

## COMPUTER NETWORK II

### INTERCONNECTION OF NETWORKS

**Interconnection networks** are composed of switching elements. Topology is the pattern to connect the individual switches to other elements, like processors,

#### **Interconnection Network Design**

An **interconnection network** in a parallel machine transfers information from any source node to any desired destination node. This task should be completed with as small latency as possible. It should allow a large number of such transfers to take place concurrently. Moreover, it should be inexpensive as compared to the cost of the rest of the machine.

The network is composed of links and switches, which helps to send the information from the source node to the destination node. A network is specified by its topology, routing algorithm, switching strategy, and flow control mechanism.

#### **Organizational Structure**

Interconnection networks are composed of following three basic components –

- **Links** – A link is a cable of one or more optical fibers or electrical wires with a connector at each end attached to a switch or network interface port. Through this, an analog signal is transmitted from one end, received at the other to obtain the original digital information stream.
- **Switches** – A switch is composed of a set of input and output ports, an internal “cross-bar” connecting all input to all output, internal buffering, and control logic to effect the input-output connection at each point in time. Generally, the number of input ports is equal to the number of output ports.
- **Network Interfaces** – The network interface behaves quite differently than switch nodes and may be connected via special links. The network interface formats the packets and constructs the routing and control information. It may have input and output buffering, compared to a switch. It may perform end-to-end error checking and flow control. Hence, its cost is influenced by its processing complexity, storage capacity, and number of ports.

#### **Interconnection Network**

Interconnection networks are composed of switching elements. Topology is the pattern to connect the individual switches to other elements, like processors, memories and other switches. A network allows exchange of data between processors in the parallel system.

- **Direct connection networks** – Direct networks have point-to-point connections between neighboring nodes. These networks are static, which means that the point-to-point connections are fixed. Some examples of direct networks are rings, meshes and cubes.
- **Indirect connection networks** – Indirect networks have no fixed neighbors. The communication topology can be changed dynamically based on the application

demands. Indirect networks can be subdivided into three parts: bus networks, multistage networks and crossbar switches.

- **Bus networks** – A bus network is composed of a number of bit lines onto which a number of resources are attached. When busses use the same physical lines for data and addresses, the data and the address lines are time multiplexed. When there are multiple bus-masters attached to the bus, an arbiter is required.
- **Multistage networks** – A multistage network consists of multiple stages of switches. It is composed of 'axb' switches which are connected using a particular interstage connection pattern (ISC). Small 2x2 switch elements are a common choice for many multistage networks. The number of stages determine the delay of the network. By choosing different interstage connection patterns, various types of multistage network can be created.
- **Crossbar switches** – A crossbar switch contains a matrix of simple switch elements that can switch on and off to create or break a connection. Turning on a switch element in the matrix, a connection between a processor and a memory can be made. Crossbar switches are non-blocking, that is all communication permutations can be performed without blocking.

### **Evaluating Design Trade-offs in Network Topology**

If the main concern is the routing distance, then the dimension has to be maximized and a hypercube made. In store-and-forward routing, assuming that the degree of the switch and the number of links were not a significant cost factor, and the numbers of links or the switch degree are the main costs, the dimension has to be minimized and a mesh built.

In worst case traffic pattern for each network, it is preferred to have high dimensional networks where all the paths are short. In patterns where each node is communicating with only one or two nearby neighbors, it is preferred to have low dimensional networks, since only a few of the dimensions are actually used.

### **Routing**

The routing algorithm of a network determines which of the possible paths from source to destination is used as routes and how the route followed by each particular packet is determined. Dimension order routing limits the set of legal paths so that there is exactly one route from each source to each destination. The one obtained by first traveling the correct distance in the high-order dimension, then the next dimension and so on.

### **Deterministic Routing**

A routing algorithm is deterministic if the route taken by a message is determined exclusively by its source and destination, and not by other traffic in the network. If a routing algorithm only selects shortest paths toward the destination, it is minimal, otherwise it is non-minimal.

### **Deadlock Freedom**

Deadlock can occur in a various situations. When two nodes attempt to send data to each other and each begins sending before either receives, a 'head-on' deadlock may occur. Another case of deadlock occurs, when there are multiple messages competing for resources within the network.

The basic technique for proving a network is deadlock free, is to clear the dependencies that can occur between channels as a result of messages moving through the networks and to show that there are no cycles in the overall channel dependency graph; hence there is no traffic patterns that can lead to a deadlock. The common way of doing this is to number the channel resources such that all routes follow a particular increasing or decreasing sequences, so that no dependency cycles arise.

## **Switch Design**

Design of a network depends on the design of the switch and how the switches are wired together. The degree of the switch, its internal routing mechanisms, and its internal buffering decides what topologies can be supported and what routing algorithms can be implemented. Like any other hardware component of a computer system, a network switch contains data path, control, and storage.

## **Ports**

The total number of pins is actually the total number of input and output ports times the channel width. As the perimeter of the chip grows slowly compared to the area, switches tend to be pin limited.

## **Channel Buffers**

The organization of the buffer storage within the switch has an important impact on the switch performance. Traditional routers and switches tend to have large SRAM or DRAM buffers external to the switch fabric, while in VLSI (Very Large Scale Integration) switches the buffering is internal to the switch and comes out of the same silicon budget as the datapath and the control section. As the chip size and density increases, more buffering is available and the network designer has more options, but still the buffer real-estate comes at a prime choice and its organization is important.

## **Flow Control**

When multiple data flows in the network attempt to use the same shared network resources at the same time, some action must be taken to control these flows. If we don't want to lose any data, some of the flows must be blocked while others proceed.

The problem of flow control arises in all networks and at many levels. But it is qualitatively different in parallel computer networks than in local and wide area networks. In parallel computers, the network traffic needs to be delivered about as accurately as traffic across a bus and there are a very large number of parallel flows on very small-time scale.