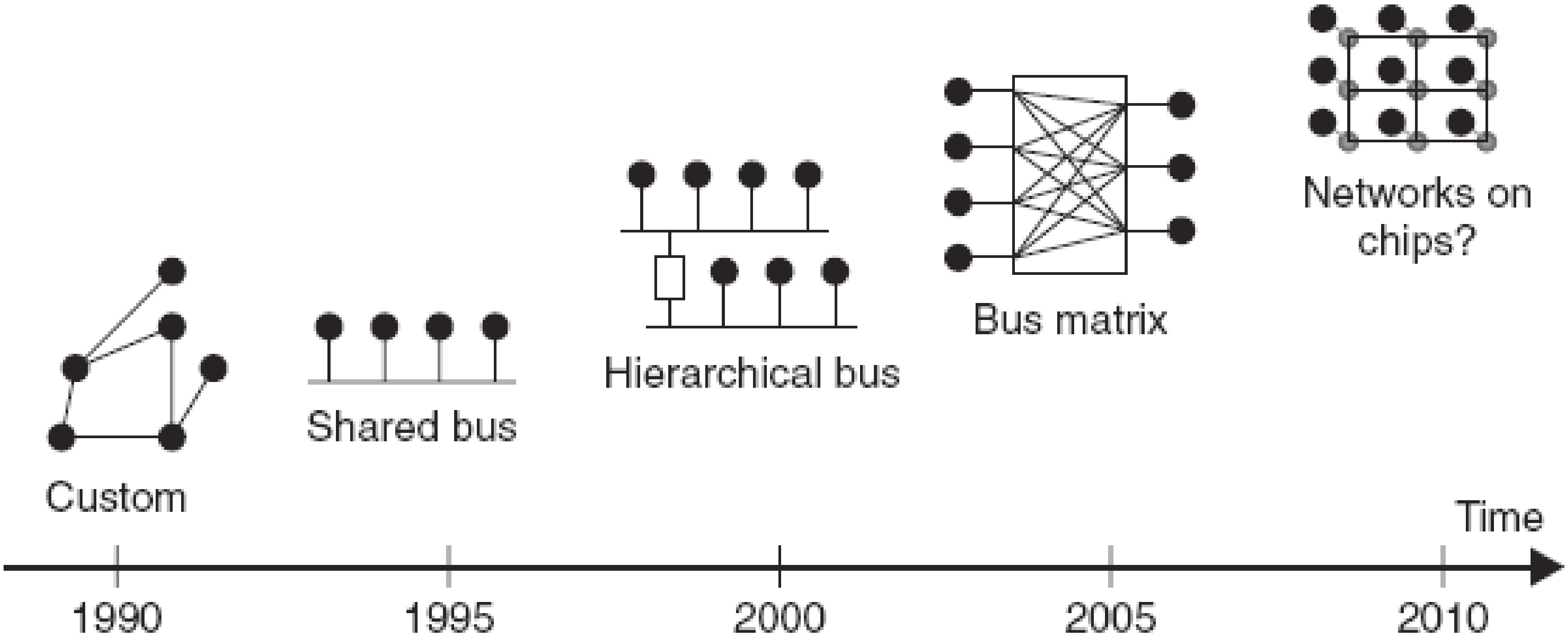
EE 531: ADVANCED VLSI DESIGN

Network on Chip Communications

Nishith N. Chakraborty

March, 2025

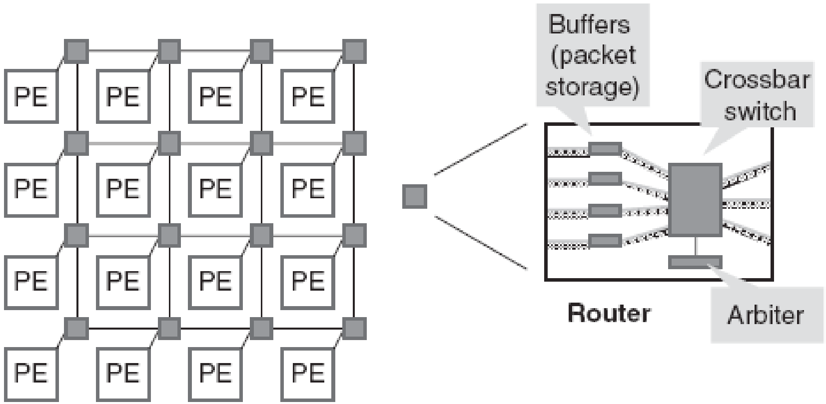
EVOLUTION OF COMMUNICATION ARCHITECTURES



Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

NETWORK-ON-CHIP

- Network-on-chip (NoC) is a packet switched on-chip communication network designed using layered methodology
  - ➢ “routes packets, not wires”
- NoCs use packets to route data from the source to the destination PE via a network fabric that consists of
  - ➢ Switches (routers)
  - ➢ Interconnection links (wires)

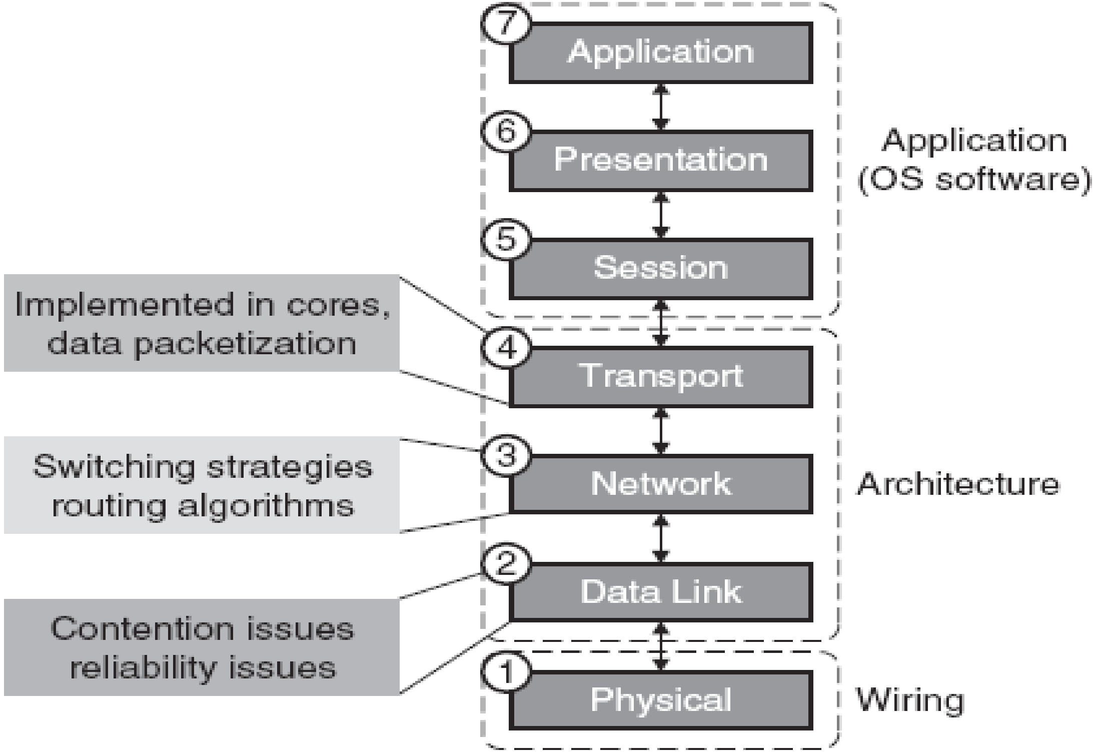


Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

NETWORK-ON-CHIP

- NoCs are an attempt to scale down the concepts of largescale networks, and apply them to the embedded system-on-chip (SoC) domain
- NoC Properties:
  - Regular geometry that is scalable
  - Flexible QoS guarantees
  - Higher bandwidth
  - Reusable components
    - Buffers, arbiters, routers, protocol stack
  - No long global wires (or global clock tree)
    - No problematic global synchronization
    - GALS: Globally asynchronous, locally synchronous design
  - Reliable and predictable electrical and physical properties

NETWORK PROTOCOL STACK



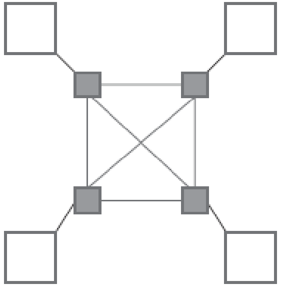
Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

CAL POLY
SAN LUIS OBISPO

NoC Topologies

DIRECT TOPOLOGIES

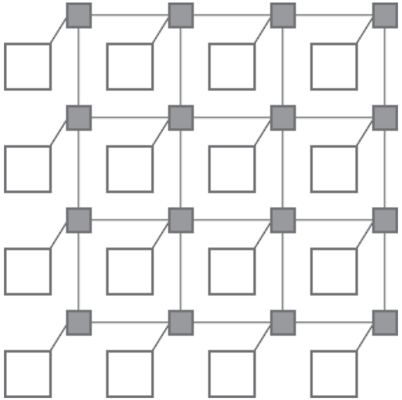
- Each node has direct point-to-point link to a subset of other nodes in the system called neighboring nodes
- Nodes consist of computational blocks and/or memories, as well as a Network Interface (NI) block that acts as a router
- As the number of nodes in the system increases, the total available communication bandwidth also increases
- Fundamental trade-off is between connectivity and cost



Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

DIRECT TOPOLOGIES

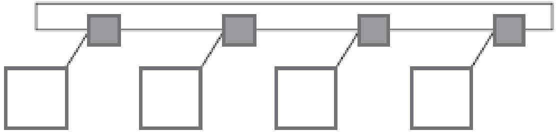
- Most direct network topologies have an orthogonal implementation, where nodes can be arranged in an n-dimensional orthogonal space
  - Routing for such networks is fairly simple
  - E.g. n-dimensional mesh, torus, folded torus, hypercube, and octagon
- 2D mesh is most popular topology
  - All links have the same length
    - Eases physical design
  - Area grows linearly with the number of nodes
- Must be designed in such a way as to avoid traffic accumulating in the center of the mesh



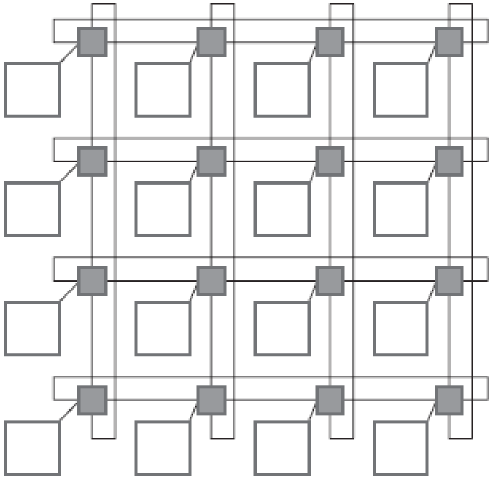
Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

TORUS TOPOLOGY

- Torus topology, also called a k-ary n-cube, is an n-dimensional grid with k nodes in each dimension
- k-ary 1-cube (1-D torus) is essentially a ring network with k nodes
  - ➢ Limited scalability as performance decreases when more nodes



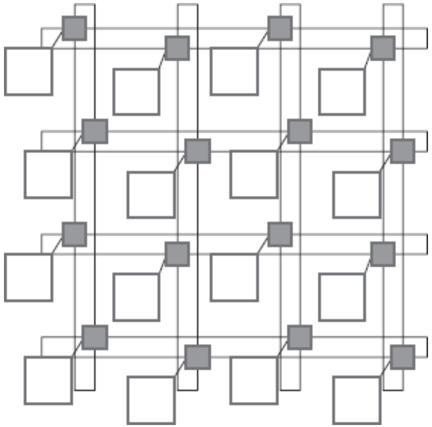
- k-ary 2-cube (i.e., 2-D torus) topology is similar to a regular mesh
  - ➢ Except that nodes at the edges are connected to switches at the opposite edge via wrap- around channels
  - ➢ Long end-around connections can, however, lead to excessive delays



Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

TORUS TOPOLOGY

- Folding torus topology overcomes long link limitation of 2-D torus
  - ➢ Links have the same size

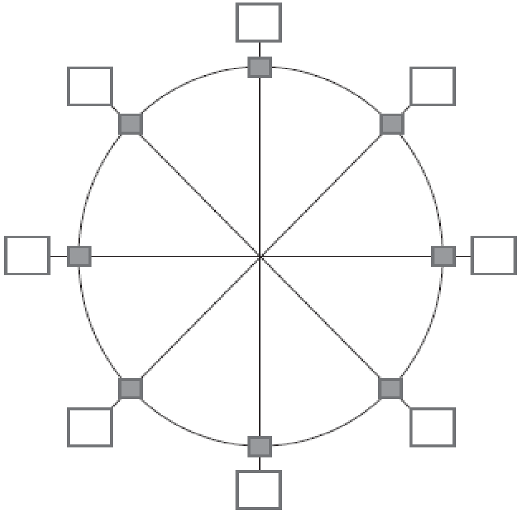


- Meshes and tori can be extended by adding bypass links to increase performance at the cost of higher area

Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

OCTAGON TOPOLOGY

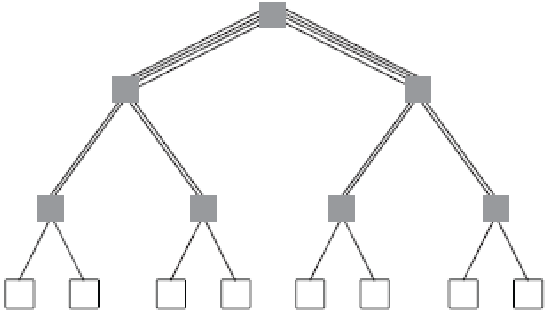
- Messages sent between any 2 nodes require at most two hops
- More octagons can be tiled together for larger designs
  - One of the nodes is used as a bridge node



Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

INDIRECT TOPOLOGIES

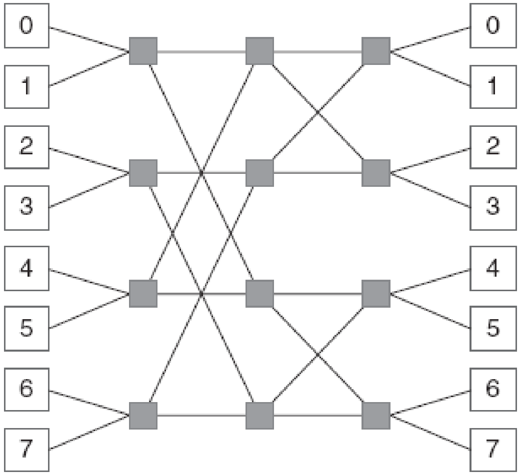
- Indirect Topologies
  - Each node is connected to an external switch, and switches have point-to-point links to other switches
  - Switches do not perform any information processing, and correspondingly nodes do not perform any packet switching
  - E.g. SPIN, crossbar topologies
- Fat tree topology
  - Nodes are connected only to the leaves of the tree
  - More links near root, where bandwidth requirements are higher



Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

BUTTERFLY NETWORK

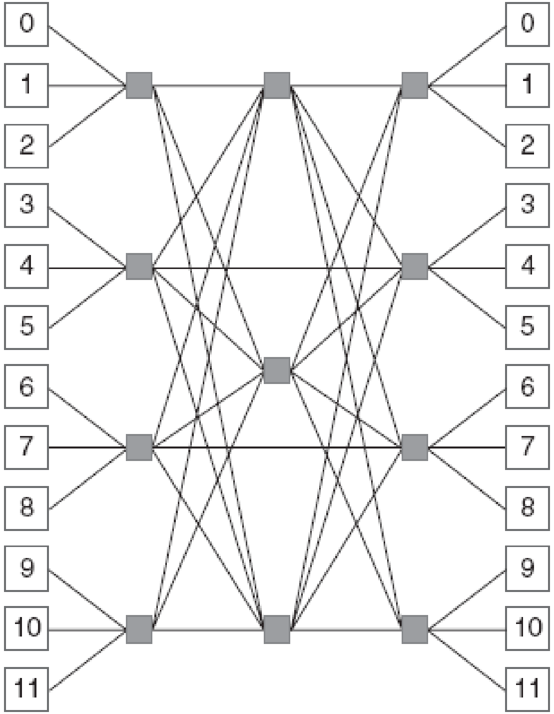
- Blocking multi-stage network – packets may be temporarily blocked or dropped in the network if contention occurs
- k^n nodes, and n stages of k^{n-1} k x k crossbar
- Example: 2-ary 3-fly butterfly network



Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

SYMMETRIC (M, N, R) CLOS NETWORK

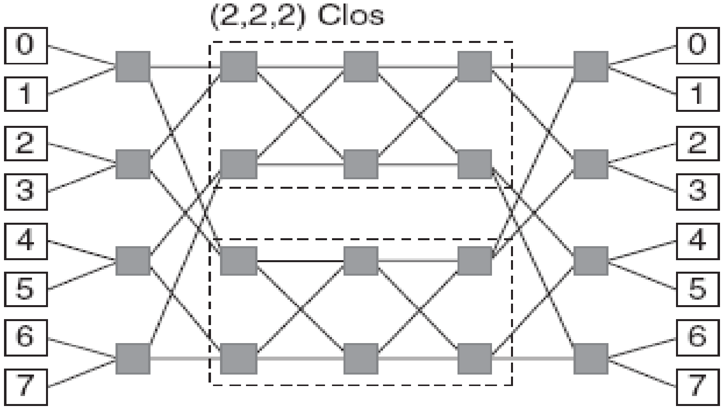
- Three-stage network in which each stage is made up of a number of crossbar switches
- **m** is no. of middle-stage switches
- **n** is number of input/output nodes on each input/output switch
- **r** is number of input and output switches
- E.g. (3, 3, 4) Clos network
- Non-blocking network
- Expensive (several full crossbars)



Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

BENES NETWORK

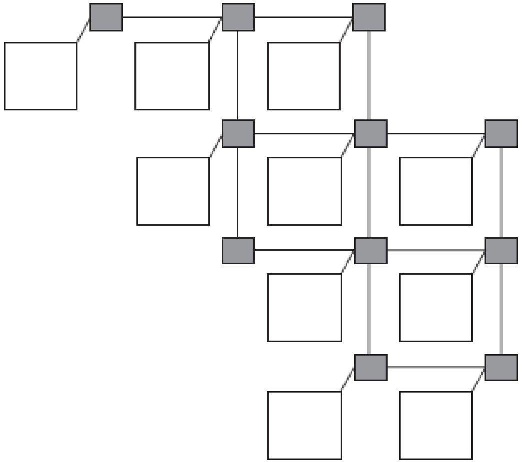
- Rearrangeable network in which paths may have to be rearranged to provide a connection, requiring an appropriate controller
- Clos topology composed of 2 x 2 switches
- Example: (2, 2, 4) re-arrangeable Clos network constructed using two (2, 2, 2) Clos networks with 4 x 4 middle switches

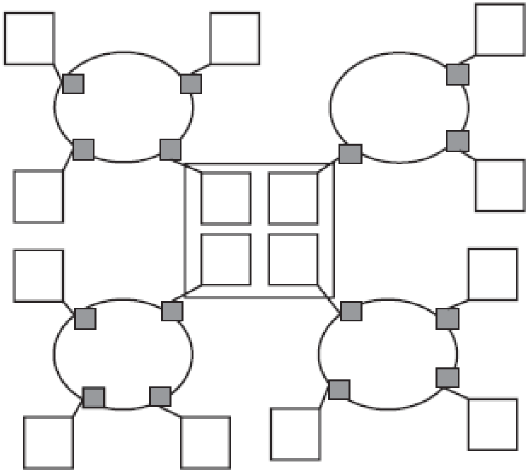


Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

IRREGULAR/AD HOC TOPOLOGIES

- Customized for an application
- Usually mix of shared bus, direct, and indirect network topologies
- Example: reduced mesh, cluster-based hybrid topology





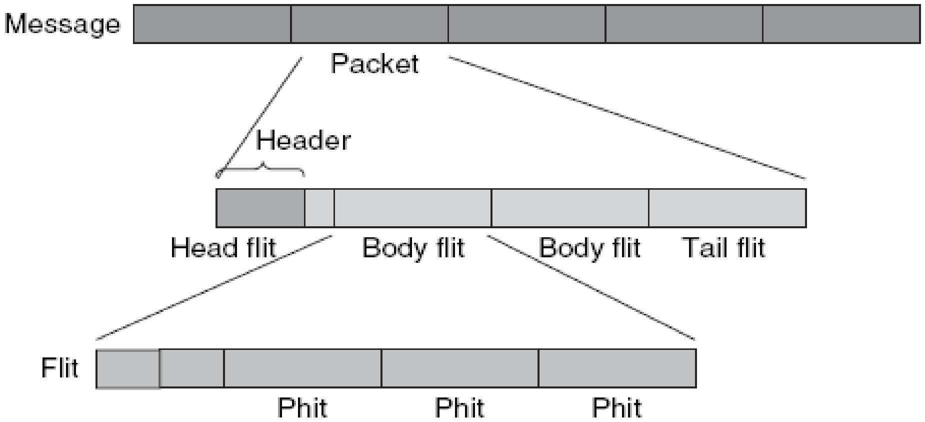
Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

Switching Strategies

CAL POLY
SAN LUIS OBISPO

SWITCHING STRATEGIES

- Determine how data flows through routers in the network
- Define granularity of data transfer and switching technique
  - *phit* is a unit of data that is transferred on a link in a single cycle



Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

SWITCHING STRATEGIES

- Two main modes of transporting flits in a NoC are circuit switching and packet switching
- Circuit switching
  - Physical path between the source and the destination is reserved prior to the transmission of data
  - Message header flit traverses the network from the source to the destination, reserving links along the way
- Pro: low latency transfers, once path is reserved
- Con: pure circuit switching doesn't scale well with NoC size
  - Several links occupied for duration of transmission, even if no data
    - For instance in the setup and tear down phases

VIRTUAL CIRCUIT SWITCHING

- Creates virtual circuits that are multiplexed on links
- Number of virtual links (or virtual channels (VCs)) that can be supported by physical link depends on buffers allocated
- Allocate 1 buffer per virtual link or 1 buffer per physical link
- Allocating one buffer per virtual link
  - ➢ Depends on how virtual circuits are spatially distributed in the NoC, routers can have a different number of buffers
  - ➢ Can be expensive due to the large number of shared buffers
  - ➢ Multiplexing virtual circuits on a single link also requires scheduling at each router and link (end-to-end schedule)
  - ➢ Conflicts between different schedules can make it difficult to achieve bandwidth and latency guarantees

ONE BUFFER PER PHYSICAL LINK

- Virtual circuits are time multiplexed with one buffer per link
- Uses time division multiplexing (TDM) to statically schedule the usage of links among virtual circuits
- Flits are typically buffered at the NIs and sent into the NoC according to the TDM schedule
- Global scheduling with TDM makes it easier to achieve end-to-end bandwidth and latency guarantees
- Less expensive router implementation, with fewer buffers

MORE SWITCHING STRATEGIES

- Packet Switching
  - Packets transmitted from source make way independently to receiver
    - Possibly along different routes and with different delays
  - Zero start up time, followed by a variable delay due to contention in routers along packet path
  - QoS guarantees harder to make in packet switching
  - Three main packet switching scheme variants
- SAF (store and forward) switching
  - Packet is sent from one router to the next only if the receiving router has buffer space for entire packet
  - Buffer size in the router is at least equal to the size of a packet
  - Disadvantage: excessive buffer requirements

MORE SWITCHING STRATEGIES

- VCT (virtual cut through) Switching
  - Reduces router latency over SAF switching by forwarding first flit of packet as soon as space for entire packet is available in the next router
  - If no space is available in receiving buffer, no flits are sent, and the entire packet is buffered
  - Same buffering requirements as SAF switching
- WH (wormhole) switching
  - Flit from packet forwarded to receiving router if space exists
  - Parts of the packet can be distributed among two or more routers
  - Buffer requirements are reduced to one flit, instead of entire packet
  - More susceptible to deadlocks due to dependencies between links

Routing Algorithms

ROUTING ALGORITHMS

- Responsible for correctly and efficiently routing packets or circuits from the source to the destination
- Routing algorithm choice depends on several trade-offs
  - Minimize power required for routing
  - Minimize logic and routing tables to achieve lower area footprint
  - Increasing performance by reducing delay and maximizing traffic utilization of the network
  - Improving robustness to better adapt to changing traffic needs
- Routing schemes can be classified into several categories
  - Static or dynamic routing
  - Distributed or source routing
  - Minimal or non-minimal routing

STATIC & DYNAMIC ROUTING

- Static routing: fixed paths used to transfer data
  - Does not take into account current state of the network
- Advantages of static routing:
  - Easy to implement, since very little additional router logic is required
  - In-order packet delivery if single path is used
- Dynamic routing: routing decisions made according to current state
  - Considering factors such as availability and load on links
  - Path between source and destination may change over time
    - As traffic conditions and requirements of the application change
  - More resources needed to monitor state of the network and dynamically change routing paths
  - Able to better distribute traffic in a network

DISTRIBUTED & SOURCE ROUTING

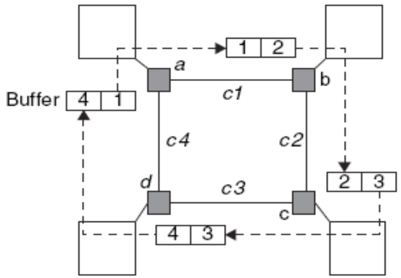
- Static & dynamic routing schemes are further classified depending on where routing information is stored, and where routing decisions made
- Distributed routing: each packet carries the destination address
  - E.g., XY co-ordinates or number identifying destination node/router
  - Routing decisions are made in each router by looking up the destination addresses in a routing table or by executing a hardware function
- Source routing: packet carries routing information
  - Pre-computed routing tables are stored at a nodes' NI
  - Routing information is looked up at the source NI and routing information added to the packet header (increasing packet size)
  - Routing information extracted from routing field in header
  - Does not require a destination address in a packet, any intermediate routing tables, or functions needed to calculate the route

MINIMAL & NON-MINIMAL ROUTING

- Minimal routing: length of the routing path from the source to the destination is the shortest possible length between the two nodes
  - E.g., a mesh NoC topology if source node is at (0, 0) and destination node is at (i, j), then minimal path length is |i| + |j|
  - Source doesn't send a packet if minimal path not available
- Non-minimal routing: use longer paths if min path not avail.
  - By allowing non-minimal paths, the number of alternative paths is increased, which can be useful for avoiding congestion
  - Disadvantage: overhead of additional power consumption

DEADLOCKS

- Routing algorithm must ensure freedom from deadlocks
- Common in WH switching
- E.g., cyclic dependency shown below



- Freedom from deadlocks can be ensured by allocating additional hardware resources or imposing restrictions on the routing
- Usually dependency graph of the shared network resources is built and analyzed either statically or dynamically

ROUTING ALGORITHMS

- Routing algorithm must ensure freedom from livelocks
    - Livelocks similar to deadlocks, except states involved constantly change with regard to one another, without making any progress
        - Occurs especially when dynamic (adaptive) routing is used
        - E.g. can occur in deflective “hot potato” routing if packet bounced around between routers and never reaches destination
    - Livelocks can be avoided with simple priority rules
- Routing algorithm must ensure freedom from starvation
    - Under scenarios where certain packets prioritized, some low priority packets never reach their intended destination
    - Can be avoided by using a fair routing algorithm, or reserving some bandwidth for low priority data packets

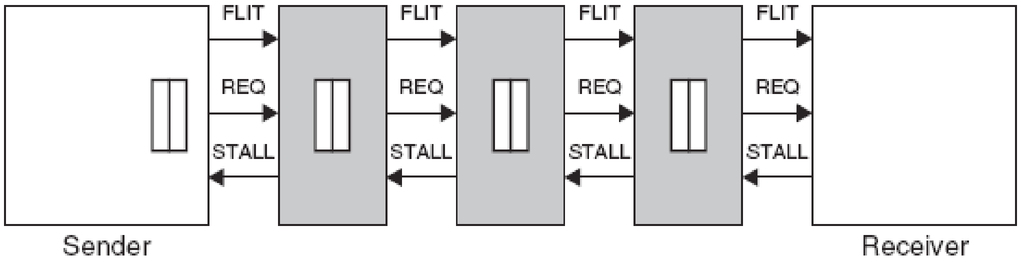
Flow Control Schemes

FLOW CONTROL SCHEMES

- Goal of flow control: allocate network resources for packets
  - Can be viewed as resolving contention during packet traversal
- At data link-layer, when transmission errors occur, recovery from error depends on support provided by flow control
  - E.g. if corrupted packet needs to be retransmitted, flow of packets from sender must be stopped, and request signaling must be performed to reallocate buffer and bandwidth resources
- Most flow control techniques can manage link congestion
- But not all schemes can (by themselves) reallocate all the resources required for retransmission when errors occur
  - Either error correction or a scheme to handle reliable transfers must be implemented at a higher layer

STALL/GO SCHEME

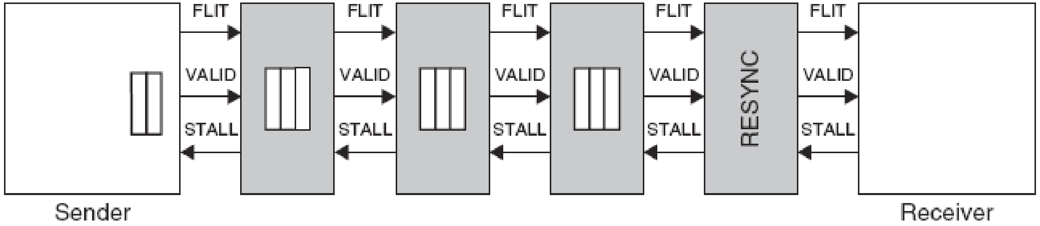
- Low overhead scheme
- Requires only two control wires
  - One going forward and signals data availability
  - Other goes backward, signals buffers filled (STALL) or buffers free (GO)
- Can be implemented with distributed buffering (pipelining) along link
- Good performance – fast recovery from congestion
- Does not have any provision for fault handling



Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

T-ERROR SCHEME

- More aggressive scheme that can detect faults
- Delayed clock re-samples input data to detect any inconsistencies
- Resynchronization stage between end of link and receiving switch
- Timing budget can be used to provide greater reliability by configuring links with appropriate spacing and frequency
- Does not provide a thorough fault handling mechanism

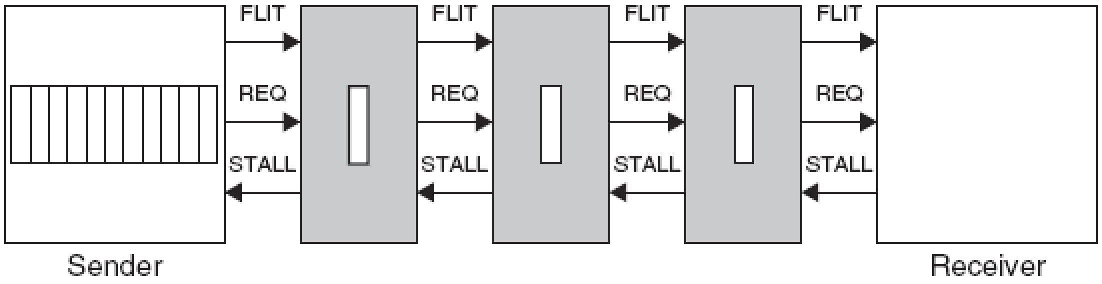


Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

CAL POLY
SAN LUIS OBISPO

ACK/NACK SCHEME

- When flits sent on link, local copy kept in buffer by sender
- When ACK received by sender, deletes copy of flit from local buffer
- When NACK received, sender rewinds output queue and starts resending flits, starting from the corrupted one
- Implemented either end-to-end or switch-to-switch
- Fault handling support comes at cost of greater power, area overhead



Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

NETWORK & TRANSPORT-LAYER FLOW CONTROL

- Flow Control without Resource Reservation
  - Technique #1: drop packets when receiver NI full
    - Improves congestion in short term but increases it in long term
  - Technique #2: return packets that do not fit into receiver buffers to sender
    - To avoid deadlock, rejected packets must be accepted by sender
  - Technique #3: deflection routing
    - When packet cannot be accepted at receiver, it is sent back into network
    - Packet keeps hopping from router to router till it is accepted at receiver
- Flow Control with Resource Reservation
  - Credit-based flow control with resource reservation
  - Credit counter at sender NI tracks free space available in receiver NI buffers
  - Credit packets can piggyback on response packets
  - End-to-end or link-to-link

Source: S. Pasricha and N. Dutt, On-Chip Comm. Arch. – SOC Interconnect, Morgan Kaufmann, 2005.

Thank you!