2.5 Interconnection Nets for Parallel Computers

2.5.1 Introduction

Harald Richter

Today, interconnection networks benefit from a rising interest in communi cation technologies. There are close connections between networks for tele communication, computers, and those inside of computers, as we can see from the asynchronous transfer mode (ATM) technology. This communication standard is very similar in its topology and protocols to networks used in multiprocessors or multicomputers. Transmission rates and latencies of both applications are approaching each other. In the future they will finally merge to multimedia devices. Therefore, research in the area of coupling techniques is a prerequisite for the integration of computers and communication. Improvements in one area influence the other and vice versa. Key issues in the design of parallel computers are in the construction of the computing nodes and their interconnects because they define the primary parameters of the machine. Regardless of the underlying programming models, e.g., shared memory, distributed shared memory, virtual shared memory or message passing, interconnection networks are responsible for the physical coupling of nodes and memory modules. Traditional coupling methods instead, like buses or multiport memories, are loosing importance.

2.5.2 Work Performed

In the year 1994 a research project on interconnection networks for parallel computers was started to investigate key issues in this field. The work is per formed under a Max-Planck scholarship. It was granted to Dr.-Ing. H. Richter for three years to enable a "Habilitationsarbeit" on that research area. In the first part of the work carried out last year, a survey of the architectures of the currently most important parallel computers was performed. Therefore, machines in industry and academia were analysed. Among them were the Meiko CS-2, nCube-1/2, MasPar 1/2, KSR 1/2, TMC CM5, Intel Paragon, Parsytec GC, Convex SPP-1000, Fujitsu VPP500, Cray T3D, IBM SP1/2, GMD MANNA, Stanford Flash/Dash, MIT J machine, and the Caltech Mosaic C. The interconnection networks of the Cray T3D and IBM SP1/2 parallel computers were studied extensively and compared with each other on basis of benchmarks and architectural properties. An interesting result of this comparison was that a sophisticated design of a network can compensate a low transmission rate. For one application (LINPACK) it became apparent that the performance of an IBM SP1 with 64 processors is not significantly worse than that of the Cray T3D, although the IBM network is much slower than that of the Cray. For processor numbers higher than 256 however, the Cray net proved to be more efficient. For such big machines, the ECL based net with its hardware support for parallel programming was able to deliver up to 50% efficiency. The implementation of the Cray T3D network is very expensive. In contrast, the much cheaper CMOS-based net of IBM implements intensive queuing and scheduling mechanisms and has thereby gained a decent throughput. The differences between both approaches become visible in terms of delay times: while Cray demonstrated 1.5 μs delay for a process-to-process communication, IBM showed 38 μs, thus limiting the granularity of parallel programs to coarse grain applications.

In the second part of the work performed last year the architectural principles of interconnection networks were summarized in talks and papers. The following aspects were investigated in detail: 1.) single stage networks, 2.) data transfer in nets, 3.) routing methods, 4.) deadlock prevention, 5.) data formats, and 6.) multi stage interconnection networks. Furthermore, a general formula was devised for the computation of the number of connection possibilities in a network. Finally, a classification scheme of networks has been developed. In the following, this classification scheme is presented in a condensed form to help the reader to orient himself in this current research area.

2.5.3 A Classification Scheme for Interconnection Networks

Interconnection networks are the coupling media for putting together proces sors, nodes, or whole computers to new entities, like multiprocessors, multi computers or distributed systems. In interconnection nets a rich variety of design possibilities is contained due to their different areas of application. Therefore, a classification scheme for connection technologies has to be flexi ble enough to cover all fields of applications. Additionally, such a scheme has to be orthogonal so that the classification parameters are separable and can be composed in arbitrary ways. Besides these properties, two goals have to be pursued in classification schemes: 1.) Existing nets should be precisely describable by using the design parameters that are offered from the scheme; 2.) New nets should be definable by grouping parameters in a novel way. Both goals require that the classification is unambiguous and that it contains most of the possible design space for network construction in its principal components, i.e., that it is complete. The proposed scheme is considered to have the mentioned properties and to fulfill the goals. The classification is presented in a Backus-Naur like notation because this is practicable and formal at the same time. To illustrate the concerns let's start with some examples:

2.5.3.1 Examples of Coupling Techniques

The most simple coupling medium used to bridge a distance of several centi meters is not a bus, as one might expect, but a shift register. Data at the serial input of such a device are moved clockwise to the output. That is a very simple method to transfer data between a transmitter and a receiver. By reversing the shift direction the flow of information is also reversed, thereby forming a bidirectional transmission medium. A more complicated communication medium is a bus. In such a device a two step hierarchy is introduced between the bus master and its slaves. Only one bus master is allowed at any time since the bus is based on timesharing of signals. Many receivers can listen to the information that is broadcasted by the master which in turn can read or write each of its slaves. Although cheap in construction, buses are limited in length and bandwidth thereby imposing severe constraints on the design of high speed multiprocessors. Extensions to a bus in terms of the distance that can be covered are serial busses like Ethernet, FDDI or Token Ring. They are optimized to transfer data over longer distances but with even smaller bandwidth. Finally, another example of a simple and well known coupling medium is the multiport memory which increases bandwidth since it allows simultaneous access to all data stored in such a memory. Although its bandwidth is high, its ability to cover distances is poor since both properties tend to be contradictory.

Besides these classic coupling techniques, the so called single- or multistage interconnection networks have gained significance in multiprocessors, multi computers and distributed systems because they allow both, high distances and bandwidth. Their historical roots lie in the switching circuit fabrics of tele phone exchange systems where they have been used for decades. The very first distinction a classification has to make is to enumerate the mentioned basic categories of coupling techniques:

Basic Coupling Techniques = {Shift Register | Parallel Bus | Local- or Wide Area Serial Bus | Multiport Memory | Interconnection Network}

Each of these techniques can be characterized by three physical parameters: 1.) the speed of the coupling medium, i.e., its bandwidth, 2.) the maximum distance which can be covered, 3.) the maximum delay between sending and receiving of data. Of course, the latter is related to the former by the signal propagation time that is limited by the speed of light. For small distances, up to several tens of centimeters, this latency is mainly determined by circuitry delay and not by the signal propagation time. Therefore, we make a distinction between distance and latency. In the following, examples are given how these three physical factors influence all applications that are based on interconnection networks.

2.5.3.2 Speed, Distance and Latency

1.) Bandwidth and latency time impact the architecture of parallel computers: The transmission speed has to match the computing speed because many appli cations need both, computation and communication in a balanced combination otherwise the computing nodes can't be used efficiently. 2.) The factors speed and latency influence the granularity of parallel programs of multicomputers: A small latency makes it worth to transfer small junks of data while high latencies forbid this for efficiency reasons. The same holds for the transmission speed. 3.) Distance and speed determine the performance of distributed systems that are coupled by means of local- or wide area serial busses (LANs or WANs). For propagation delay time reasons it makes no sense to couple largely remote computers to solve a numerical problem for example. 4.) In multimedia applications speed and latency of the interconnection network decide whether data can be transferred in real time. Nets for wide-, metropolitan-, or local-area applications are approaching each other in terms of the physical parameters and their technical structure. In the ATM definition for example, the differences between WAN-, or LAN based systems or applications between single crates have been almost diminished. Although the transmission speed is converging, the latencies in the networks are not since they directly reflect the finite speed of light. For a classification of interconnection networks it is reasonable to categorize nets for different applications in a single scheme because they have many features in common. This holds mainly for the so called architectural (or technical) parameters. The second distinction in the classification scheme is to differentiate between physical and technical parameters as the principal factors of a network:

Principal Factors = {Physical Parameters | Technical Parameters} Physical Parameters

= {Speed | Distance | Latency}

In the following, the technical parameters of interconnection networks are ca tegorized in more detail.

2.5.3.3 Categorization of Technical Net Parameters

The categorization of the technical parameters is made hierarchically. In the first level of the scheme the principal components are regarded. These components are:

Technical Parameters = { Number of Switching Stages | Topology | Number of Hierarchies | Number of Parallel Nets | Feed Back Connections | Way of Transmission | Connection Capability | Connection Plurality | Input Stream | Flow Control | Path Finding | Routing Methods | Buffering | Transmission Directions | Data Paths | Reliability | Protocol | Scalability | Real-Time Features }

a) Number of Switching Stages

The first parameter of the first level denotes the number of switching stages. It classifies networks in static and dynamic. A static net has no special switching stages. Only computing nodes with routing functions inside are present. A dy namic net instead, consists mainly of one or more switching stages which are not used for computing. The computing nodes are connected to the outside input-/output ports of this network type.

Number of Switching Stages = {no switching stages (static) | multiple stages (dynamic)}

There are many examples of static and dynamic networks illustrating that ordering scheme in deeper levels of classification:

static = { chain | ring | star | grid | torus | hypercube | hypertree | Moore graph | reduced Kneser graph |...}
dynamic= {Crossbar | Clos | Benes Class | Banyan Class | ...}
Benes Class = { Benes net | Lee net | double baseline net | shuffle exchange net | double Flip net | double indirect binary n-cube| ...}
Banyan Class = {Banyan net | butterfly net | Omega | Xomega | Flip | Baseline | indirect binary n-cube | Delta | ...}

In Figure 2.51 and Figure 2.52 on page 120 some single- and multistage networks are given.

In static networks in general several intermediate nodes are located between a transmitting and receiving node. In a ring or grid for example at most (N/2)-2 or 2*sqrt(N)-2 resp. intermediate nodes are present. The distance in a dynamic net instead, is always one because all nodes are directly connected together with the ports of the separate network. In static nets intermediate coupling nodes are normally time consuming to pass (several μs to ms). Switching stages in dynamic nets instead, can be built to be much faster (several ns). In a crossbar net for example only one switching stage has to be passed for each connection. A Clos net has three switching stages and a Benes net consists of 2logN-1 stages (N is the number of input-/output ports). An important distinction has to be made between the Benes and the Banyan classes of multistage nets. Banyans are defined as single path multistage networks where only one path is possible between any input-output pair while Benes nets have multiple paths. It was shown by graph- and group theoretical methods that all nets of one class are topological equivalent. Normally,

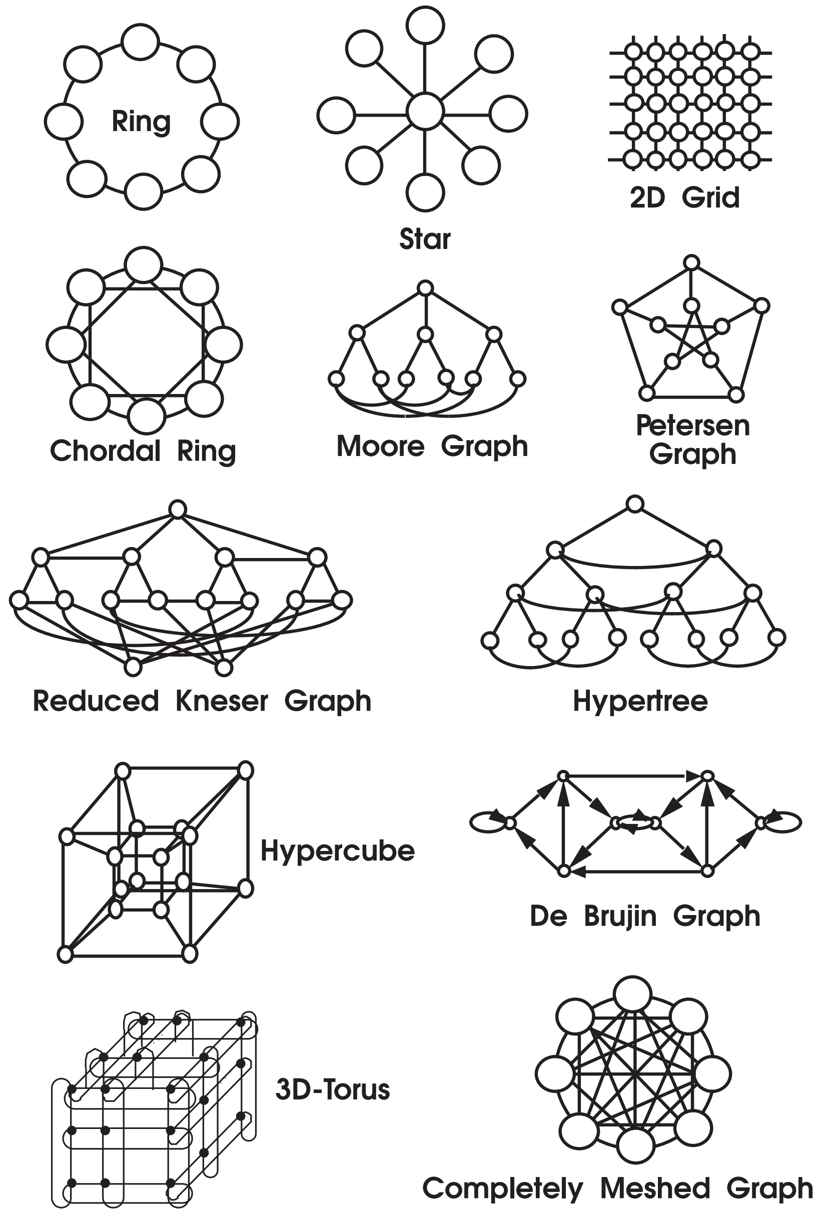

Figure 2.51: Examples of single interconnection networks. (computing nodes are inside the net)

a Benes type net can be constructed by arranging two Banyans end-to-end. Note, that a series of two Banyans of the same topology is not the same as a series of two other Banyans although the single components are equivalent.

b) Topology

In Figure 2.53 on page 121 and Figure 2.54 on page 122 symmetric and asymmetric topologies are shown that are used in multistage networks. The corresponding single stage net graph is also shown for four computing nodes.

The interconnection topology is an important classification parameter because it defines the cabling scheme of nodes or stages in a net and many of their technical properties.

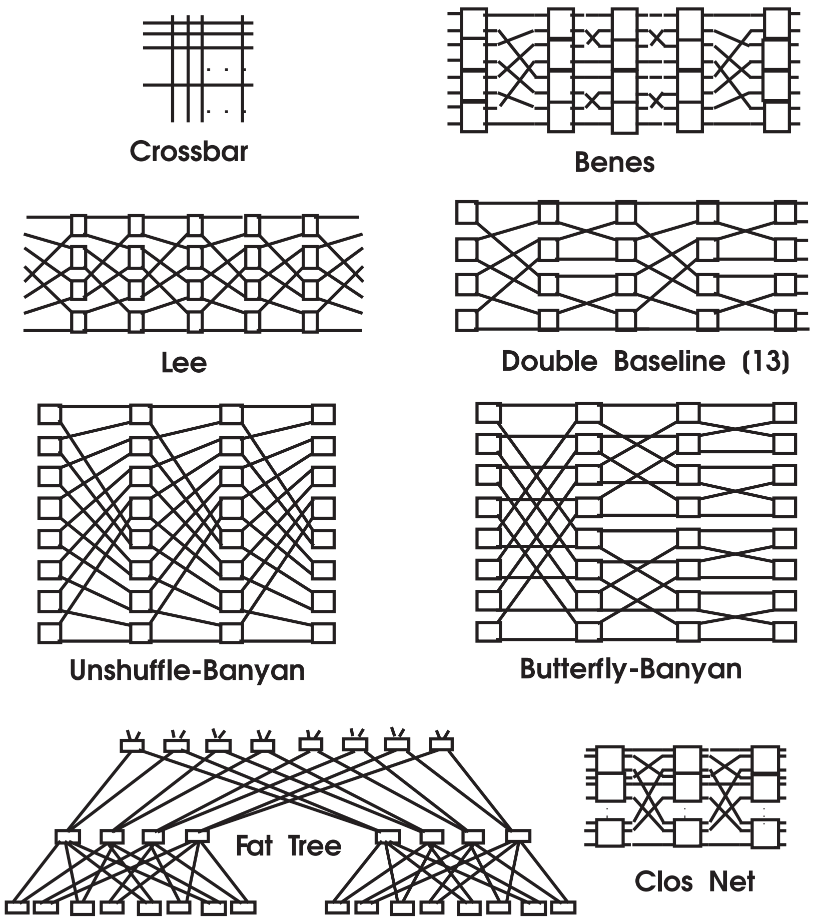

Figure 2.52: Examples of multi stage interconnection networks. (computing nodes are outside)

Topology = {regular | not regular }
regular = {symmetric | asymmetric}
symetric = {perfect shuffle | butterfly | reversal...}
asymmetric = { delta permutation | ... }

Within the category of static networks the diameter of the net graph is an important measure. Normally, the maximum- and the mean distance of all nodes in a static graph are used as metric values. For cost reasons, also the number of links of each node is important. This number is equivalent to the edge count of a (d,k)-graph where d is the degree and k is the distance of the graph. All properties of graphs can be used to distinct net topologies.

c) Number of Hierarchical Steps

Many networks in computers or telephone switching systems are organized hierarchically because that reflects the 'principle of locality' inherent to the communication patterns of those applications. Hierarchical structures are more economical in those cases where 'nearest neighbor' or other local communica tions often take place. Examples for a two layer hierarchy are the MANNA computer of the GMD FIRST (Berlin, Germany) or the 64 Processor CM5 machine from Thinking Machines Corp. An IBM SP2 computer

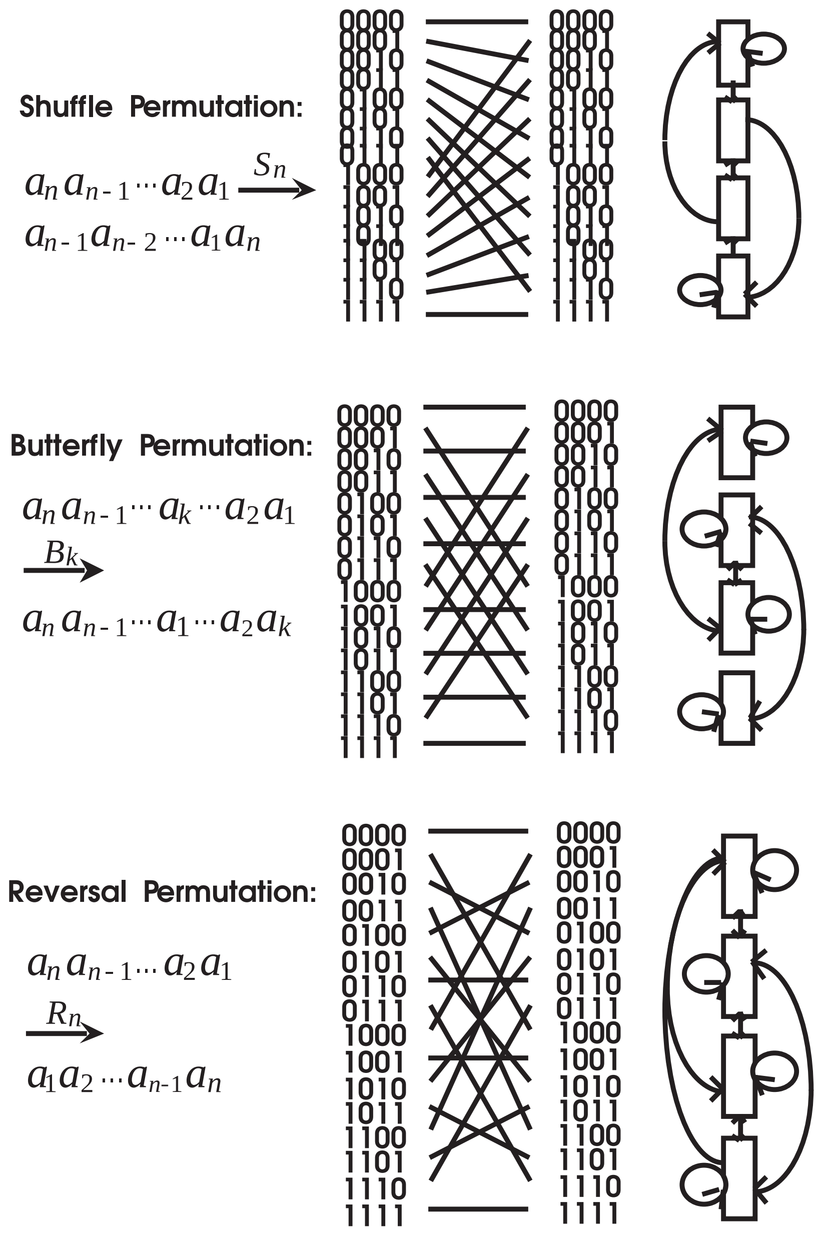
Shuffle Permutation:
$a_n a_{n-1} \cdots a_2 a_1 \xrightarrow{S_n}$
$a_{n-1} a_{n-2} \cdots a_1 a_n$
Butterfly Permutation:
$a_n a_{n-1} \cdots a_k \cdots a_2 a_1$
$\xrightarrow{B_k}$
$a_n a_{n-1} \cdots a_1 \cdots a_2 a_k$
Reversal Permutation:
$a_n a_{n-1} \cdots a_2 a_1$
$\xrightarrow{R_n}$
$a_1 a_2 \cdots a_{n-1} a_n$
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

Figure 2.53: Three examples of regular and symmetric topologies in multistage networks.

with 128 processors exhibits three layers of hierarchy: 1.) the switch-chip layer, 2.) the switchboard layer, and 3.) the inter-switch-board layer.

Number of Hierarchical Steps = { 1 | 2 | 3 | ... }
1 layer = {nohierarchy}
2 layers = { e.g., the 144 processors MANNA | 64 Proc. CM5 | ... }
3 layers = { e.g., the 128 proc. IBM SP2 |...}

In Figure 2.55 , Figure 2.56 on page 123, and Figure 2.57 on page 124 some hierarchical static and dynamic nets are given.

d) Number of Parallel Nets

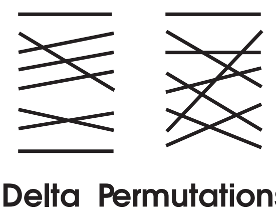

Figure 2.54: Two examples of regular and asymmetric topologies in Delta networks.

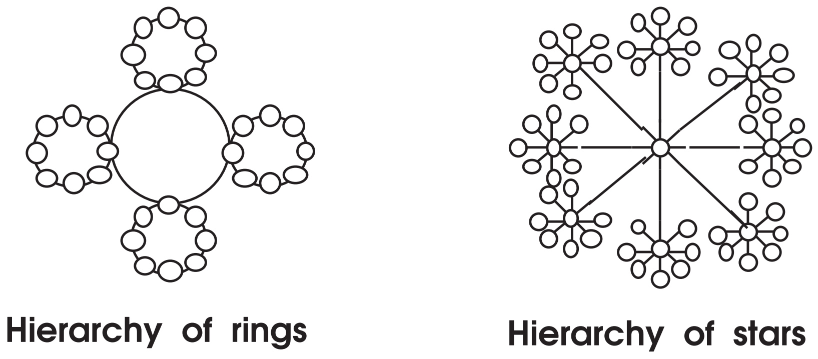

Figure 2.55: Two Examples of hierarchical static nets with two layers.

It is also possible to connect one and the same computing nodes by means of two or more (different) networks. In Figure 2.58 on page 124 an example for two superimposed networks is given. (nodes are drawn twice).

One net is for multicast and the other for inverse multicast functions. They are utilized for synchronization of parallel programs in the Cray T3D computer. The reasons to use parallel nets are manifold: 1.) for fault tolerance, 2.) for performing special network functions like multicast- and inverse multicast on dedicated nets, 3.) to increase communication bandwidth. We have therefore:

Number of Parallel Nets = { 0 | 1 | 2 | 3 | ... }

e) Feed Back

In some cases output links from a multistage interconnection network are feed back to its inputs to allow for a rerouting of data. A multistage shuffle- exchange net for example can be emulated by using one shuffle-exchange stage with folded outputs. Data can recirculate the closed loop logN times to reach their final destinations.

Feed Back = {not folded | folded }

If N, M, and Q are the numbers for the input-, output-, and feed back lines resp. we have the following representation (Figure 2.59 on page 124) for such a net.

f) Way of Transmission

There are three basic ways information can be passed from a transmitter to a receiver: 1.) In circuit switching a permanent path between both is established which lasts for the whole duration of the communication. Data travelling on that path make the communication partners feel as if they were directly connected. 2.) In packet switching data is chopped

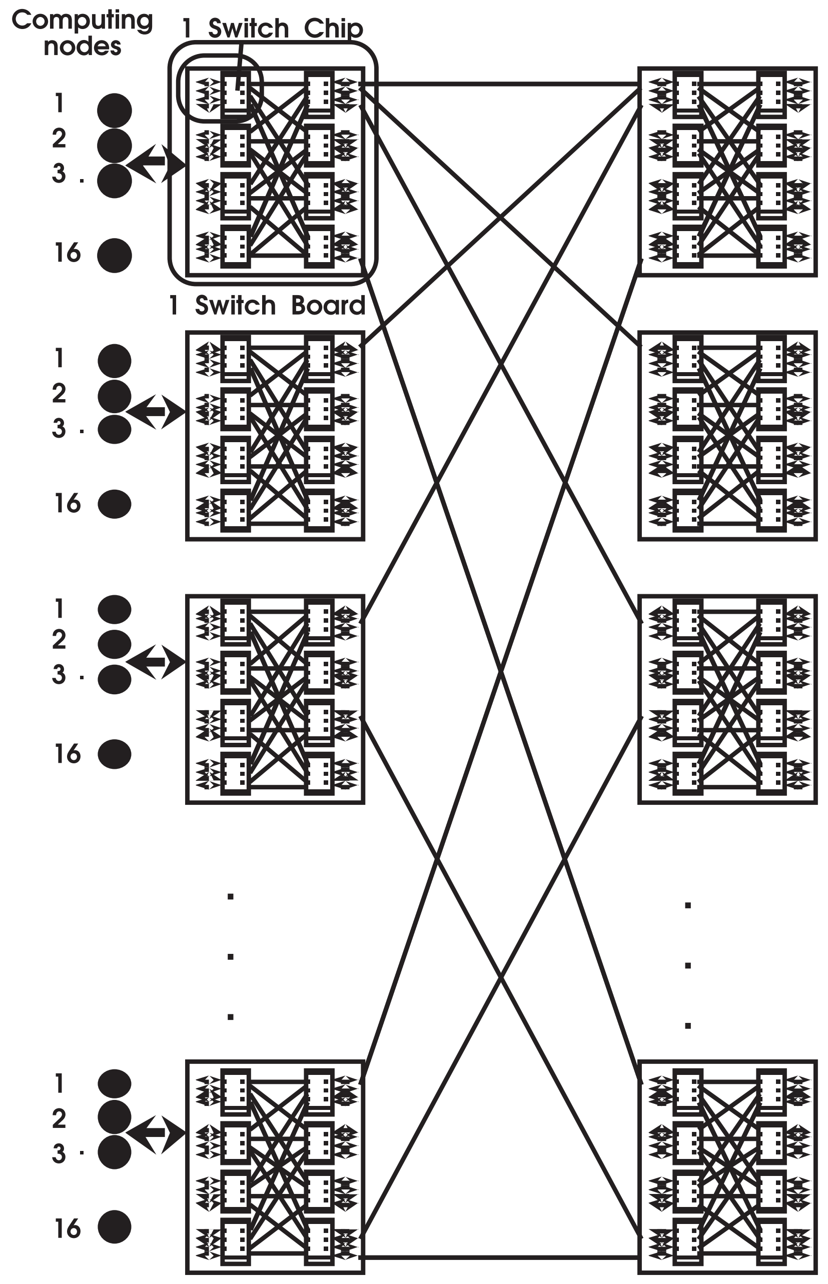

Figure 2.56: Example of a three layer dynamic network (IBM SP2).

into smaller units called packets which contain additional information like address headers and packet termination tokens. The advantage of this technique is that physical channels are not permanently occupied. By multiplexing packets from different trans mitters on the same physical channel the communication resources are much better utilized. 3.) In message switching the junk size is not a packet but a whole message. Only for the time it takes to transfer the message a physical channel is exclusively reserved. Message switching is therefore a mixture between circuit- and packet switching. We have:

Way of Transmission = {circuit switching | packet switching | message switching}

g) Connection Capability

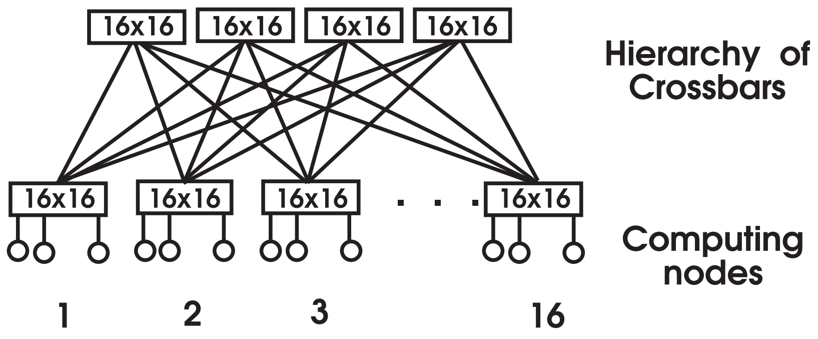

Figure 2.57: Example of a two layer dynamic network.

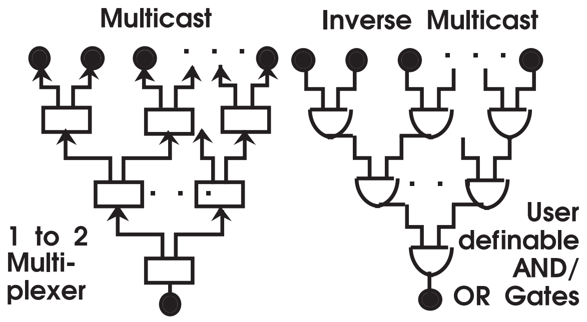

Figure 2.58: Two different binary trees for the same set of nodes.

Under connection capability there can be understood whether a net is strictly non-blocking, or non-blocking only by rearranging internal paths or rerouting data packets, or whether it blocks in general. Blocking is defined as a situation where two paths or two packets want to occupy the same net resource at the same time. It can happen for example that two packets on different inputs of a stage or a router want to get to the same output simultaneously. Then, one of them is blocked and has to wait. A Clos net, e.g., can be constructed as strictly non-blocking, or with less effort, i.e., costs, as non-blocking by rearranging internal paths. A Benes net is of the non-blocking type because data packets can always be rerouted while a Banyan net blocks most of the time. The connection capability is determined by the internal redundancy a net has. A strictly non- blocking network for example is highly redundant since it can always guarantee that there is at least one free path from any input to any output. Of course, such a net is also most expensive. In contrast to this are the nets of the Banyan family where by definition exactly one path exists from input to output but not for all inputs at the same time. There-

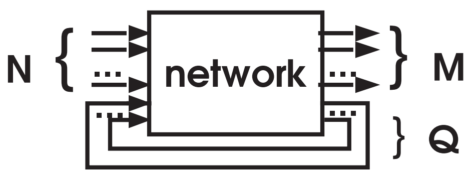

Figure 2.59: Folding part or all the output lines back to the input.

fore, Banyan nets have no redundancy at all and can be realized cheaply. According to their minimal property their latency is also minimal if a packet is not blocked by chance. But experience shows that in single- or multistage interconnection networks most of the packets have to be buffered at least one time on their way through the net. In networks of the Benes family at each switching stage two paths exist from input to output and, together with a centralized routing scheme, it was shown that always paths can be laid such that blocking does not occur for each given connection permutation. In Benes nets with decentralized routing instead, this can not be accomplished due to the lack of 'global' information about the connection wishes. But it is possible to minimize buffering by a sophisticated adaptive routing scheme which exploits the net redundancy. To summarize, we have:

Connection Capability = { non blocking | rearrangeable non blocking | blocking }

h) Connection Plurality

For practical reasons the connection plurality is of importance since it defines the 'richness' of connections a net can realize. Beside the well-known point-to- point couplings also multicasts, broadcasts, inverse multicasts and -broadcasts, as well as conference connections are important. In an inverse multicast for example data is gathered from many computing nodes and concentrated in one node which in turn can perform reduction operations like computing the minimum or maximum of the input values. In conference connections a subset of nodes is coupled together by allowing each transmitter in the group a multicast to the others in the group. Of course, already simple multicasts and broadcasts support the writing of parallel programs because those functions don't have to be coded in software. They are executed much faster by hardware from the net.

Connection Plurality = {point-to-point | multicast (scatter)/broadcast | inverse multicast (gather)/broadcast | conference }

i) Input Stream

According to the definition of ATM input streams with different phase relations and transmission speeds can be processed simultaneously in one ATM network. This means that traffic with low baud rates like telephone speech and high speed multimedia applications like video conferences can be input to a common network at the same time and the net has to process them. In Single Instruction Multiple Data (SIMD) computers instead, information production and data exchange is phase coupled and thus synchronous. Networks for SIMD machines have therefore to be operated synchronously. For the network it is more easy to accept synchronous data at the input ports while for transmitting purposes asynchronous transfers are easier to implement. We have to make a distinction between both operating modes:

Input Stream = { synchronous | asynchronous }

j) Flow Control

If a receiver, an intermediate computing-, or routing node, or a physical channel is temporarily occupied the producer of data has to be informed. Then, generating new data is stopped or buffers have to be allocated for intermediate data storage. Both actions can only take place if the occupied unit is able to signal this situation to the transmitting unit or the buffers. I.e. data in reverse direction has to flow from the destination to the source by means of flow control signal wires or by special tokens in packet form. Therefore, flow control is an inevitable method for networks where blocking can occur. It is up to the network designing engineer whether flow control is added and how it is implemented.

Flow Control = { yes | no }

k) Deadlock Behavior

Closely related to flow control is the deadlock problem. If flow control is en abled many nets tend to deadlock under certain circumstances. Consider the case where the buffers of a transmitter and its corresponding receiver are full with data destined for each other (Figure 2.60) and both want to get rid of them. Transmitter and receiver can't proceed since no extra buffer space is available. This very simple situation can occur arbitrarily often and may also happen in a ring like structure where more than two nodes are involved (Figure 2.61). This means that full buffers are always deadlock prone. In Figure 2.62 an other case of circular wait with four participants is depicted. In this case a deadlock is provoked by the fact that the communication channel between two adjacent nodes is occupied by a transfer of a predecessor node. If a closed loop of occupied channels can be formed nobody can start with the transfer. Multiplexing many virtual channels onto one physical channel solves this problem. An other solution is the introduction of 'virtual nets' to exclude any closed loops.

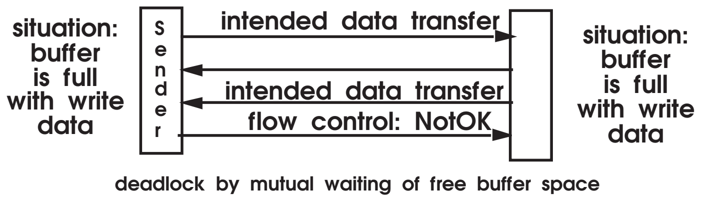
situation: buffer is full with write data
sender
intended data transfer
intended data transfer
flow control: NotOK
situation: buffer is full with write data
deadlock by mutual waiting of free buffer space

Figure 2.60: Deadlock by mutual waiting for free buffer space.

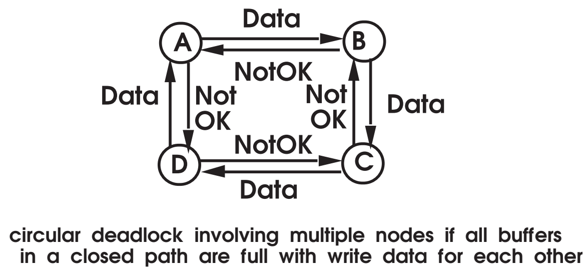
Data
A
B
NotOK
Data
Not OK
Not OK
Data
NotOK
D
C
Data
circular deadlock involving multiple nodes if all buffers in a closed path are full with write data for each other

Figure 2.61: Circular deadlock by occupied buffers.

The network designer has to be aware whether the network graph allows the formation of closed loops and whether the application is sensitive to deadlocks. Deadlock Behavior is an important factor in many practical applications and a classification scheme has to reflect it:

Deadlock Behaviour = { critical | non critical }

l) Path Finding

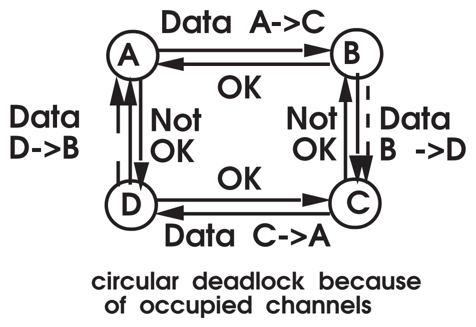

Figure 2.62: Circular deadlock by occupied channels.

In each network it is a concern to find the 'right' path from source to destina tion. The right path can be the shortest or the fastest which is not identical. The path finding process can be performed by a central instance or by decentral routers that decide locally where to go. Today, local path finding is preferred because this doesn't imply any bottleneck and is thus scalable. Additionally, we have the choice to route a data packet or to lay a transmission line determinis tic or adaptively. The adaptive routing method promises a higher throughput because channels with a low traffic load can be chosen to transmit a data packet thereby eliminating traffic imbalances and hot spots. But such methods are load dependent and therefore not deterministic. This is not tolerable in real-time systems where latencies must be predictable. A simple and well-known decen tralized routing scheme is the so called 'destination based' method where the output port of a router is chosen by the address field of the data packet. In source based routing mechanisms the transmitter has to know about the difference in position between him and the destination. From this information it computes routing decisions in advance quite similar to the manner "go straight, turn left, ...". The most popular centralized path finding scheme is the 'looping-, or in-out routing' which was first used for the Benes net.

Path Finding = {centralized | decentralized | deterministic | non deterministic | adaptive | random}

deterministic = {source based | destination based | looping routing | ...}

m) Routing Method

The term 'routing' is normally used for how data is physically transferred from node to node in a static network or from stage to stage in a dynamic net and is not used for the previously described path finding methods although this would be more appropriate. The three most popular 'routing' methods are store and forward-, virtual cut trough-, and wormhole routing.

Routing Method = {store and forward | virtual cut through | wormhole | ...}

In store and forward routing data packets are completely stored at each inter mediate node before they are forwarded to a neighboring node or stage. If a node or stage is blocked all data packets destined for that node have to be stored in the predecessor node. Therefore, large buffers must be available. In wormhole routing only packet addresses are inspected when they pass a node. If a node or channel blocks the whole caravan of data packets has to stop. Thus, only few temporary storage has to be present. Additionally, passing a node can be implemented to be very fast because no storing is required. In virtual cut through routing a mixture of both methods is applied. In the non blocking case virtual cut through routing is identical to wormhole routing. In the blocking case medium sized buffers can store packets for a certain period of time before the whole packet caravan must be finally stopped.

n) Buffering

In telecommunication it is of much interest where buffers have to be placed in a net, how large their sizes must be, and which queuing discipline they should obey because large and/or fast buffers are more expensive than the whole transmission line. On the other hand, buffer positions and sizes as well as their scheduling schemes determine throughput and latency of the network. Buffers can be located at the inputs or outputs of the net, or at each node or stage, or central buffers can be introduced which are shared by a group of nodes. To determine the buffer size for a certain probability of data loss is subject to systems- and queuing theory. Often, no analytic results are obtainable. Beside the well known first-in-first-out (FIFO) scheduling strategy and its reverse (LIFO) other disciplines can be of value, like priority -, preemptive-, round robin-, or random service. More sophisticated queueing disciplins are shortest latency time first (SLTF), and the last-come-first-served preemtive resume (LCFSPR) technique. We have listed them below:

Buffering = {buffered | unbuffered }
buffered = { buffer location | buffer queueing disciplin}
buffer location = {input buffer | output buffer | central buffer | switch buffer | combinations}
buffer queueing discipline = {FIFO | LIFO | priority | preemtive | round robin | random | SLTF | LCFSPR | ... }

o) Transmission Directions

This parameter simply signifies whether there is a preferred direction for in formation flow on a communication channel. In high speed applications it is much easier to implement unidirectional channels than bidirectional ones due to impedance, termination and crosstalk problems. There is always a certain DC component in the signal current on a link, even though it is constant which differential signalling. If the link direction were to be changed the line drivers at one end would have to be turned off while the signal travels to the other end where the drivers would turn on. Due to this process the direction of the aver age DC current would have to reverse. This reversal causes severe electrical noise and ground-shift problems. In unidirectional links DC components are not zero but constant thus causing little trouble. Because the question whether a net consists of unidirectional or bidirectional channels is of significance we have to differentiate the transmission directions.

Transmission Directions = {unidirectional | bidirectional}

Even if all channels are unidirectional a bidirectional network can be establi shed as Figure 2.63 shows for the case of 4 inputs and outputs.

p) Data Paths

The width of the data paths of the internal channels and the external ports determines speed and costs of the net. The data path width has to reflect the environment where

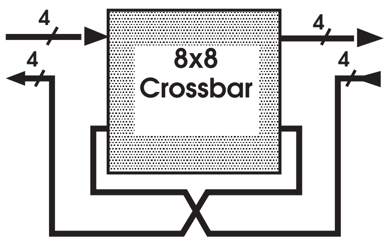

Figure 2.63: An 8x8 crossbar used as a 4x4 bidirectional switch.

the net will be installed. In wide-, metropolitan-, or local- area networks 1 bit wide data paths are standard for cost reasons. In multipro cessor- or multicomputer environments 8, 16, or 32 bit data transmission is desirable for speed reasons. With the CMOS technology serial ports are limited to speeds of up to several hundreds of MBit/s while GaAs chips allow to reach the GBit/s range. Standards for serial ports propose speeds of 155 MBit/s, or 622 MBit/s resp. on glassfiber links for ATM nets and 1 Gbit/s for Scalable Coherent Interface (SCI) networks. The parallel port version of SCI allows 1GByte/s transmission speed on 16 Bit input/output lines.

Data Paths = { serial | parallel }
parallel = { 8 Bit | 16 Bit | 32 Bit | ... }

q) Reliability

In an industrial environment reliability is of superior importance. Especially, in the area of real-time control of machines, power stations, and nuclear power plants where personnel safety is mandatory high efforts have to make to ensure reliable data transmission. The control systems of those applications are gene rally distributed because this makes it easier to manage the complexity of such problems. But task distribution requires usage of communication nets which in turn have to be highly reliable. Of course, in military and space applications reliable nets are a must. In multistage networks of the Banyan type reliability can be achieved by introducing extra stages giving some redundancy. The same holds for nets of the Benes class. In static networks redundancy is always pre sent in the network graph since each node has normally more than one edge, i.e., port to its neighbours.

Reliability = {fault recognizing | fault tolerant | nor neither}

r) Protocols or Data Formats

In circuit switched networks no special protocol is necessary because a direct connection exists between a sender and a receiver. In packet- or message swit ching networks instead, data is assembled into smaller units and a flow control mechanism is established. Therefore, protocols and data formats have to be arranged between the communication partners. In Figure 2.64 we see the hierarchical structure of this data exchange.
The 'atoms' of the messages are the so called physical transfer units (phits) which are 1- or 2 bytes in size, e.g., at the IBM SP2 and Cray T3D resp. A synonym for phits is flow control digits (flits). The data format of a packet is shown in Figure 2.65 for the example of a transputer T9000. The header section of the packet is variable in length according to the size of the net.

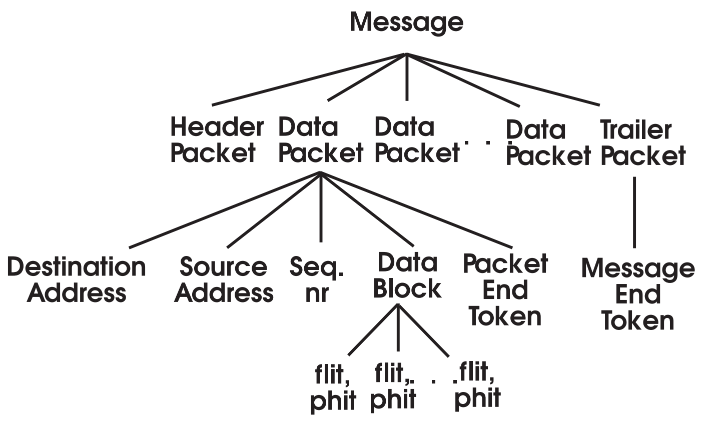

Figure 2.64: Hierarchical structure of a message in a communication network with packet switching.

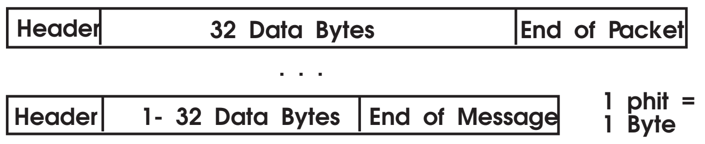

Figure 2.65: Data packet format of the transputer T9000.

In Figure 2.66 and Figure 2.67 we see the formats of ATM and SCI, two communication standards for wide- and local area-, or short range applications resp.

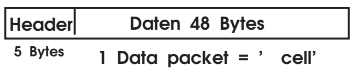

Figure 2.66: Data packet format of ATM.

The question which data format the net uses is reflected in the classification scheme.

Protocol = {none | Ethernet Format | FDDI Format | ATM | SCI | Transputer Format...}

s) Scalability

The trend in parallel computing is to use scalable machines. This allows to preserve investments by starting with small machines that can be upgraded according to the needs. Scalability requires that no bottlenecks in the architecture of the net, its protocols and routing methods are present. The goal is that the size of the network can be arbitrarily increased, preferably in small junks and that already existing parts can be reused in the upgraded version. In multistage interconnection networks the complexity of the net grows with O(NlogN) where N is the number of network ports. Although grower faster than linear still a good efficiency is provided comparable with the O(NlogN) increase of efficient numerical algorithms, like the Fast Fourier Transform. So multistage networks can be scalable in principle.

Scalability = {scalable | not scalable}

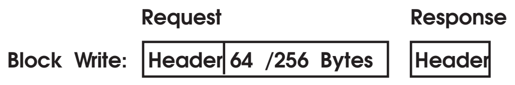

Figure 2.67: Data packet format of SCI.

t) Real Time Feature

Real-time networks guarantee a maximum service time at any port for any time. This can't be taken for granted since queuing theory says that each service center has a saturation point from which on response time is significantly increased. Additionally, in a net with adaptive routing the latency is not deterministic. Real-time networks are mandatory in distributed control sys tems, in robotics, military-, and space applications, and in multimedia systems. Thus, a distinction whether a net is real-time capabable or not has to be made.

Real Time Feature = {real time capable | not real time capable}

2.5.3.4 Summary and Conclusions

A classification scheme was presented that is considered to cover the most important factors of interconnection networks. The scheme can be applied for networks for telecommunication as well as for parallel processing. The classi fication parameters were chosen to be disjoint so that combinations of their values can be composed arbitrarily. Each principal component of the scheme was decomposed and explained in detail by means of examples. The principal components are: number of switching stages, topology, number of hierarchies, number of parallel nets, feed back connections, way of transmission, connection capability, connection plurality, input stream, flow control, path finding, routing methods, buffering, transmission directions, data paths, reliability, protocols, scalability, and real-time features. The purpose of this scheme is to give an overview on coupling techniques with emphasis on interconnection networks as well as to structure the knowledge in this field. Table 1 summarizes the Backus-Naur notation of the classification.

Basic Coupling Techniques = {Shift Register | Parallel Bus | Local- or Wide Area Serial Bus | Multiport Memory | Interconnection Network}

Principal Factors = {Physical Parameters | Technical Parameters}
Physical Parameters = {Speed | Distance | Latency}
Technical Parameters = { Number of Switching Stages | Topology | Number of Hierarchies | Number of Parallel Nets | Feed Back Connections | Way of Transmission | Connection Capability | Connection Plurality | Input Stream | Flow Control | Path Finding | Routing Methods | Buffering | Transmission Directions | Data Paths | Reliability | Protocol | Scalability | Real-Time Features }

Number of Switching Stages = {no switching stages (static) | multiple stages (dynamic)}

static = { chain | ring | star | grid | torus | hypercube | hypertree | Moore graph | reduced Kneser graph |...}
dynamic={Crossbar | Clos | Benes Class | Banyan Class | ...}
Benes Class = { Benes net | Lee net | double baseline net | shuffle exchange net | double Flip net | double indirect binary n-cube| ...}

Banyan Class = {Banyan net | butterfly net | Omega | Xomega | Flip | Baseline | indirect binary n-cube | Delta | ...}

Topology = {regular | not regular }
regular = {symmetric | asymmetric}
symetric = {perfect shuffle | butterfly | reversal...}
asymmetric = { delta permutation | ... }

Number of Hierarchical Steps = { 1 | 2 | 3 | ... }
1 layer = {no hierarchy}
2 layers = { e.g., the 144 processors MANNA | 64 Proc. CM5 | ... }
3 layers = { e.g., the 128 proc. IBM SP2 |...}

Number of Parallel Nets = { 0 | 1 | 2 | 3 | ... }

Feed Back = {not folded | folded }

Way of Transmission = {circuit switching | packet switching | message switching}

Connection Capability = { non blocking | rearrangeable non blocking | blocking }

Connection Plurality = {point-to-point | multicast (scatter)/broadcast | inverse multicast (gather)/broadcast | conference }

Input Stream = { synchronous | asynchronous }

Flow Control = { yes | no }

Deadlock Behaviour = { critical | non critical }

Path Finding = {centralized | decentralized | deterministic | non deterministic | adaptive | random} deterministic = {source based | destination based | looping routing | ...}

Routing Method = {store and forward | virtual cut through | wormhole | ...}

Buffering = {buffered | unbuffered }
buffered = { buffer location | buffer queueing disciplin}
buffer location = {input buffer | output buffer | central buffer | switch buffer |combinations}
buffer queueing discipline = {FIFO | LIFO | priority | preemtive | round robin | random | SLTF | LCFSPR | ... }

Transmission Directions = {unidirectional | bidirectional}

Data Paths = { serial | parallel }
parallel = { 8 Bit | 16 Bit | 32 Bit | ... }

Reliability = {fault recognizing | fault tolerant | nor neither}

Protocol = {none | Ethernet Format | FDDI Format | ATM | SCI | Transputer Format...}

Scalability = {scalable | not scalable}

Real Time Feature = {real time able | not real time able}

Table 1: The complete classification of interconnection networks.

References

[HB85] *K. Hwang and F. A. Briggs.*
"Computer Architecture and Parallel Processing, pp. 332-354, 481-508".
McGraw-Hill (1985).

[HJ81] *Hockney and Jesshope.*
"Parallel Computers".
Adam Hilger, Bristol (1981).

[HP93] *P. G. Harrison and N. M. Patel.*
"Performance Modelling of Communication Networks and Computer Architectures".
Addison-Wesley (1993).

[Hwa93] *K. Hwang.*
"Advanced Computer Architecture, pp. 75-96, 331-347".
McGraw-Hill (1993).

[Lei92] *F. Thomas Leighton.*
"Introduction to Parallel Algorithms & Architectures: Arrays, Trees & Hypercubes".
Morgan Kaufmann (1992).

[MTW93] *M. D. May, P.W. Thompson, and P.H. Welch.*
"Networks, Routers & Transputers".
IOS Press, Amsterdam (1993).

[Ric] *H. Richter.*
Modular Interconnection Network.
U.S. Patent Nr. 5,175,539, Dec. 1992.

[Sch87] *M. Schwartz.*
"Telecommunication Networks, pp. 483-552".
Addison- Wesley (1987).

[Sie90] *H. J. Siegel.*
"Interconection Networks for Large-Scale Parallel Processing".
McGraw-Hill (1990).

[SY94] *I. D. Sherson and A. S. Youssef.*
"Interconnection Networks for High-Performance Computers".
IEEE Computer Society Press (1994).

[VR93] *A. Varma and C. S. Raghavendra.*
"Interconnection Networks for Multiprocessors and Multicomputers".
IEEE Computer Society Press (1993).

[WF85] *C. Wu and T. Feng.*
"Interconnection Networks for Parallel and Distributed Processing".
IEEE Computer Society Press (1985).

[Wu91] *C. L. Wu.*
Tutorial on System Integration with Interconnection Networks.
In "Int.Conf. on Par. Proc.", St. Charles,ILL. (August 1991).