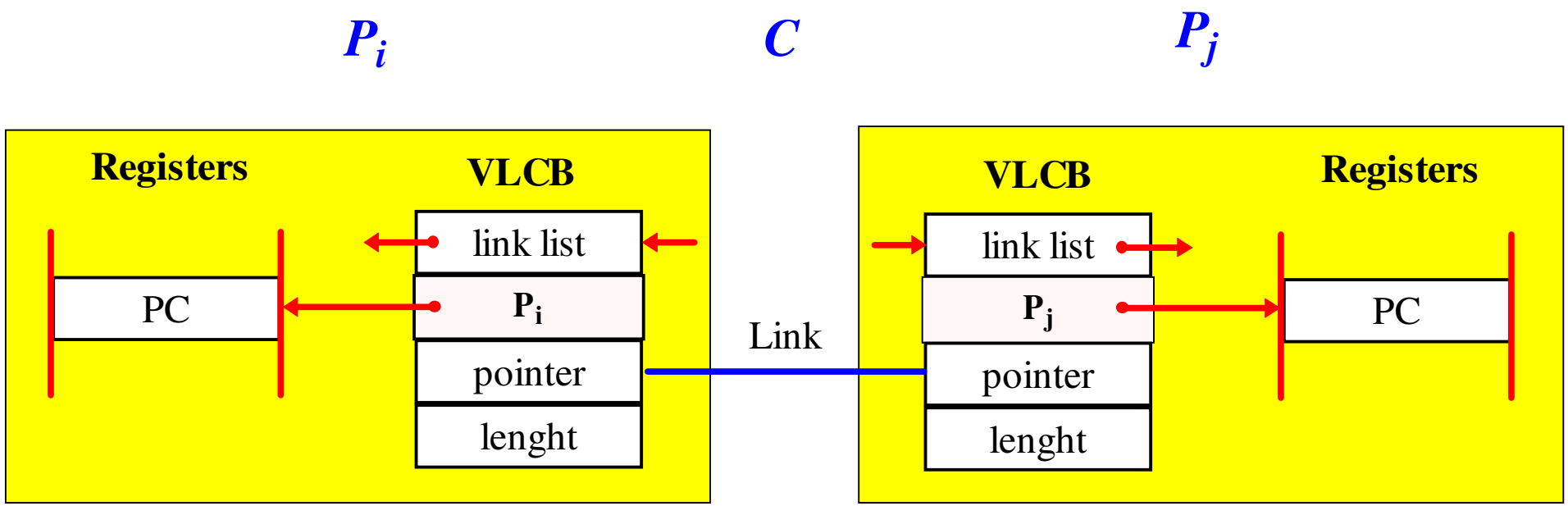
PROBLEM WITH SCHEDULING INVARIANCE

Gyula Fehér feher@novserv.obuda.kando.hu



(a) Preparation of external virtual link communication

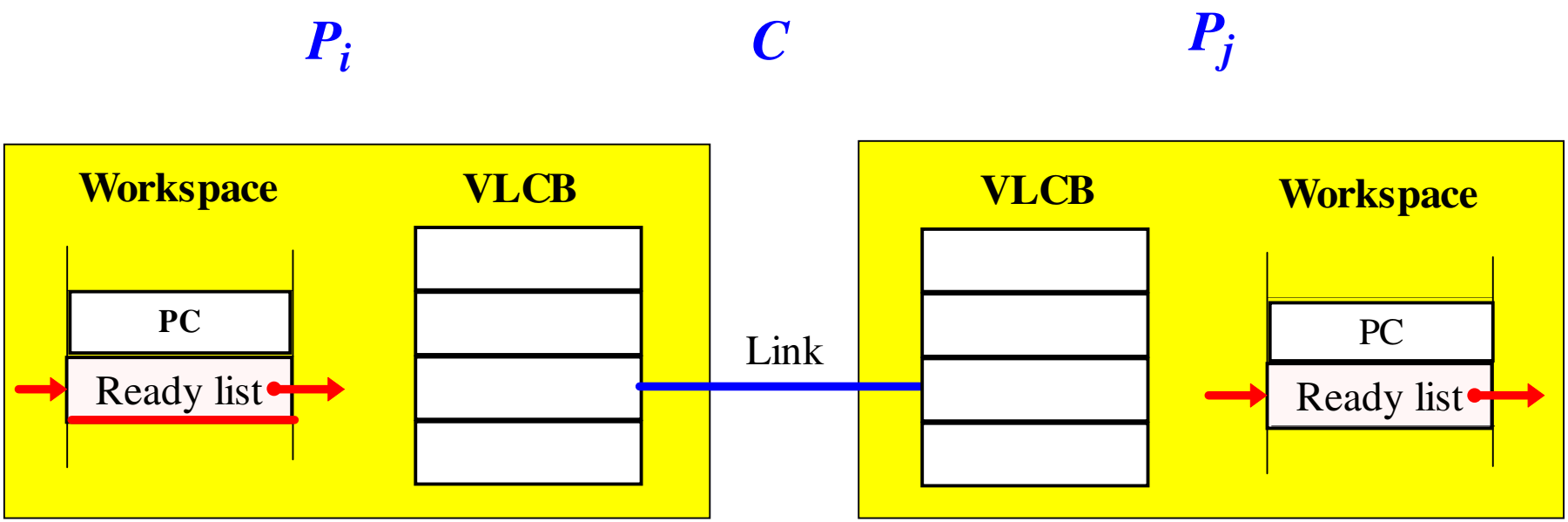
EXTERNAL CHANNEL COMMUNICATION BETWEEN TRANSPUTERS



(b) External virtual communication in progress

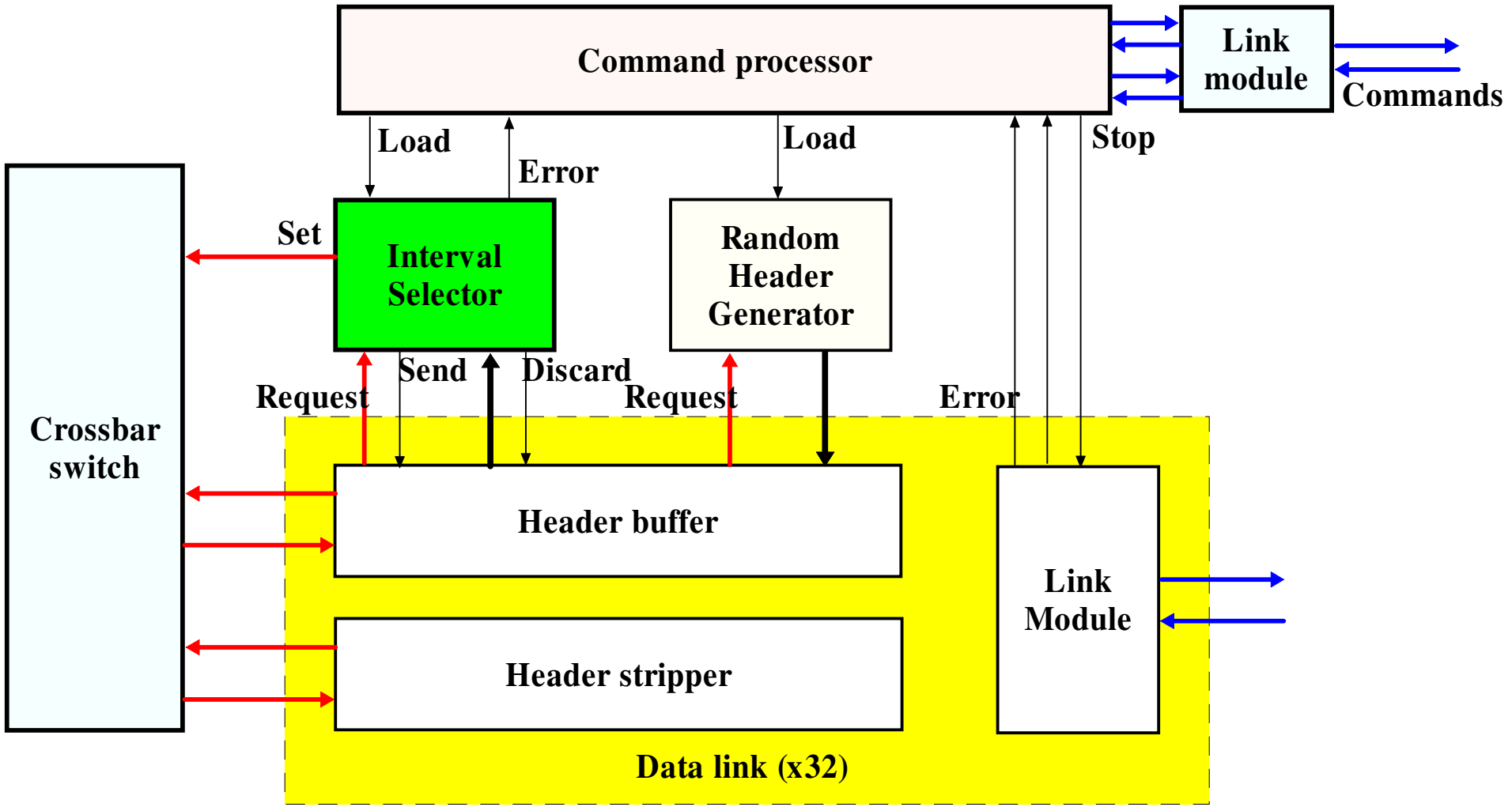
EXTERNAL CHANNEL COMMUNICATION BETWEEN TRANSPUTERS

Gyula Fehér feher@novserv.obuda.kando.hu



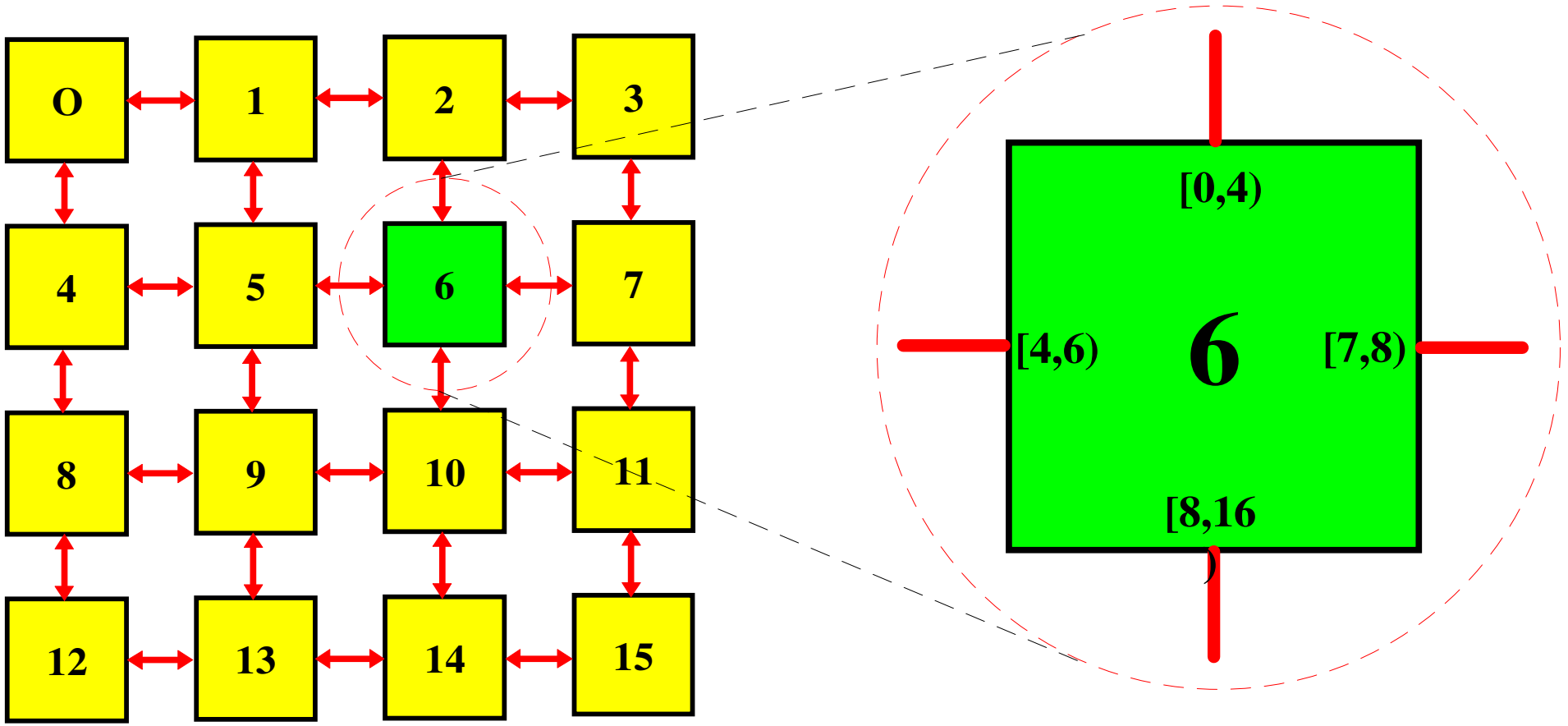
(c) External virtual link communication is completed

EXTERNAL CHANNEL COMMUNICATION BETWEEN TRANSPUTERS



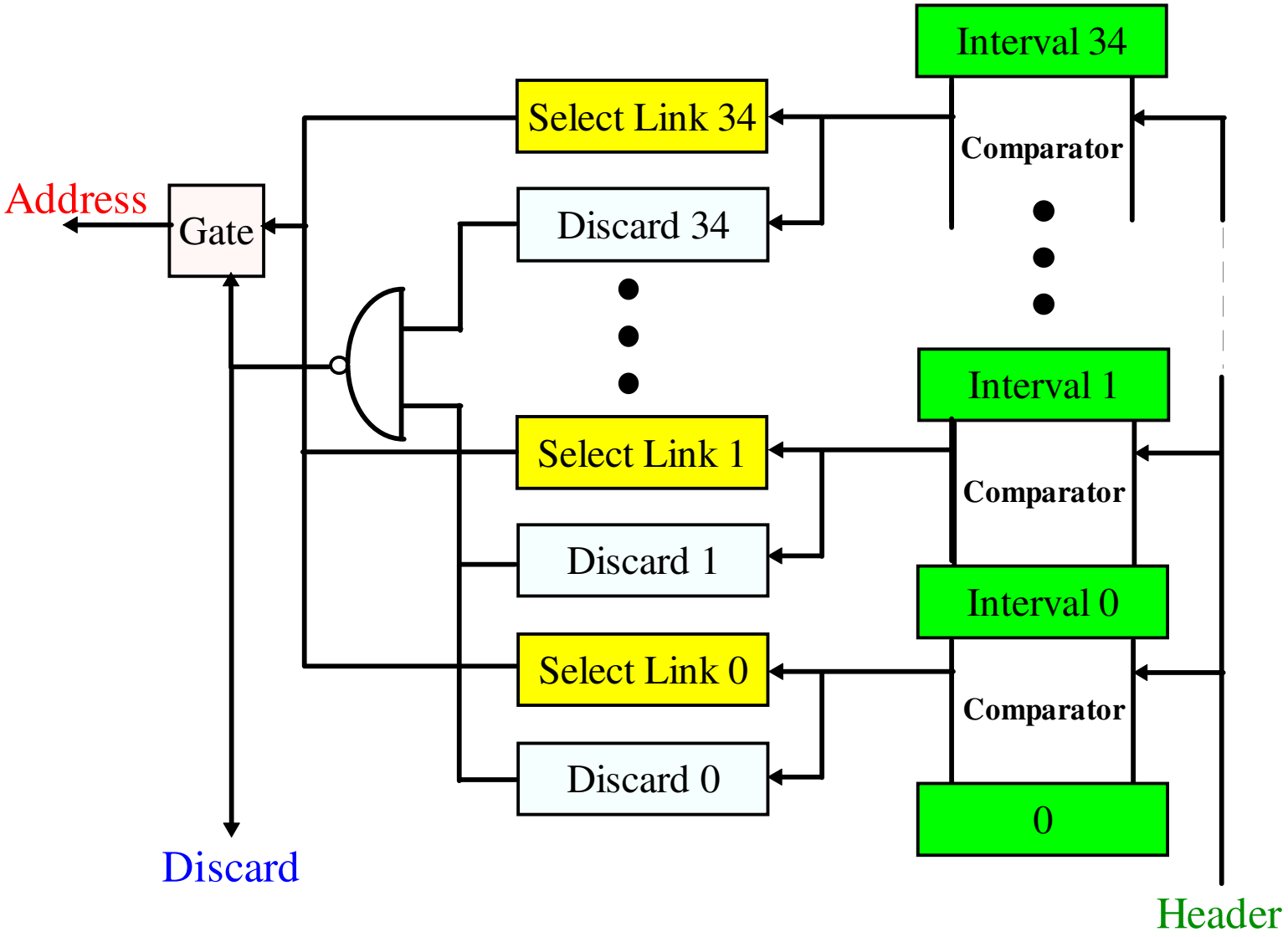
STRUCTURE OF THE C104 PACKET ROUTING SWITCH

Gyula Fehér feher@novserv.obuda.kando.hu



INTERVAL LABELING FOR AN ARRAY

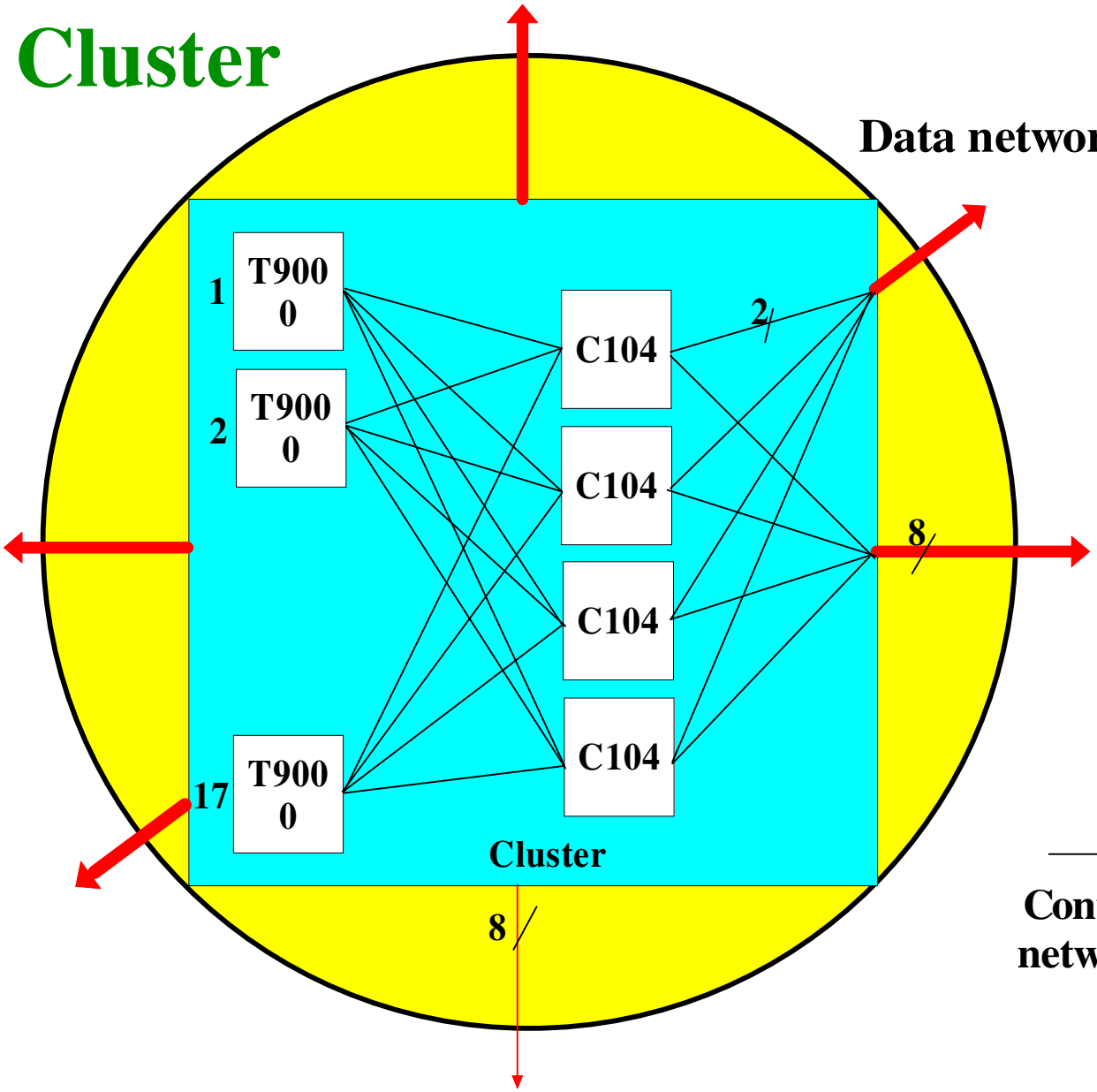
Gyula Fehér feher@novserv.obuda.kando.hu



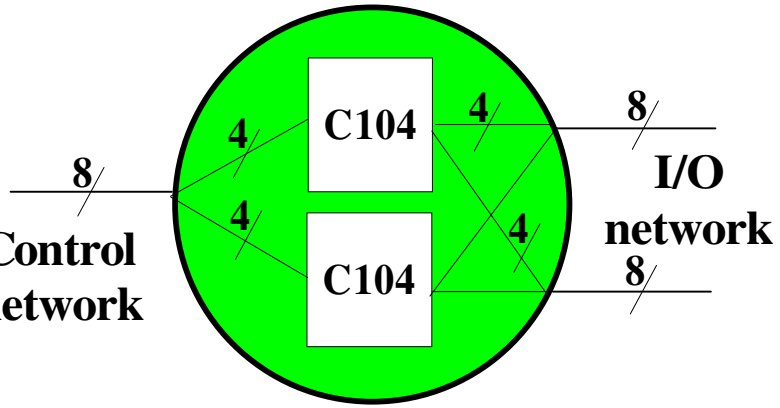
INTERVAL SELECTOR UNIT OF THE C104

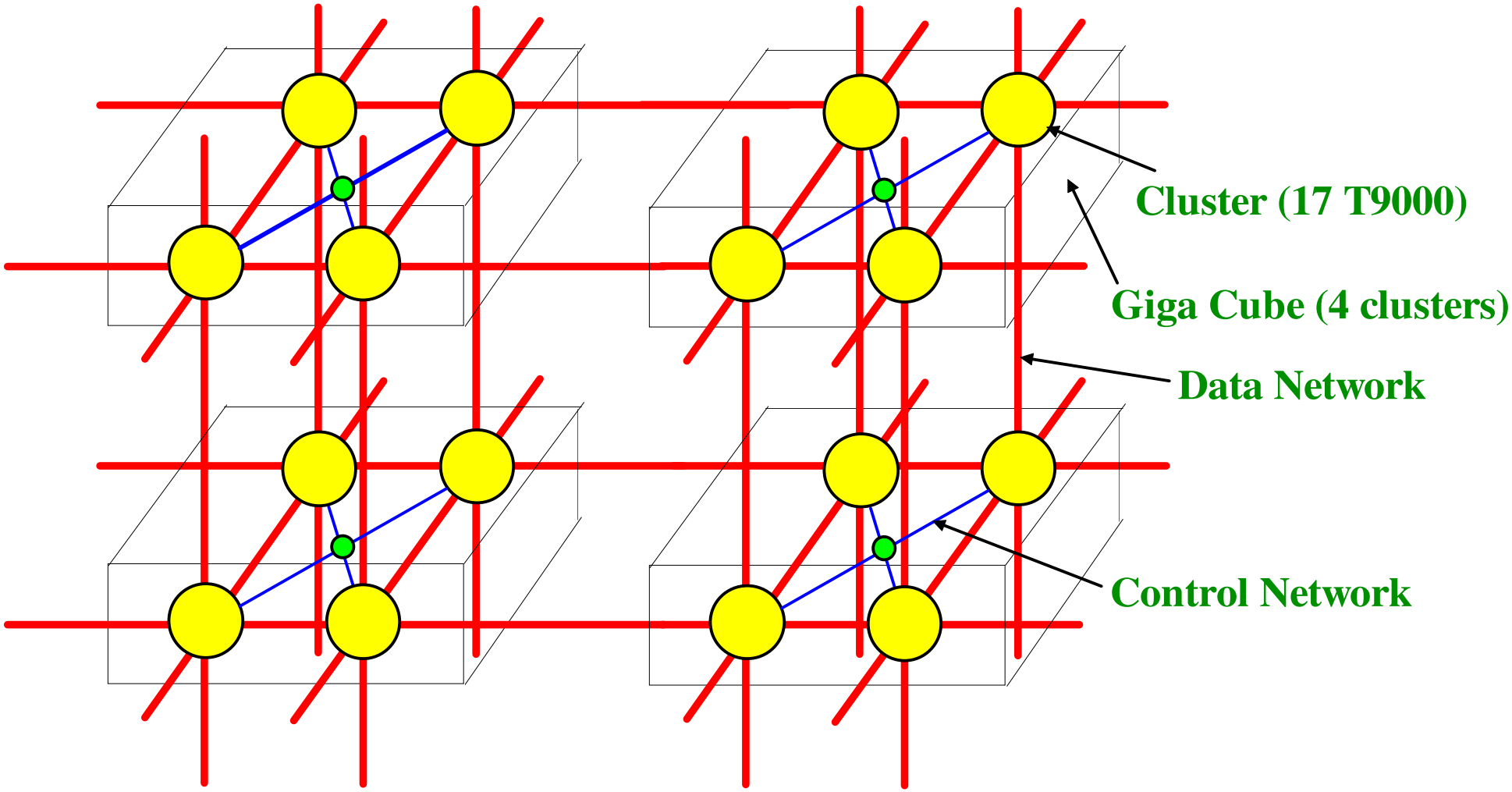
Gyula Fehér feher@novserv.obuda.kando.hu

Cluster



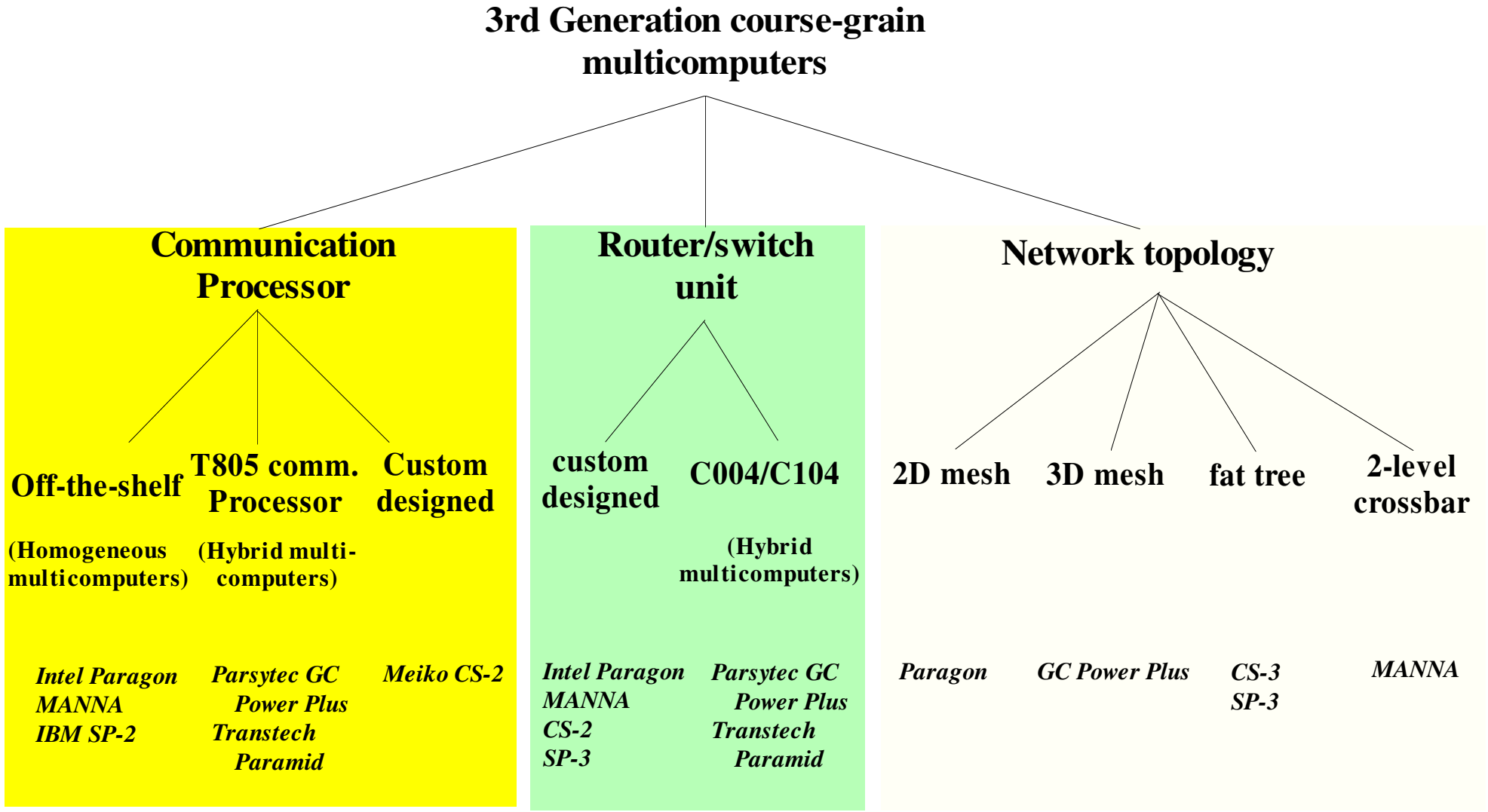
Service node





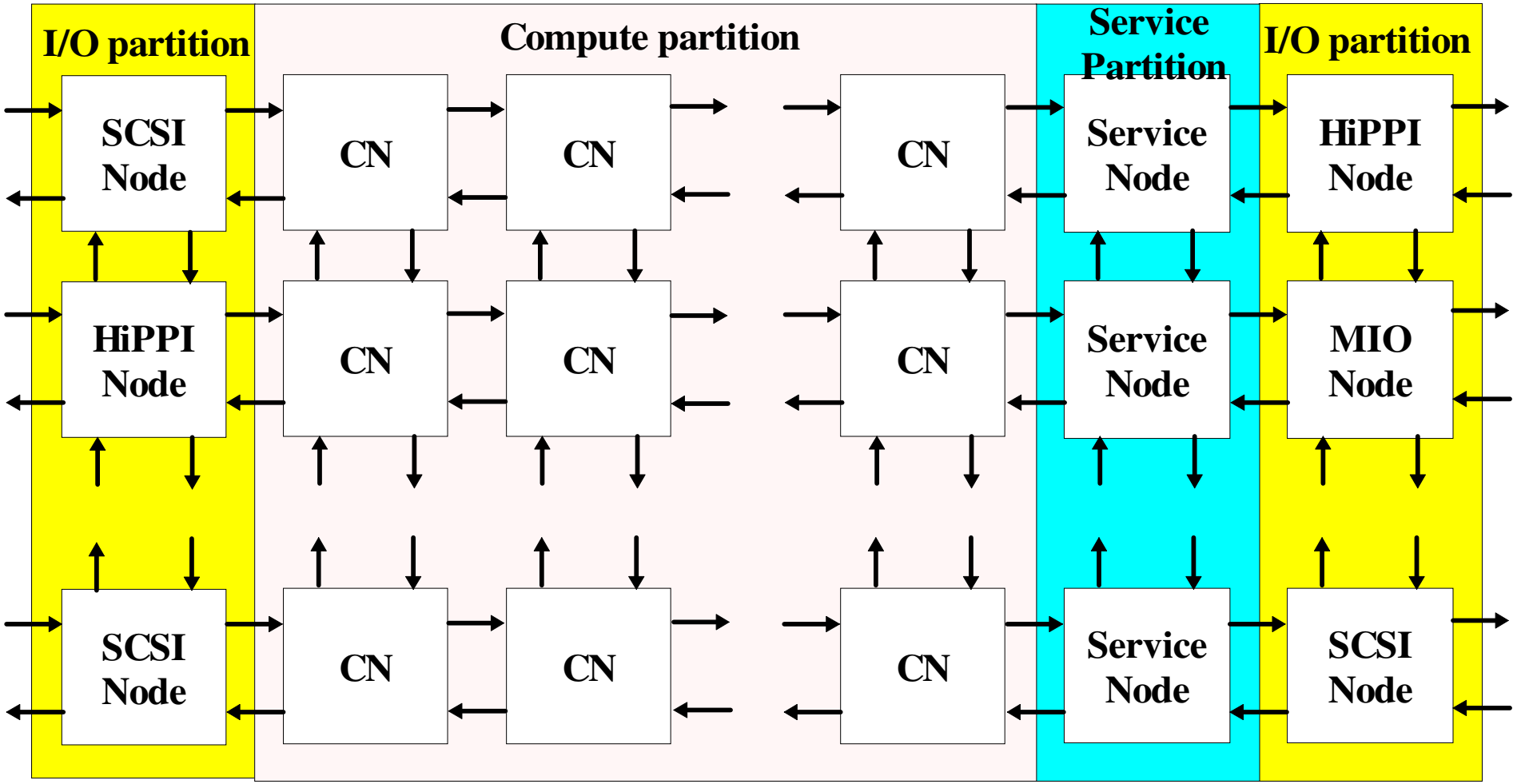
STRUCTURE OF THE PARSYTEC GC MACHINE

Gyula Fehér feher@novserv.obuda.kando.hu



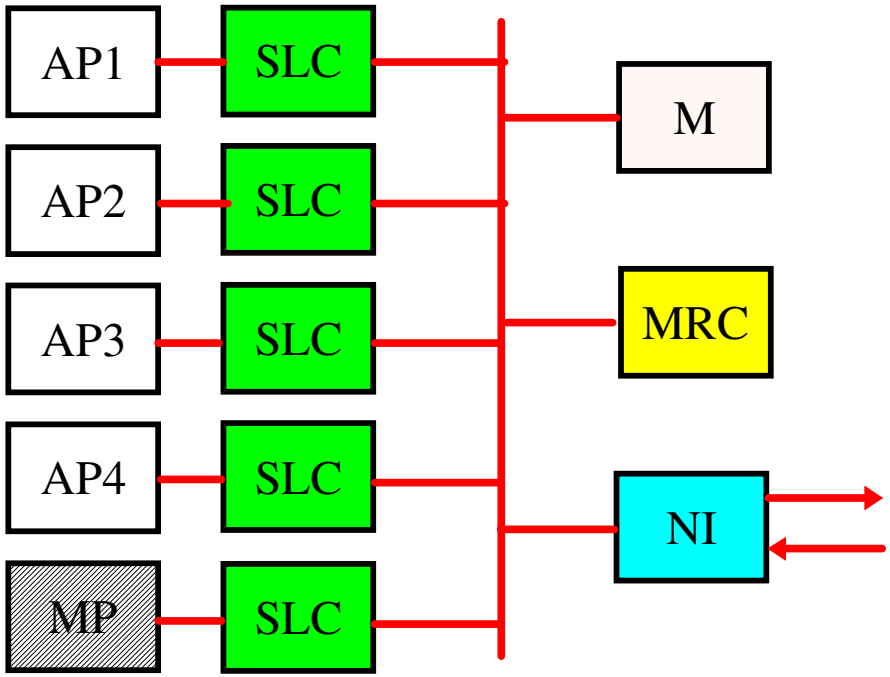
DESIGN SPACE OF 3rd GENERATION COURSE-GRAIN MULTICOMPUTERS

Gyula Fehér feher@novserv.obuda.kando.hu

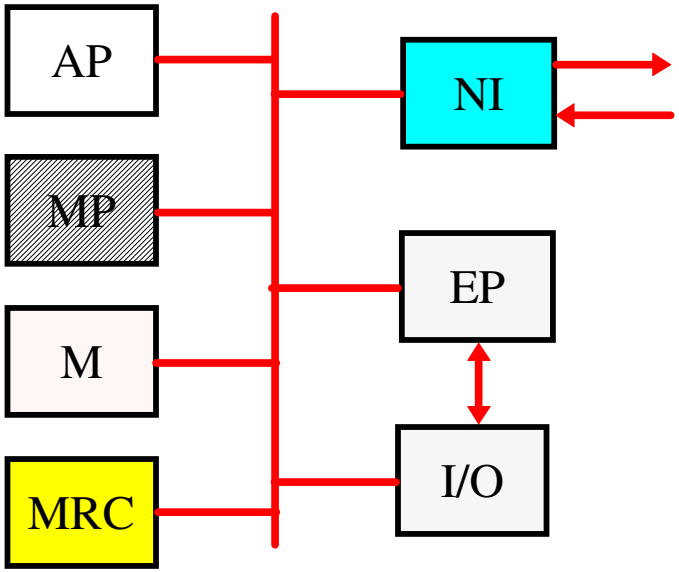


STRUCTURE OF THE PARAGON MACHINE

Gyula Fehér feher@novserv.obuda.kando.hu



(a) Multiprocessor Node



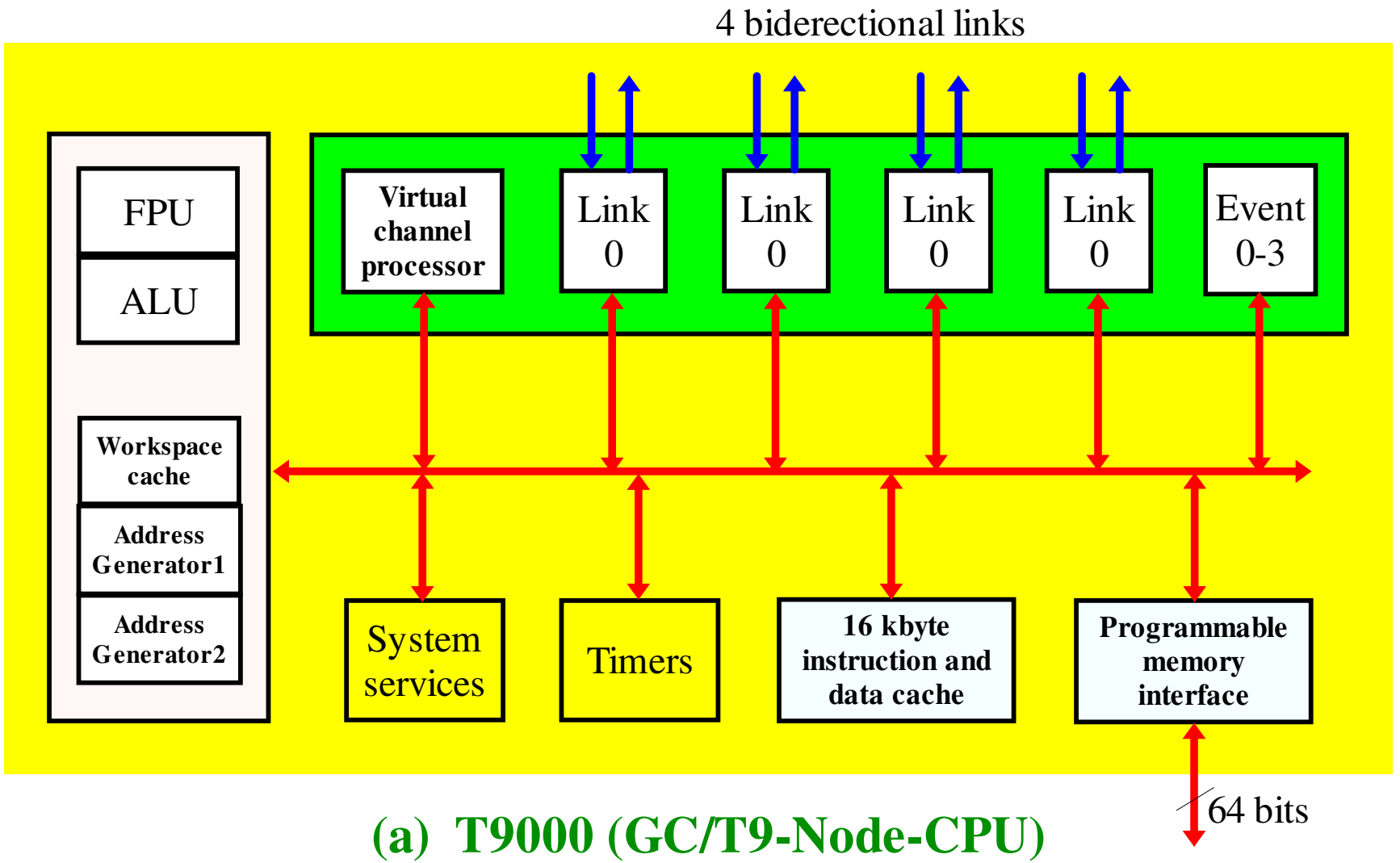
(a) Multiprocessor Node

AP = Application Processor
SLC = Second Level Cache
MP = Message Processor
EP = Expansion Port

M = Memory
MRC = Mesh Router Controller
NI = Network Interface

STRUCTURE OF NODES IN THE PARAGON

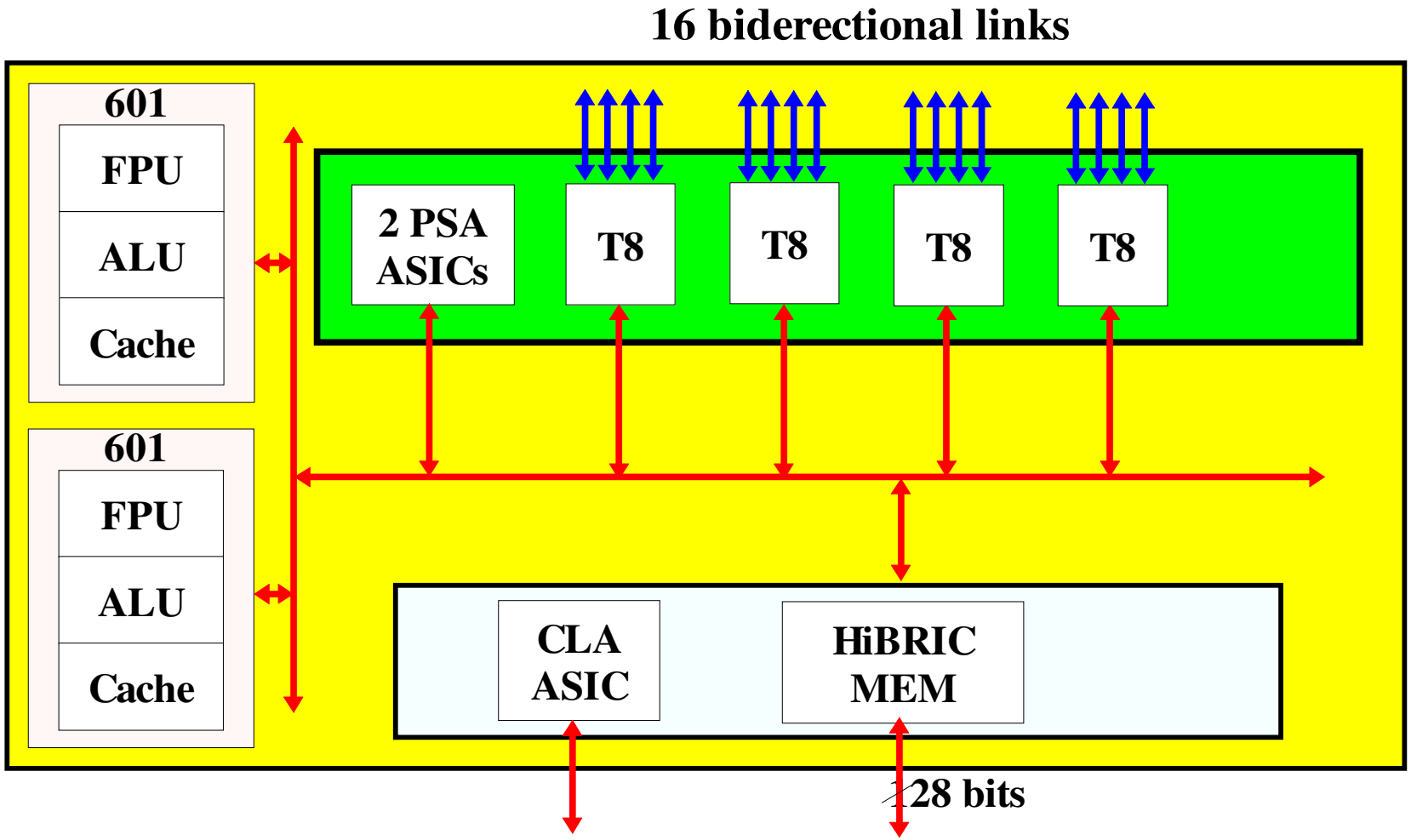
Gyula Fehér feher@novserv.obuda.kando.hu



(a) T9000 (GC/T9-Node-CPU)

COMPARISON OF THE T9000 AND THE GC/PowerPlus-Node-CPU

Gyula Fehér feher@novserv.obuda.kando.hu

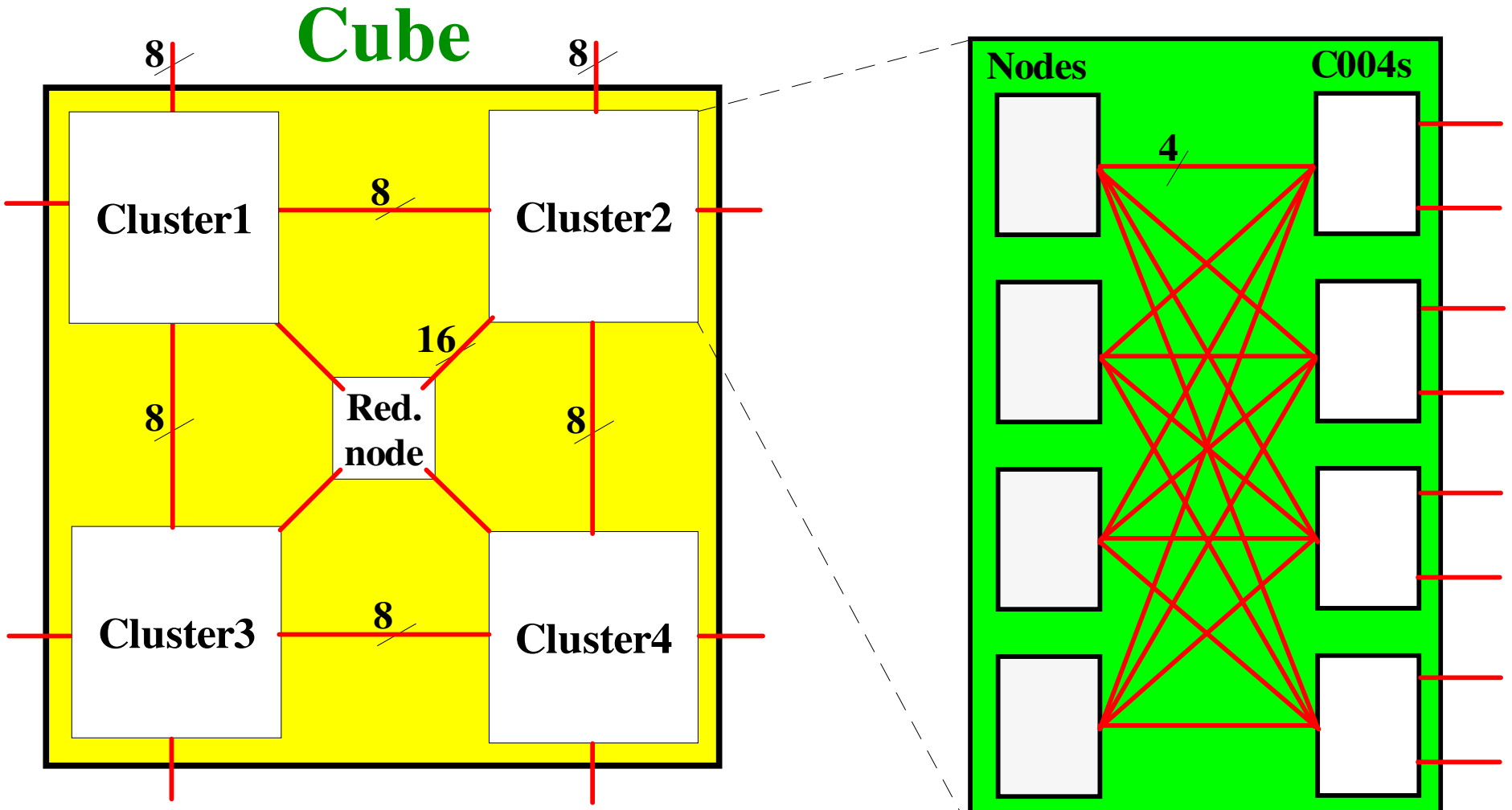


(b) GC/PowerPlus-Node-CPU

COMPARISON OF THE T9000 AND THE GC/PowerPlus-Node-CPU

Gyula Fehér

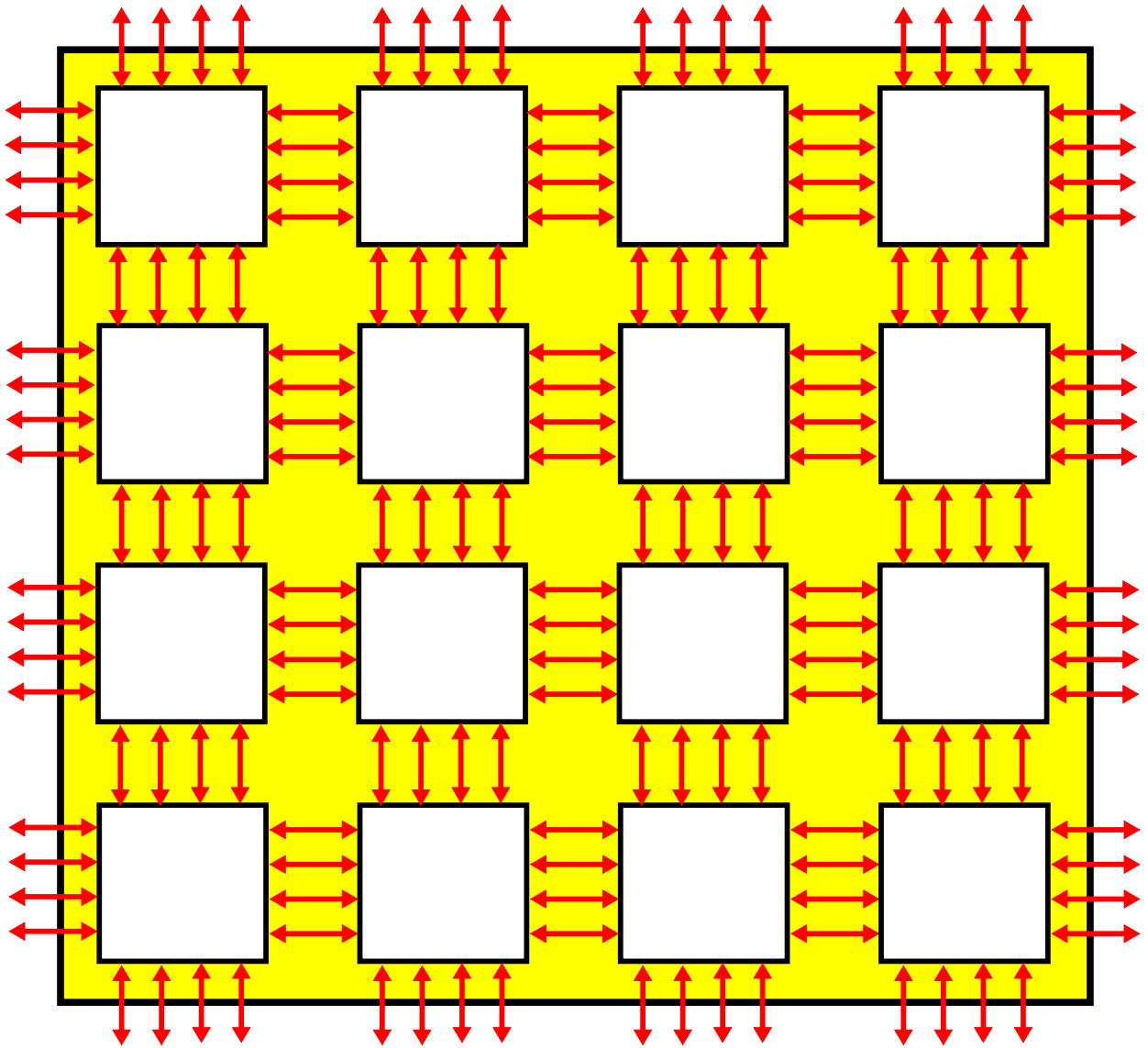
feher@novserv.obuda.kando.hu



INTERCONNECTION NETWORK OF A CUBE IN THE GC/PowerPlus

Gyula Fehér feher@novserv.obuda.kando.hu

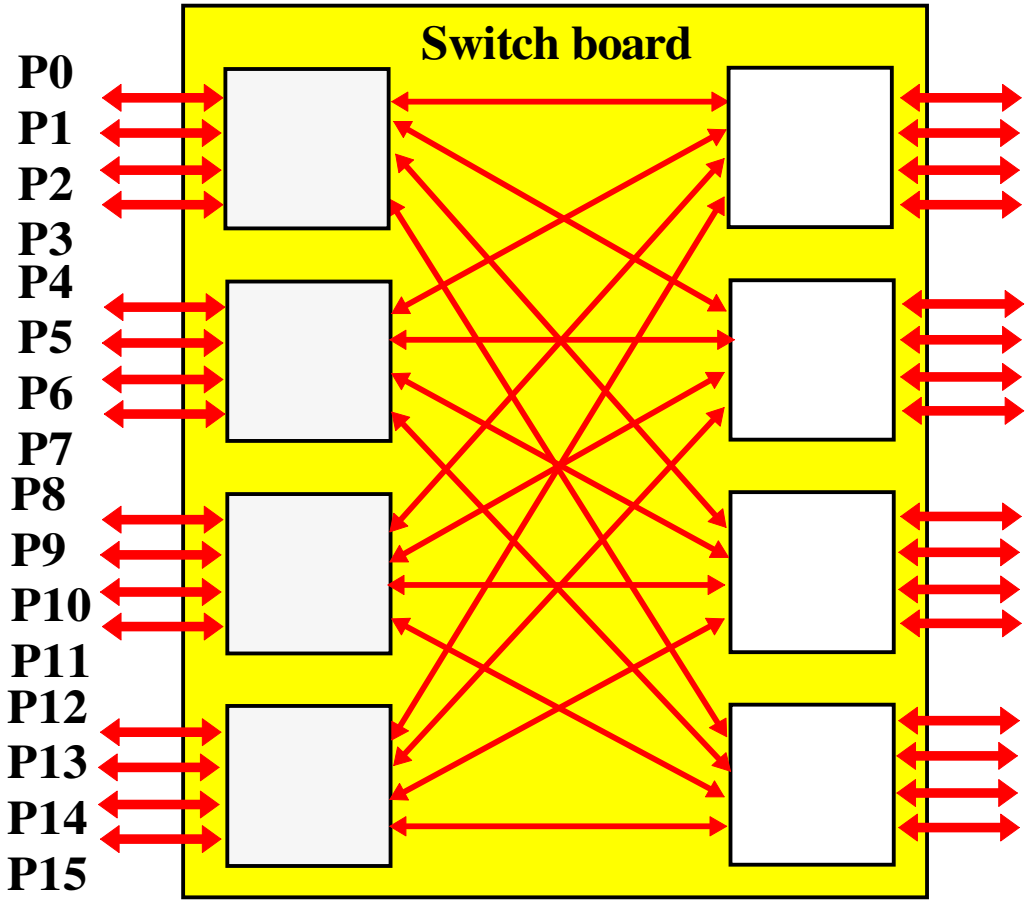
Cube



INTERCONNECTION NETWORK OF A CUBE IN THE GC/PowerPlus

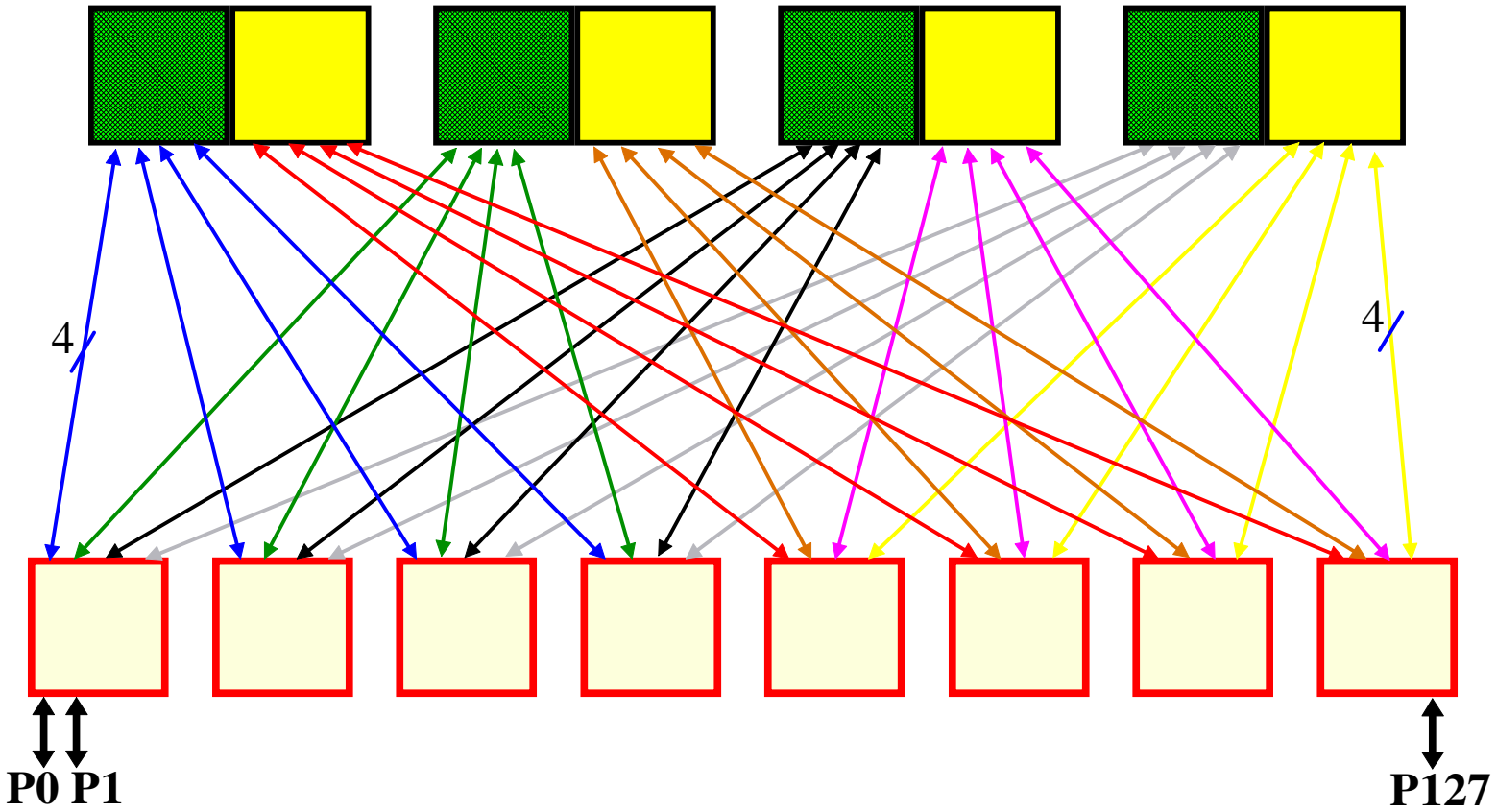
Gyula Fehér

feher@novserv.obuda.kando.hu



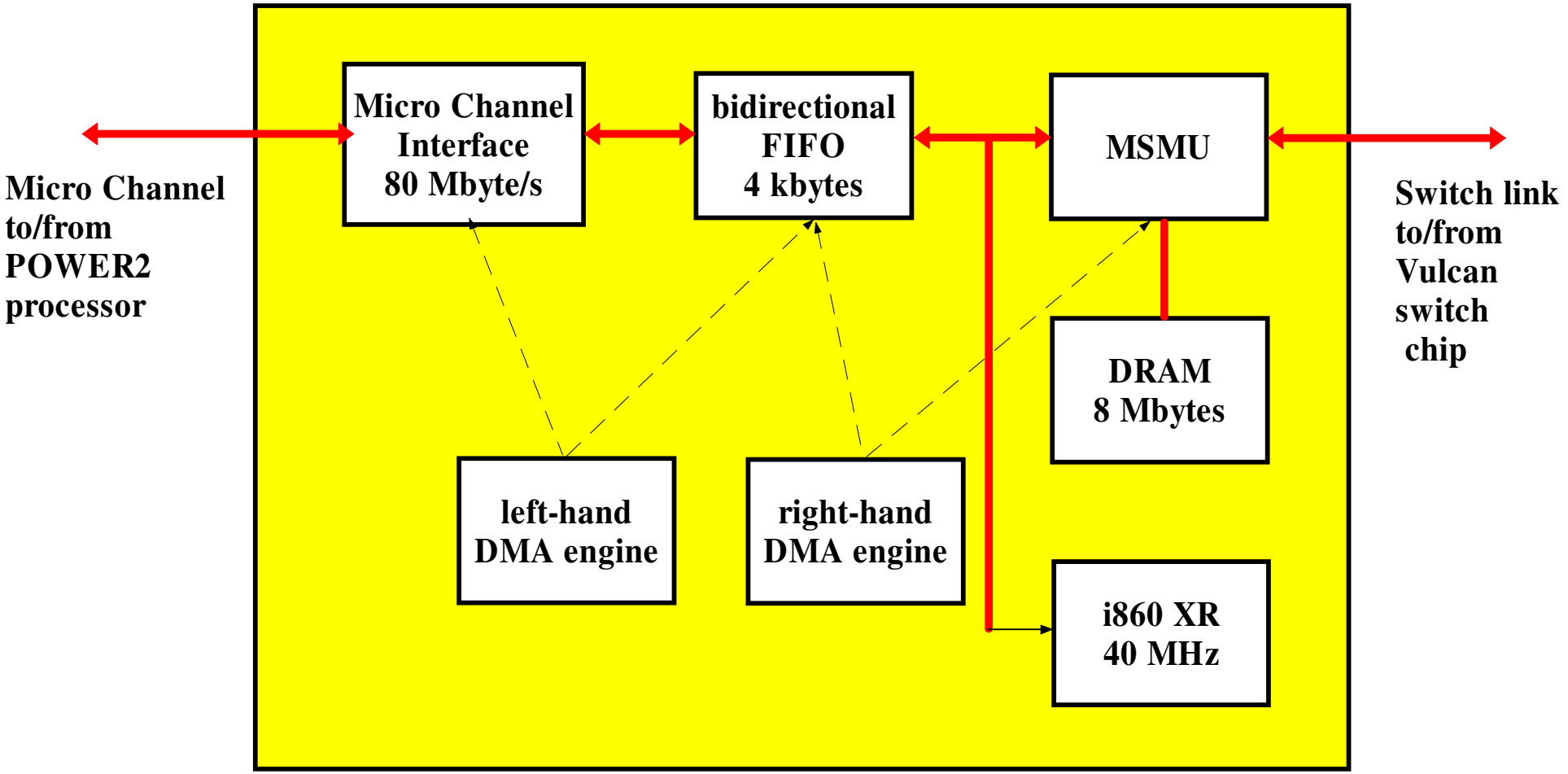
(a) The logical frame of SP2

THE INTERCONNECTION SCHEME OF SP2



(b) Topology of a 128-node SP2

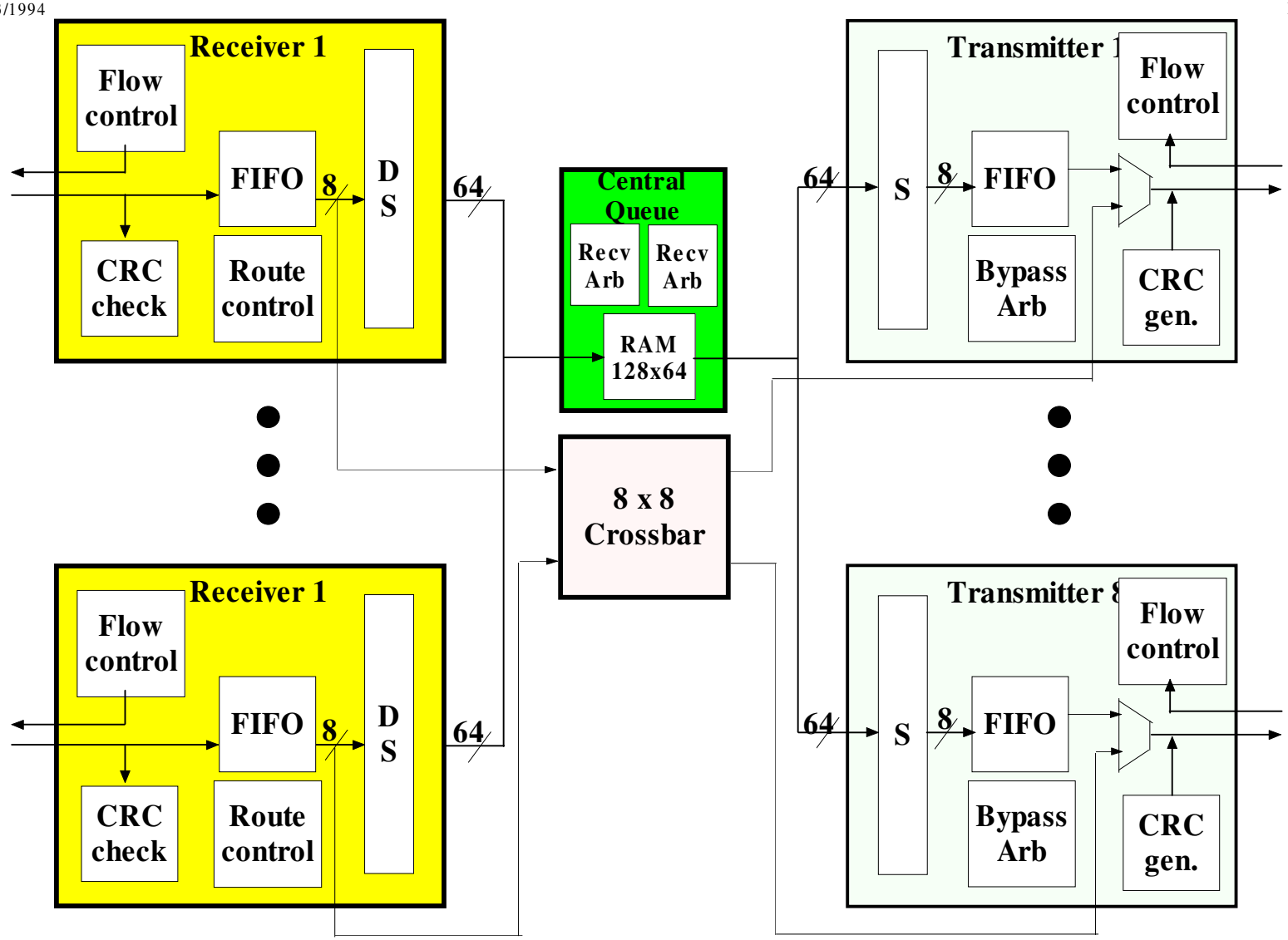
THE INTERCONNECTION SCHEME OF SP2



Structure of the SP2 communication adapter

DIFFERENT COMMUNICATION ADAPTERS

Gyula Fehér feher@novserv.obuda.kando.hu



DS: Deserializer **S: Serializer**

Structure of the Vulcan switch unit

Gyula Fehér

feher@novserv.obuda.kando.hu

Machine	*Computing Processor*	*Communi-cation Processor*	*Vector Processor*	*Communi-cation Switch Unit*	*Network Topology*
nCUBE 2	custom designed	-	-	custom designed	hypercube
iPSC-2	i386	-	yes	yes	hypercube
Intel Paragon	i860 (1 or 4)	i860	-	custom designed	2D mesh
GENESIS	i870	i870	custom designed	-	2-level crossbar
MANNA	i860	i860	-	16x16 crossbar	hierarchical crossbar
Parsytec GC/PowerPlus	PowerPC601 (2)	T805 (4)	-	C004	3D mesh
Transtech Paramid	i860	T805	-	C004	variable
IBM SP-2	POWER2	i860	-	custom designed	fat tree
Meiko CS-2	SPARC	custom designed	Fujitsu uVP (2)	custom designed	fat tree
Parsys SN9800	T9000	T9000	-	C104	hierarchical switch
J-Machine	custom designed (MDP)	MDP	-	MDP	3D mesh

	1st generation	2nd generation	3rd generation
Topology	hypercube	hypercube/mesh	2-D and 3-D mesh
Switching technique	packet (store and forward)	circuit switched or wormhole	wormhole
Comm. latency of 100 byte message Neighbour (μsec) Remote (μsec)	 2000 6000	 5 5	 0.5 0.5
Node organisation	single computation processor	computation processor + router	computation proc. + communication proc. + router
Typical node number	64	256	1024
Node memory (Mbytes)	0.5	4	32
MIPS	1	10	100
Mflops scalar	0.1	2	40
Mflops vector	10	40	200
System memory (Mbytes)	32	1K	32K
System MIPS	64	2560	100K
System Mflops scalar	64	512	40K
System Mflops vector	640	10K	200K

Comparison of three generations of multicomputers

Gyula Fehér feher@novserv.obuda.kando.hu

TECHNOLOGY SCALING TRACKS OF MULTICOMPUTERS

- Multicomputers can usefully be implemented with nodes that **range in size** today from
-

  1. a part of a single chip to a
  2. *large circuit board containing hundreds of chips.*

- The realationship between *node size* and *complexity* evolves with advances in microelectronic technology.
- In order to appreciate some of the rationale and motivation for the development of multicomputers, it is helpful to understand the implications of these technology-scaling tracks.

Reference: The *Cosmic Cube* (1981) was the origin of a different scaling track that has been followed by different developers.

Constant-node-complexity scaling track

Constant-node-cost scaling track

Constantly-increased-node-cost scaling track

Constant-node-complexity scaling track

Mosaic C (1992) is an example of this scaling track in which advances in technology are employed to re-implement nodes of similar logical complexity, but faster, smaller, lower power, and less expensive.

From *Cosmic Cube* to *Mosaic C*. Mosaic C is

60 times faster, use
20 times less power, are
100 times smaller and are
25 times less expensive to manufacture.

Mosaic C nodes exhibit approximately 60x25= **1500 times higher performance/cost** than the *Cosmic Cube* nodes.

Motivations for exploring the domain of fine-grain multicomputers

- they provide sustantionaly higher performance/cost than medium-grain multicomputers, and
- this advantage will continue to grow with future advances in microelectronics.
- Fine-grain multicomputers can deliver this cost-effective computing in relatively small, low-cost configurations for embedded-system applications, eg., *they are not aimed exclusively at the "Terops race".*

Constant-node-cost scaling track

In this scaling track advances in technology are employed using to scale node performance and memory size.

Examples:

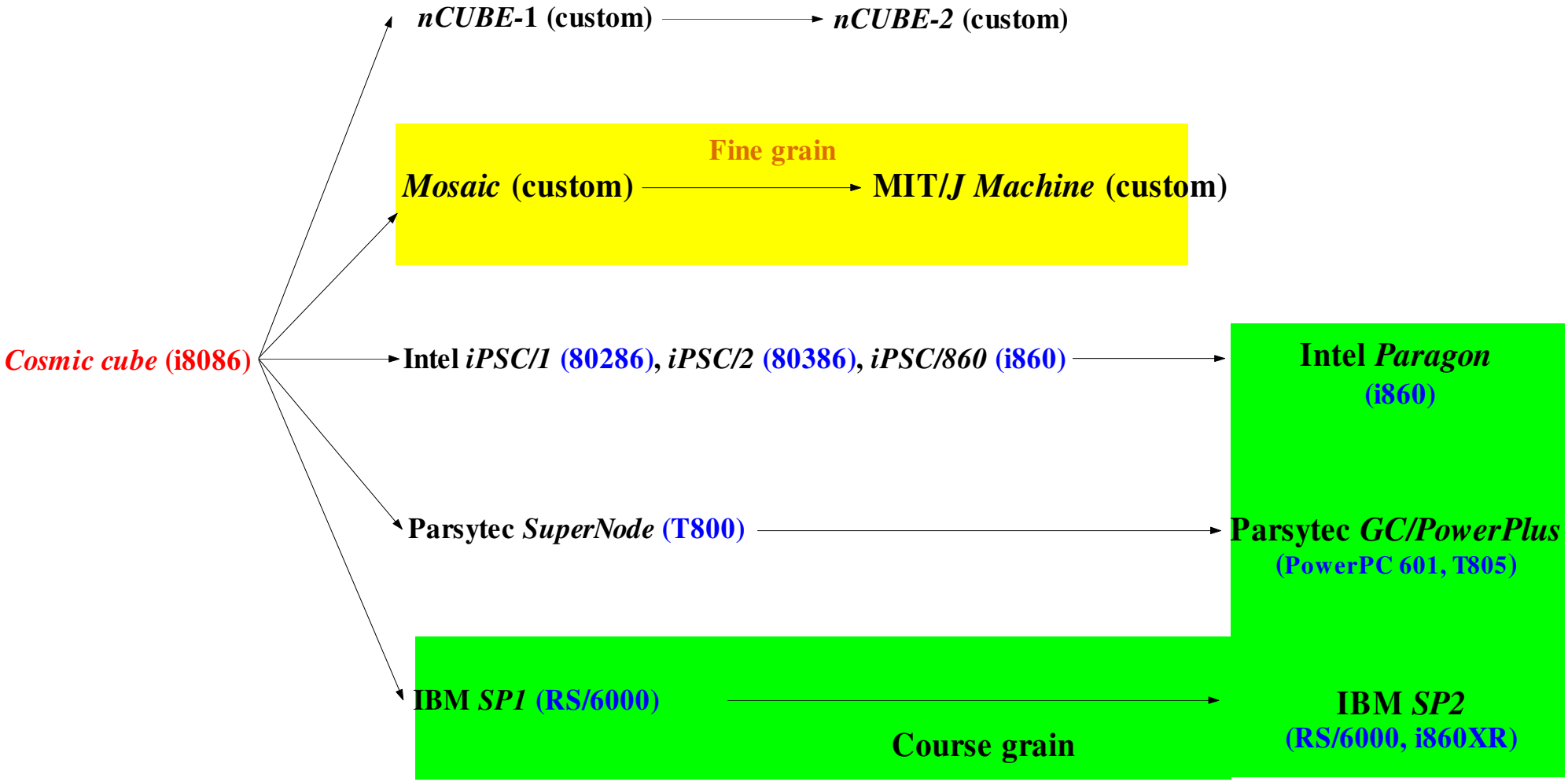
- From *Mosaic C* to *Mosaic T*

Other example of commercial multicomputer is *N-CUBE*, that would lie closer to a contant-node-cost scaling track from the *Cosmic Cube*.

Constantly-increased-node-cost scaling track

The scaling track of the commercial, medium-grain multicomputers that, for example , Intel has followed has been using advances in technology to scale node performance and memory size. However, Intel has also been slowly but steadily increasing the cost of the nodes.

Advantages the nodes, because they have been similar in complexity to contemporary personal and workstation computers, have been able to employ the same commodity processor, memory, and software technology.
The disadvantage is that the performance/cost scales much more slowly than contant-node-complexity scaling. For example, the Intel Paragon has only about 100 times higher performance/cost than the 1983 Cosmic Cube.



ARCHITECTURAL DEVELOPMENT TRACKS OF MESSAGE PASSING MULTICOMPUTERS

Gyula Fehér feher@novserv.obuda.kando.hu