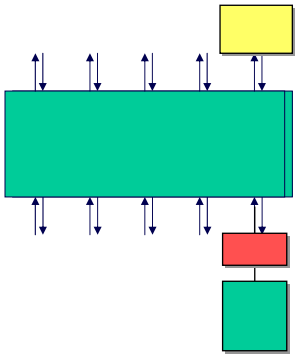
Index

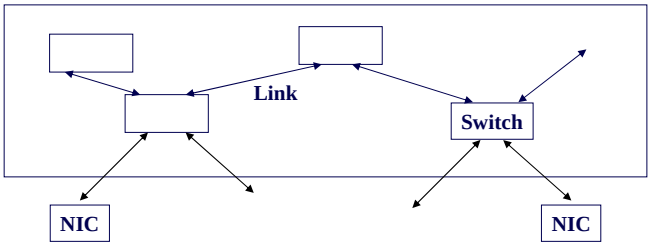
Interconnection networks

- **Need to send messages (commands/responses, message passing)**
  - Processors ↔ Memory
  - Node ↔ Node



Interconnection networks

- **Components**
  - Links
  - Switches
  - Network interface

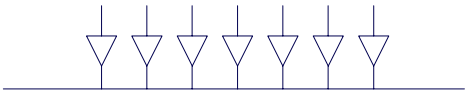


Links

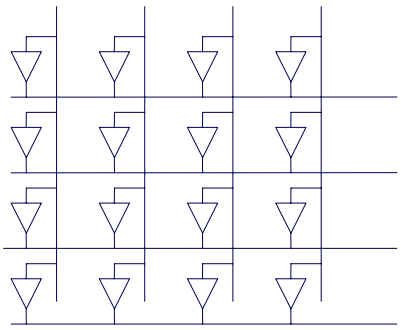
- **Shared (busses) / point to point**
  - Speed ➔ point to point
- **Uni / bidirectional**
  - Faster: 2 unidirectional
- **Width**
  - 1bit / N bits
    - ✓ Skew
- **Length**
  - Transmission line
- **Bandwidth**
  - Bits/s
- **Weight, layout, space,…**

Switches

- **Switching structures**
  - Bus

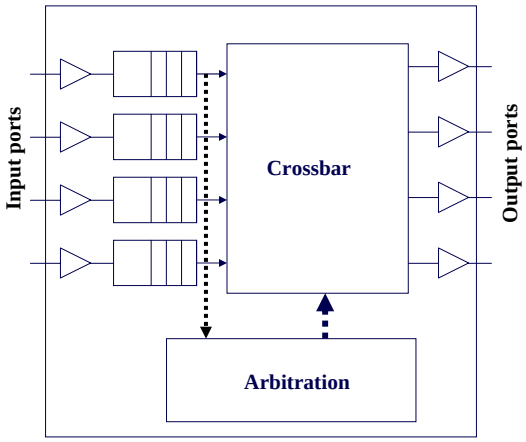


  - Crossbar



Switches

- **Buffering**
  - At input
  - At output
  - Both
- **Arbitration**
  - Fixed, Random,…
  - Head of line blocking



Interconnection networks

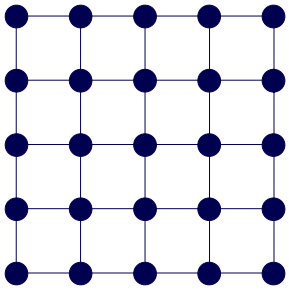
- **Issues**
  - Topology
    - ✓ Physical interconnection structure of the network graph
  - Routing algorithm
    - ✓ Determine the path followed by a message
  - Switching strategy
    - ✓ How the data in a message traverses the components along the path
  - Flow control
    - ✓ When does data progress along the path

Topology

- **Types**
  - Direct
    - ✓ 1 node per switch
  - Indirect
    - ✓ Internal switches without nodes connected
- **Properties**
  - Distance between two nodes
  - Average distance
  - Diameter
  - Bisection

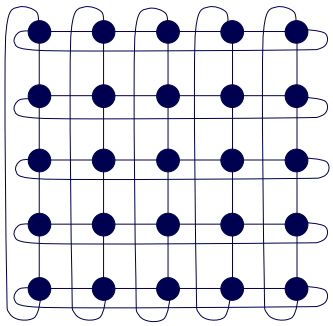
Topology

- **Grid**
  - 1D: Linear array
  - 2, 3D



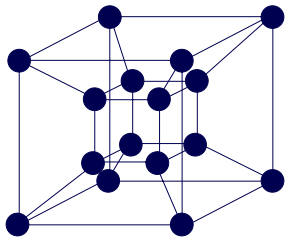
Topology

- **Torus**
  - 1D: ring



Topology

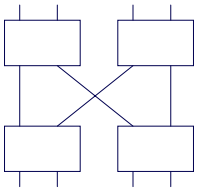
- **Hypercube**



Topology

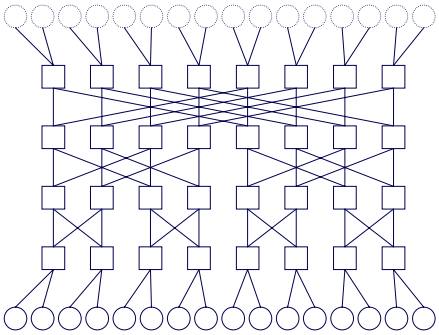
- **Indirect networks**
  - Basic component: N x N Crossbar switch

  - Multistage



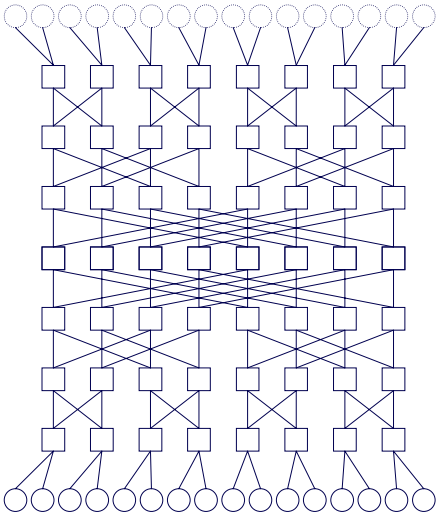
Topology

- **Butterfly**



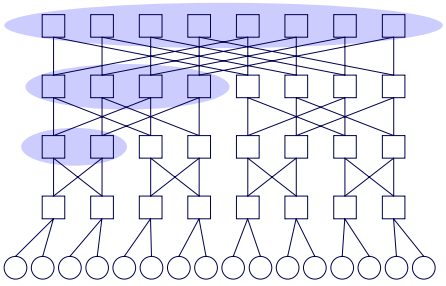
Topology

- **Benes**
  - Conflict free routing of permutations
  - Random intermediate destination



Topology

- **Fat tree**
  - Random destination at least common ancestor fat node

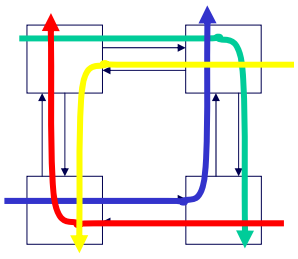


Routing

- **Approaches**
  - Arithmetic
    - ✓ Local at each node, based on destination and regular topology
    - ✓ Dimension order routing
      - – One dimension at a time, progress towards the target coordinate in that dimension
  - Table lookup
  - Source based routing
    - ✓ Source node computes route and emits into the header the set of output links to be used by successive switches
- **Other alternatives**
  - Static / Adaptive

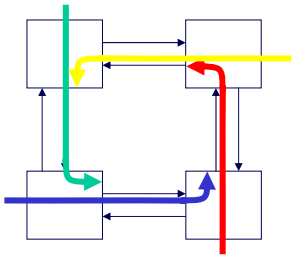
Routing

- **Deadlock**
  - Possibility to have all messages waiting for a resource already used by other message.
  - Model: channel dependence graph
    - ✓ Nodes: resources (Incoming buffer at switches)
    - ✓ Directed arcs: Possibility for message to go from one to other
    - ✓ Deadlock free: no cycles
- **Examples**
  - Deadlock free XY routing
  - Possibility in torus



Routing

- **Deadlock**
  - Possibility to have all messages waiting for a resource already used by other message.
  - Model: channel dependence graph
    - ✓ Nodes: resources (Incoming buffer at switches)
    - ✓ Directed arcs: Possibility for message to go from one to other
    - ✓ Deadlock free: no cycles
- **Examples**
  - Deadlock free XY routing
  - Possibility in torus



Routing

- **Deadlock avoidance**
  - Restrict possible paths
    - ✓ Up*-Down* routing
      - – Spanning tree of graph
      - – Route: some steps up, then some down
    - ✓ Turn model routing
      - – Minimum set of forbidden turns?
        West-first, North-last, Negative-first
  - Restrict resource allocation
    - ✓ Virtual channels: Partition resources such that each physical channel represents several virtual channels
      - – Ex: 1 channel for messages going to lower numbered nodes and one channel for messages to higher numbered nodes

Switching strategy

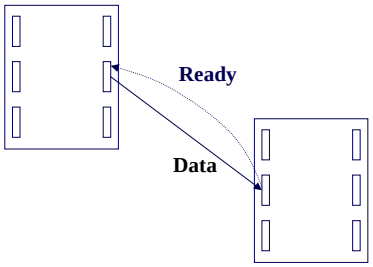
- **Types**
  - Circuit switching
    - ✓ Set up connection between source and destination
    - ✓ Fully dedicated resources
  - Packet switching
    - ✓ Packets handled individually
    - ✓ Only use resources as they progress from source to destination
- **Packet**
  - Header
    - ✓ Routing and control information
  - Payload
  - Trailer
    - ✓ ECC

Switching strategy

- **Packet switching**
  - Store and forward
    - ✓ Full packet buffered at each node before retransmitting
  - Cut through
    - ✓ Virtual cut through
      - – After processing header, message retransmission starts.
      - – If not possible ➔ buffer incoming message
    - ✓ Wormhole
      - – If not possible to retransmit, block message along the path

Flow control

- **Control progress of data through the network limited by resource availability (storage to receive incoming data)**
- **Link level flow control**
  - Flit
    - ✓ Minimum unit
  - Back pressure
- **End to end flow control**
  - End to end acks



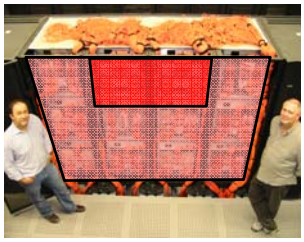
Cost – Performance issues

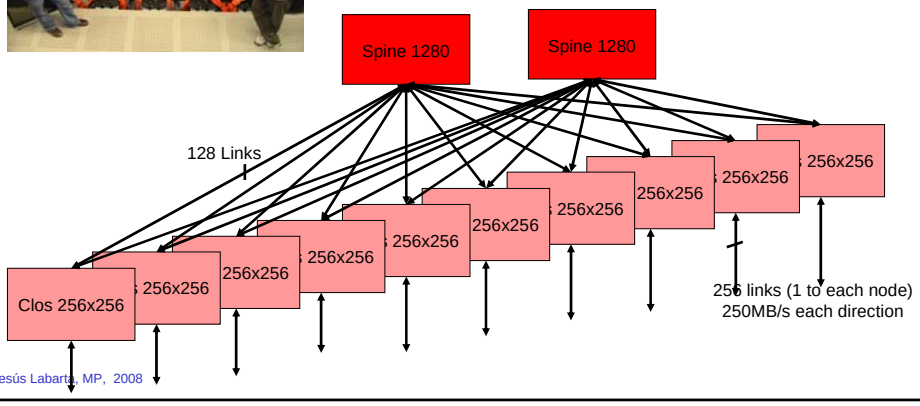
- **Amount of buffer space**
- **Latency**
  - F(routing/packaging overhead, # hops)
- **Bandwidth**
- **Contention**
  - Hot spots
    - ✓ Tree saturation

Network interface

- **DMA channel**
  - IO device.
  - Incoming messages have to be processed after arrival by main processor.
  - Protection issues
    - ✓ Kernel mode access
- **User level access**
  - Map control registers into user address space
  - Network transactions identify system / user
- **Dedicated message processors**

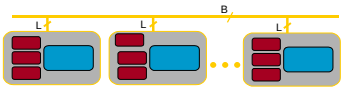
Hardware: Myrinet

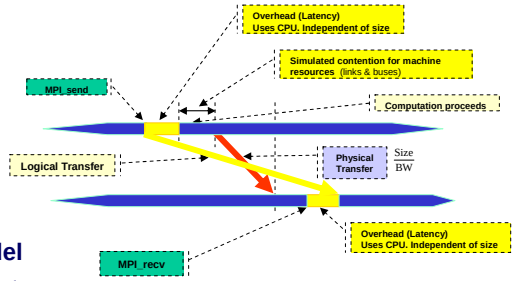




Objective

- **Parameters for a Performance Model**
  - Dimemas
- **Observe detailed MPI behavior**
  - Paraver

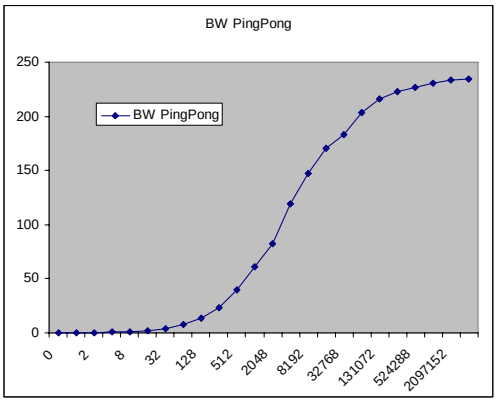




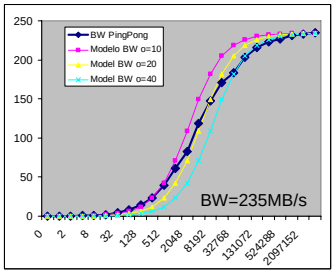
- **Basic model**
  - Late receiver

Ping pong

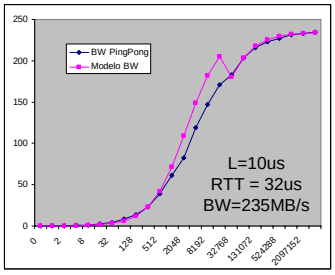
- **MPICH-GM 2.0.61**



Ping pong

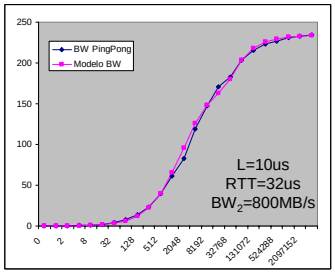


$$T = L + \frac{size}{BW}$$

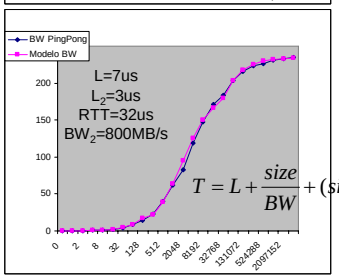


$$T = L + \frac{size}{BW} + (size > 32K\,?\,RTT : 0)$$

Ping pong



$$T = L + \frac{size}{BW} + (size < 32K\,?\,\frac{size}{BW_2} : 0) + (size > 32K\,?\,RTT : 0)$$

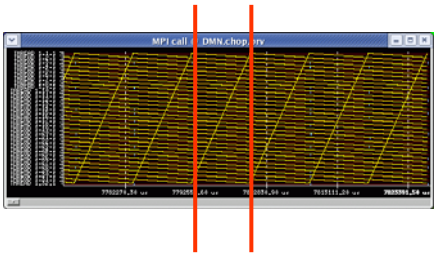


$$T = L + \frac{size}{BW} + (size > 128\,?\,L_2 : 0) + (size < 32K\,?\,\frac{size}{BW_2} : 0) + (size > 32K\,?\,RTT : 0)$$

IMB SendRec Multi

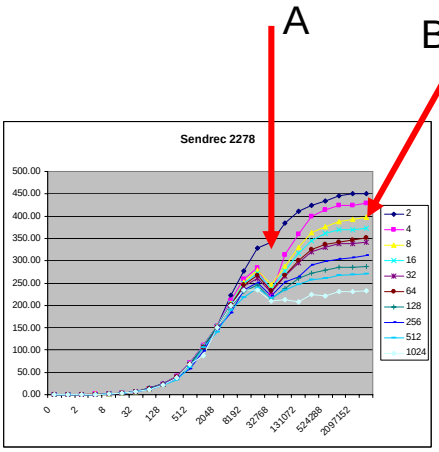
g=2
g=4
g=8

Expected behavior within each group



Jesús Labarta, MP, 2008

GM: IMB SendRec Multi



g=2
g=4
g=8

1process per node (if p<2400)

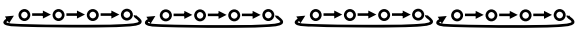
Jesús Labarta, MP, 2008

Curiosities

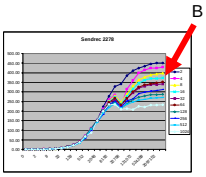
- **A:**
  - Correlated to protocol change, but reason?
    - ✓ i.e. Why not at g=2
    - ✓ Why very similar value for all G
- **B:**
  - Contention
  - Preemptions ???
  - Communication errors: Lost packets, … ???

Contention

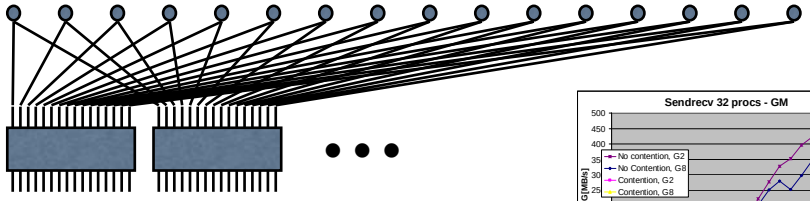
g=2 

g=4 

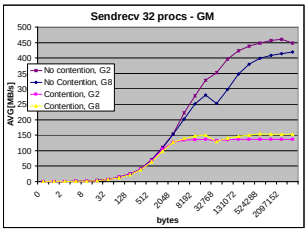
- **Links shared between N communications → BW / N** **(Very drastic)**
- **Contention dependent on communication pattern**
  - Depends on application pattern
  - Depends on processor allocation
  - Network topology (routing) varies with time.
- **Communication pattern multiplies effect of contention**
  - Group bandwidth determined by worst link (Very drastic)
- **Averaging independent patterns smoothes numbers**



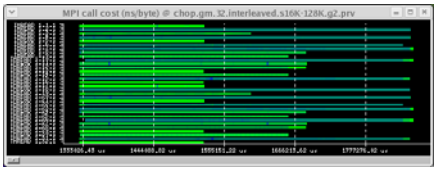
Contention



(g,0) → (g,1) → (g,2) → (g,3)

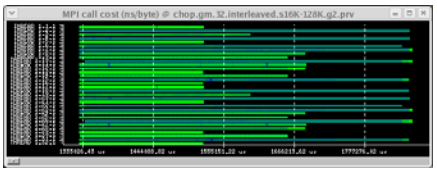


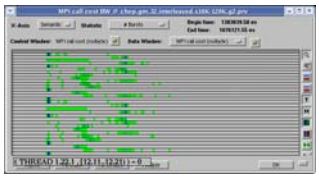
(g,0) (g,2) (g,1) (g,3)

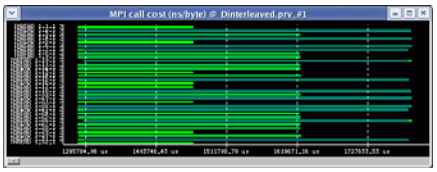


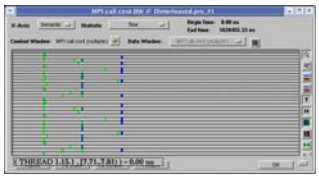
Contention

(g,0) (g,2) (g,1) (g,3)



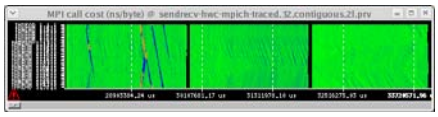


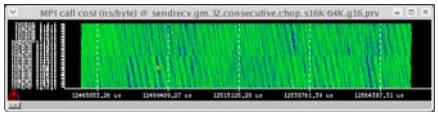




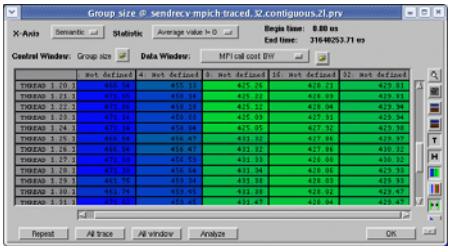
Bubbles

- **Guaranteeing no contention:**
  - Cost of each MPI call (ns/byte)?





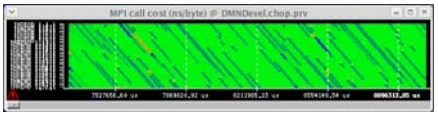
- **Bubbles:**
  - Propagation
  - Cause?



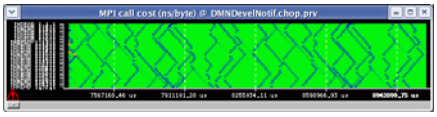
Bubbles

- **How they propagate?**
  - Dimemas: inject periodic bubbles

Eager

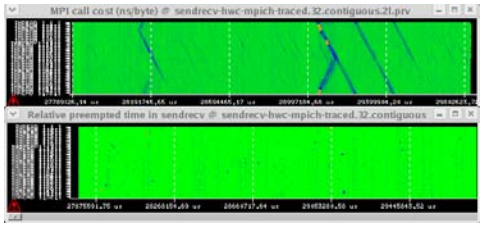


Rendez vous



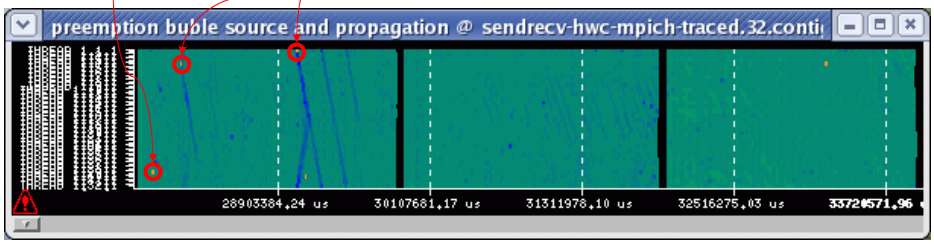
Bubbles

- **Cause?**
  - Preemptions



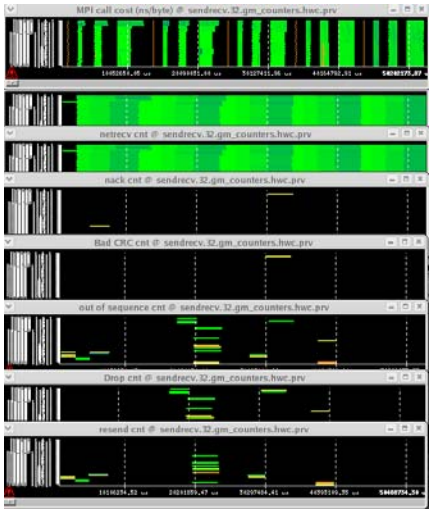
Preemption causes bubble chain

Preemption does not cause bubble chain



Bubbles

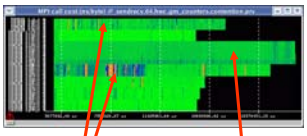
- **Read GM counters**
  - High overhead
  - CRC, resends,….
    - ✓ High potential, risk of reproducing processor counters
- **Cause or effect??**



Global impact

- **on large systems?**
- **In multiuser environments**

64 nodes, G=8, 4MB

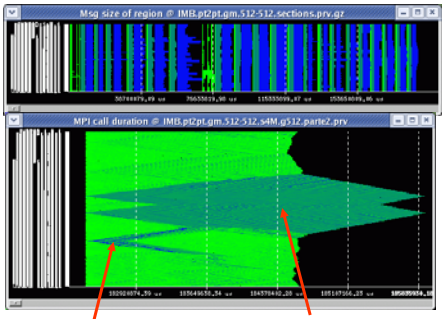


External contention

Internal contention

512 nodes, 4MB

Dependence on appl. phase (comm. Pattern)



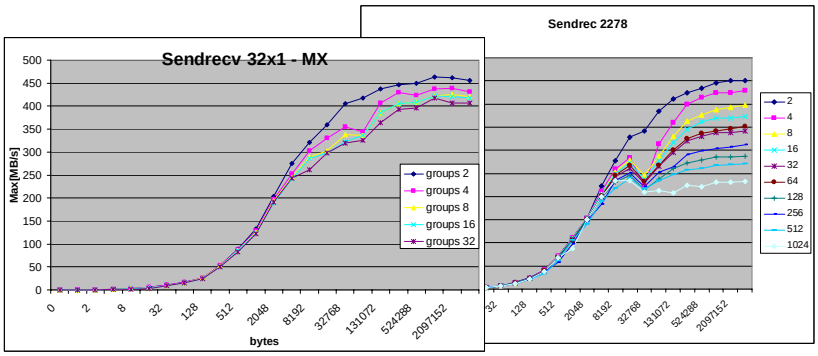
Bubble propagation

Propagation of internal contention

What is the benchmark measuring?
Appropriate number of iterations?

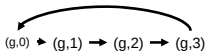
Bubbles

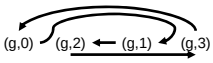
- **Makes things a dynamic routing any better?**



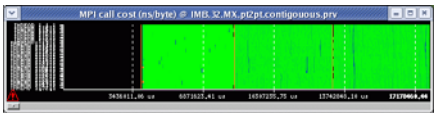
MX

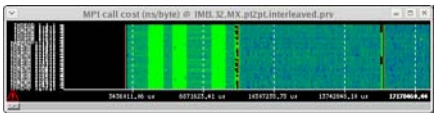
- **Should route dispersion correct problems??**





32 nodes 1 2 4MB



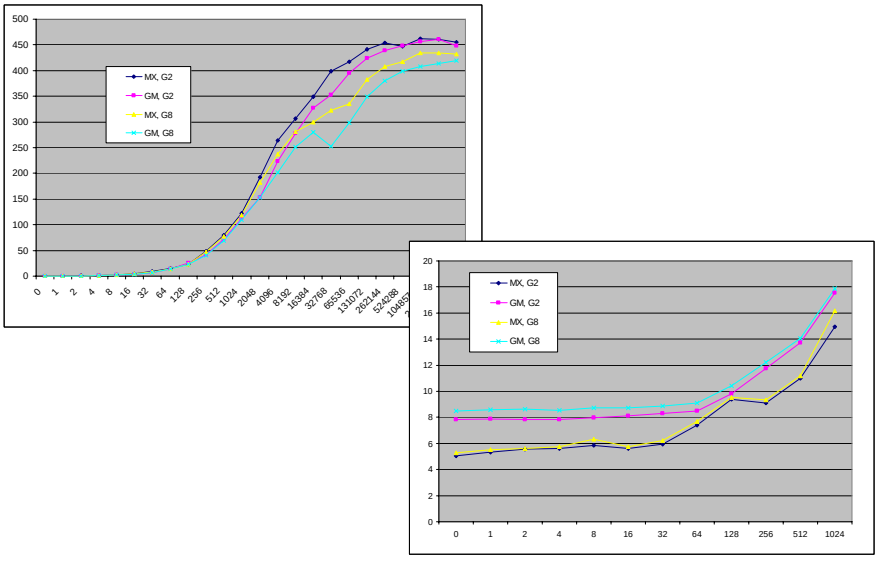


Stability issues

How to reach a globally good set of routes

How to avoid leaving a globally good set of routes

MX vs GM

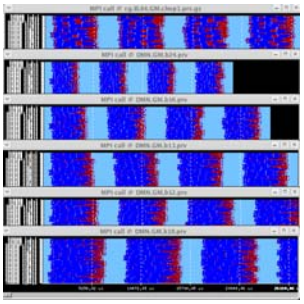


NAS CG benchmark

- **Modeling contention**

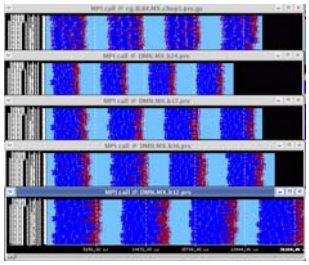
GM

Real
B=24
B=16
B=13
B=12
B=10



MX

Real
B=24
B=17
B=16
B=12



Jesús Labarta, MP, 2008