7 Performance of C104 Networks

The use of VLSI technology for specialised routing chips makes the construction of high-bandwidth, low-latency networks possible. One such chip is the IMS C104 packet routing chip, described in chapter 3. This can be used to build a variety of communication networks.

In this chapter, interconnection networks are characterized by their throughput and delay. Three families of topology are investigated, and the throughput and delay are examined as the size of the network varies. Using deterministic routing (in which the same route is always used between source and destination), random traffic patterns and systematic traffic patterns are investigated on each of the networks. The results show that on each of the families examined, there is a systematic traffic pattern which severely affects the throughput of the network, and that this degradation is more severe for the larger networks. The use of universal routing, where an amount of random behavior is introduced, overcomes this problem and provides the scalability inherent to the network structure. This is also shown to be an efficient use of the available network links.

An important factor in network performance is the predictability of the time it will take a packet to reach its destination. Deterministic routing is shown to give widely varying packet completion times with variation of the traffic pattern in the network. Universal routing is shown to remove this effect, with the time taken for a packet to reach its destination being stabilized.

In the following investigation, we have separated issues of protocol overhead, such as flow control, from issues of network performance.

7.1 The C104 switch

The C104 is a packet routing chip. The use of VLSI to create such a chip means that routing is fast, and the flexibility of the C104 ensures that the chip can be used in many situations. The C104 contains a 32-way crossbar switch, in which all of the 32 inputs can be routed simultaneously to the 32 outputs. Routing delays are minimized by the use of wormhole routing, in which a packet can start to be output from a switch whilst it is still being input. The C104 is described in more detail in Chapter 3.

A packet arriving at a switch is routed according to its header. If the required outbound link is available, the packet utilizes the link. However, if the link required is currently in use, the packet will be blocked. The tail of the packet may now start to catch up with the head. If there is enough buffering, the whole packet may be taken into a buffer, waiting to have access to its required output. Therefore if the network is very busy, the performance will approximate to the performance of a store-and-forward network. The C104 provides roughly one packet of inbound buffering, and one packet of outbound buffering on each link, for packets of a small average size, such as those used by the virtual channel protocol. The simulations reported in the chapter use a model with precisely one packet of buffering on each input and one on each output.

The C104 supports universal routing, which requires each packet to be sent to a randomly chosen intermediate node before it travels to its real destination. Any of the links on the C104 can be set to create a random header for each inbound packet on that link. At the randomly chosen intermediate node, this random header is deleted, leaving the original header to route the message to its real destination.

All routing, header creation and deletion is performed on a per link basis. There is no shared resource within the C104. This has the effect of making the links of the network the shared resources, rather than the nodes of the network.

7.2 Networks and Routing Algorithms

In a communication network connecting p terminals, we can realistically expect the distance a packet will travel to increase with $log(p)$. Consequently, if we wish to maintain throughput per terminal, the number of packets in flight from each terminal will scale with $log(p)$. Therefore network capacity required for each terminal will scale with $log(p)$. The total capacity of a network with p terminals must therefore scale as $p \times log(p)$. One structure which achieves this is the hypercube or binary n–cube. Another structure is the (indirect) butterfly network, which has constant degree. Conversely, the two–dimensional grid and indirect multistage networks do not maintain throughput per node as the network scales.

Three topologies are considered. The first structure is the binary n-cube. The second structure is the two-dimensional grid, which is appealing practically. The last structure is the indirect multistage network.

In a binary n-cube, node i is connected to node j in dimension k if the binary representation of i and j differ only in the k^{th} bit. The n-dimensional cube has N=2^n nodes, diameter n and uses n links per network node for network connections.

A grid is a 2-dimensional array of routing chips. If the network is drawn onto integer axes, there is a router at each of the intersections, and links in both the x and the y directions. Only links internal to the grid are used, since, although it is possible to construct toroidal networks using the C104, the number of links used to ‘wrap around’ must be doubled to avoid the possibility of deadlock. This contrasts with the appealing simplicity of the grid, and so such networks are not studied here.

The indirect multistage networks considered in this chapter provide a low cost switch for small networks, and make economical use of the C104 switches for large networks. An example of an indirect multistage network, with 512 terminals, is shown in figure 7.1.

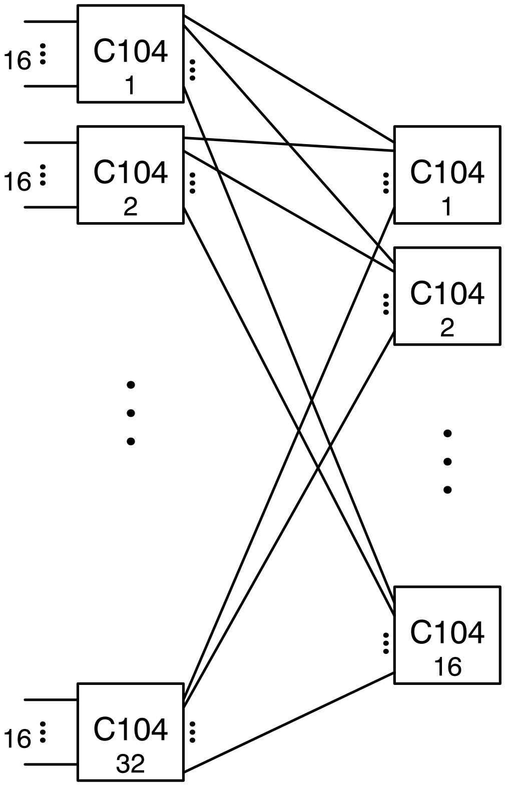

Figure 7.1 A 512–way multistage network

There are 16 external links on each C104 in the left hand column, and there are 32 switches in the column. There are 16 switches in the right hand column. Indirect multistage networks can also be built using 8 links to the left of the left hand column, and 24 links to the right, providing greater throughput per terminal. Similarly, 24 links to the left and 8 to the right provide less throughput per terminal. For very large networks, where the switches in the right column need to switch more than 32 links, they can be implemented by small indirect networks.

7.3 The Networks Investigated

In the following performance evaluation, three sizes of network are considered, where the size of a network is measured as the number of terminals from it. The networks studied are mostly not practical, in that they do not make efficient use of the routing chips[22], but they are such that the results can be easily interpreted in terms of scalability, and extended directly to other networks of similar form.

Three sizes of network are considered: 64, 256 and 1024. These are natural sizes for the topologies considered.

7.3.1 The binary n–cube

The network sizes are all powers of 2. The smallest network is constructed from 64 C104 switches. These form a six-dimensional cube. On each switch there are six links in use for the network, and one more for the traffic source and sink (the terminal link).

The 256 size cube is constructed as 256 switches, connected as a degree 8 cube. The 1024 size cube is 1024 switches, constructed as a degree 10 cube. For all three sizes, smaller ‘‘fatter’’ cubes would more fully utilize the C104s. These are cubes of lower dimension, with several links in parallel where only a single link would otherwise be used.

Deterministic routing on the hypercube is done from the highest dimension downwards, providing the deadlock free routing described in chapter 1.

7.3.2 The two–dimensional grid

The grids examined are all square. Each switch uses one link in each direction ($+x$, $-x$, $+y$, $-y$) and there is one terminal at each switch. The links at the edges of the grid are not used.

The 64 size grid is therefore made from 64 switches, arranged as an 8 by 8 square. The 256 size is 16 switches square, and the 1024 size is 32 switches square. The same number of terminals would be given by using smaller grids with, say, four terminals per node, and parallel links between the nodes.

Deterministic routing on the grid is first in the y direction, then in the x direction, providing the deadlock free routing described in chapter 1.

7.3.3 Indirect multistage networks

The indirect multistage networks considered here are all *low-cost* networks. A larger number of switches can be used to make the network more highly connected: this tends towards the indirect butterfly. The networks examined therefore indicate the performance characteristics of the ‘‘cheaper’’ networks in the class. The 64-way network illustrated in figure 7.2 has eight links

[22] In most cases a far larger number of terminals could be connected with the number of switches used.

for each one from left to right shown in the diagram, and the 256-way network illustrated in figure 7.3 has two. A 1024–way network can be constructed from 32–way routers by using a 64–way network for each of the center stage switches, as illustrated in figure 7.4.

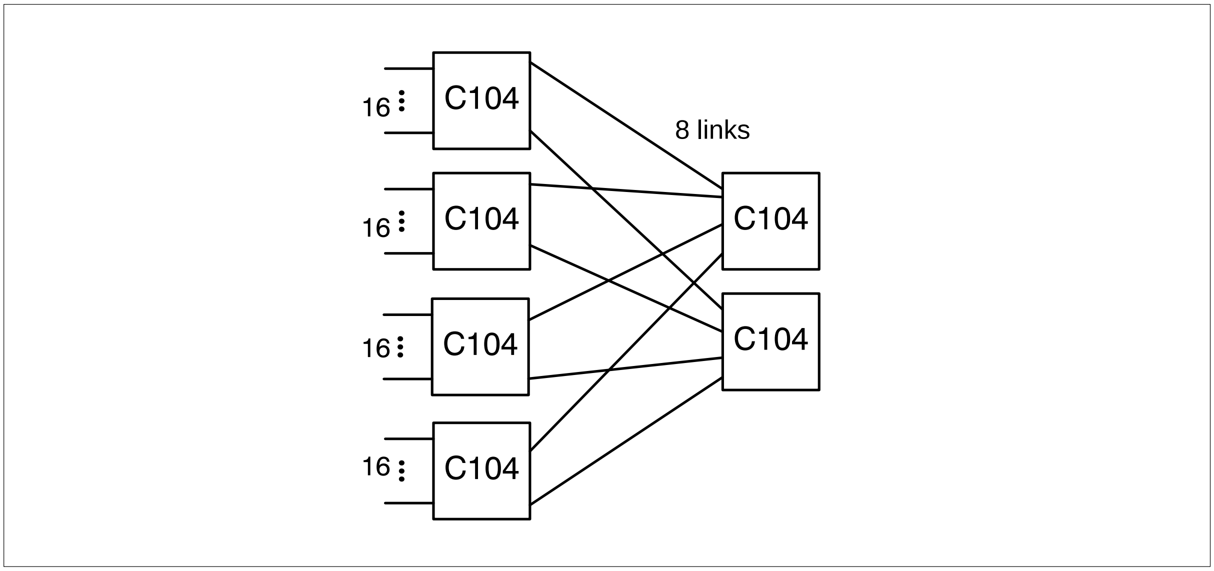

Figure 7.2 64–way multistage network

Deterministic routing on the indirect multistage network routes an inbound packet at the left hand side via an appropriate right hand side node to the destination terminal at the left hand side.

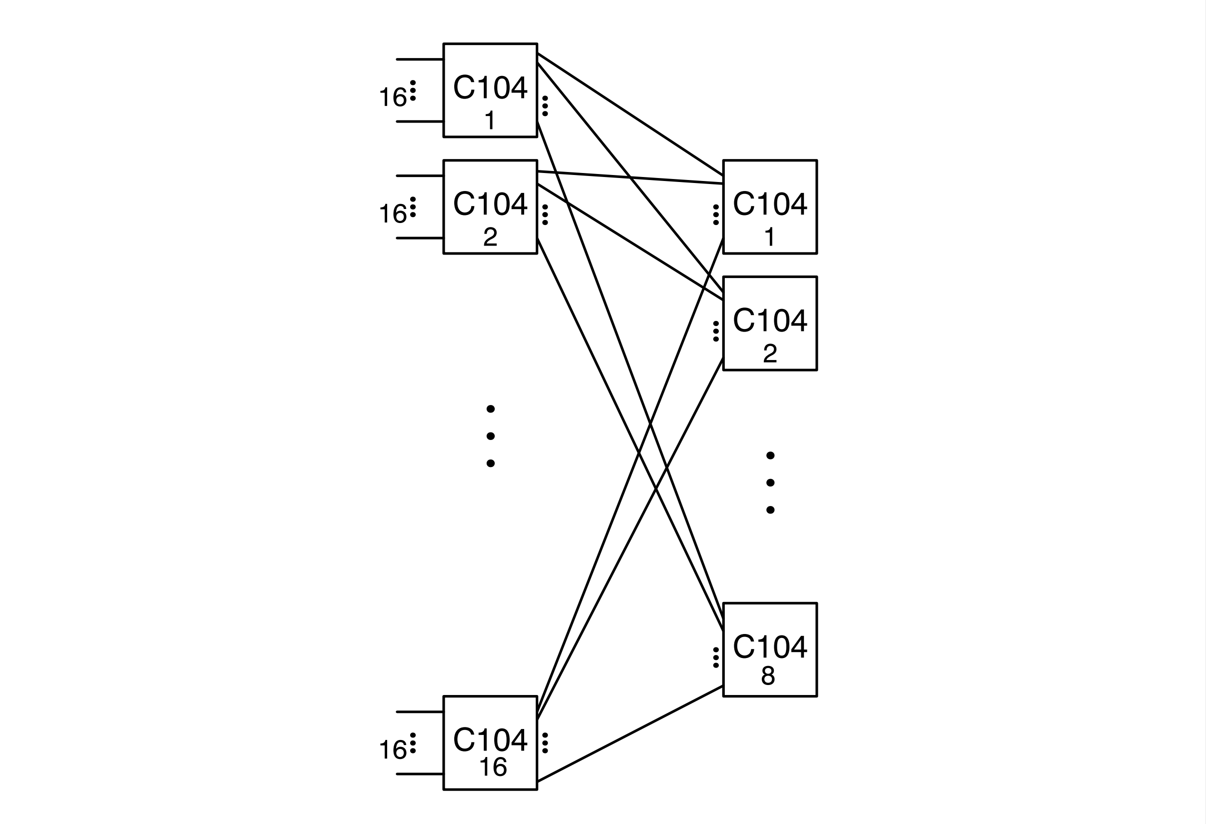

Figure 7.3 256–way multistage network

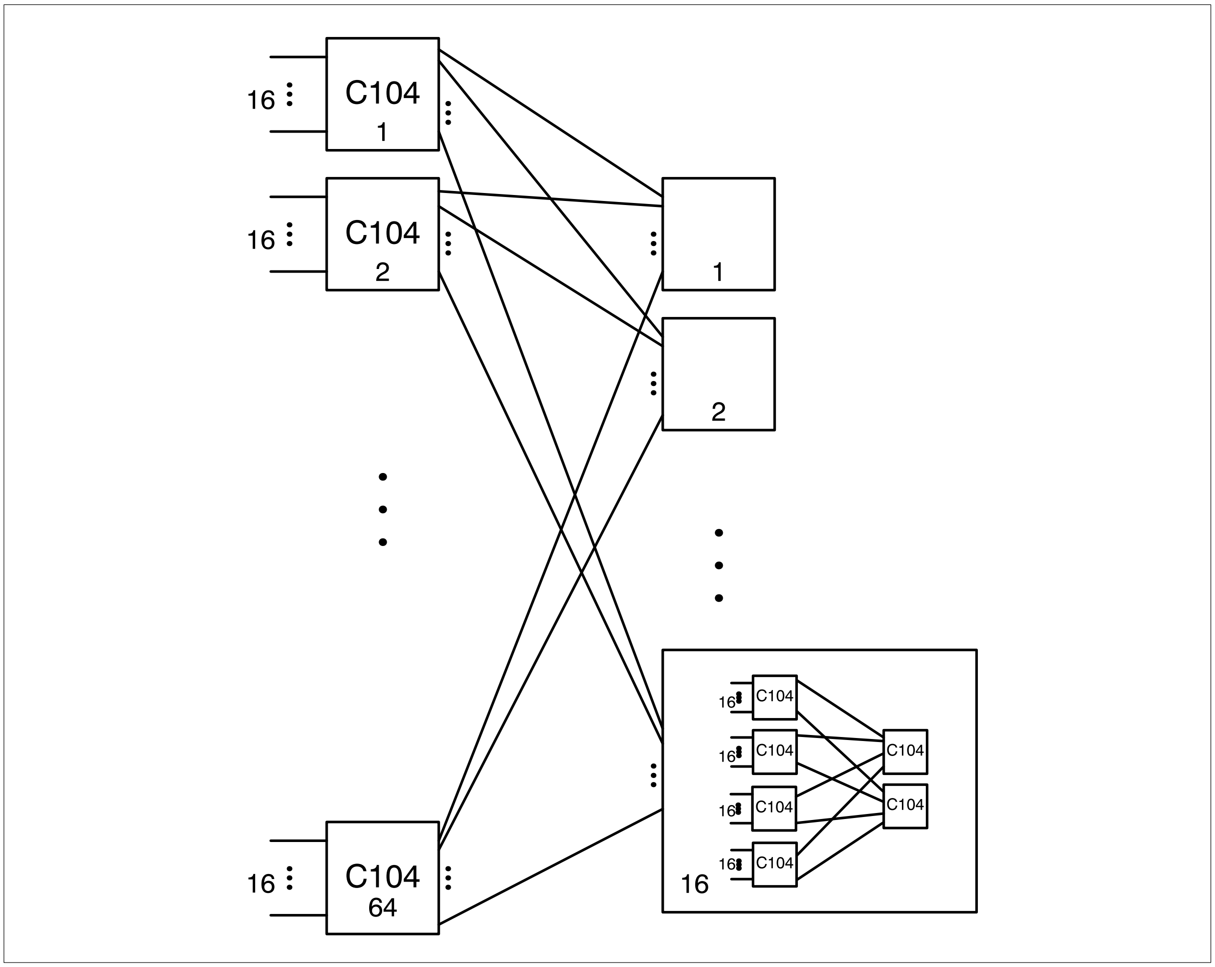

Figure 7.4 1024–way multistage network

7.4 The traffic patterns

In designing universal communication networks, we are interested in network throughput where network properties are not exploited by the traffic pattern. The important features are the local throughput and delay for continuous operation, and the way in which throughput and delay scale with network size. Once the continuous throughput of the network has been determined, the load may be adjusted to take advantage of the throughput.

The symmetry of the cube and multistage networks mean that for continuous traffic approximately the same number of packets are injected at each node. On the grid network, one would expect the edge nodes to inject a smaller number of packets than the centre nodes. To understand these effects, we also measured a traffic pattern in which a fixed number of messages was injected from every terminal, but the difference produced by this traffic pattern was not significant.

7.4.1 Continuous Random traffic

The continuous traffic is created at each source. Whenever an input queue is empty, a new packet is created and put on the queue. The destination of this packet is chosen at random from all possible destination addresses. This is a good pattern as it dissipates traffic over the network, in a similar manner to that of universal routing. However, such random behavior will obviously create a number of packets from different sources which are going to the same destination. This causes contention at the destination, and the effects of this are discussed later on.

7.4.2 Systematic traffic patterns

For a systematic pattern, each source sends to a specific destination. When an input queue is empty, a packet is created to this pre-defined address. Each of the patterns chosen is a permutation, so that no two source nodes send to the same destination. Therefore the contention seen in these patterns is wholly a feature of the network and routing algorithm.

For each network, a systematic traffic pattern is chosen. The patterns seem harmless enough, and represent an operation which could be reasonably expected to be performed. However, in each case the pattern chosen will create severe hot–spots in the network. These are, in a sense, worst-case traffic patterns.

7.5 Universal Routing

For a communication network, we would like to be able to bound delay and achieve scalability of throughput. The bound on delay will, with deterministic routing, depend on the traffic patterns currently in transit in the network. Some of these patterns will be fast, others slow. Universal routing overcomes this problem by bounding the amount of time a set of communications is likely to take. The probability of exceeding this time can be made arbitrarily small. Improvement of the upper bound is of considerable benefit, and since we are only interested in the upper bound any detrimental effect on fast patterns is inconsequential.

The practice of universal routing is straightforward. An amount of random behavior is introduced. This ‘‘upsets’’ systematic traffic patterns which cause the exceptional delays, and disperses the load across the network so that more links can be used concurrently. The realization of the random behavior depends on the underlying topology.

On the cube, a packet is sent to a random intermediate node in the network, then it continues to its destination. The journey to the random intermediate node and the final destination node makes use of the appropriate deterministic routing algorithm. This means that the average packet travels twice as far, so in order to maintain throughput, twice as many links are needed. The links are partitioned into two parallel networks, one of which carries the traffic on its journey to the random intermediate node, and the other carries the traffic from the random intermediate node to the required destination.

On the grid, it is only necessary to randomize in one dimension. This is the second direction in which the packet would normally travel. So for routing which goes first in the y direction, then the x direction, universal routing takes a packet first to a random node in the x direction. An extra set of links is used in the x direction specifically for this random step.

Universal routing on the multistage network sends a packet via a randomly chosen node on the right hand side, see figure 7.1. This does not increase the number of links required in the network. In practice, even better results can be obtained by using grouped adaptive routing to make the selection of link to the right hand switches.

7.6 Results

The simulation examines continuous traffic in a network in equilibrium. The throughput is measured as a percentage of the maximum possible throughput of each input link. Delay is measured in terms of header times: this is the amount of time it takes a header to be output from a switch, received at the next switch, and processed ready for output at that switch, which for the C104 is approximately 500ns. Time units are therefore consistent throughout.

7.6.1 The n–cube

The systematic traffic pattern

The n–cube is perhaps the most difficult of the three structures to visualize, especially for the larger examples. Therefore the systematic traffic patterns on the cube will be described for a small part of the network, then extended.

A seven dimensional cube can be partitioned into a number of three-dimensional cubes. Two of these 3-cubes can be joined by a ‘‘middle dimension’’ link. The 7-cube can be partitioned so that each node lies in exactly one such sub-structure.

For a permutation, which is one-to-one by definition, the maximum congestion will occur at the middle dimension. Therefore the 3-cube on one end of the middle link is mapped to the 3-cube at the other end of the middle link, and vice versa. This is the essence of the underlying permutation for systematic traffic on the cubes.

The cubes which are examined are all of even dimension. So the traffic pattern for one dimension less is used, and each packet moves along the first dimension. (This will not increase the contention, but will increase slightly the time taken). On the 6-cube, a 2-cube (square) maps to a 2-cube, therefore giving four-way contention. On the 8-cube, a 3-cube maps to a 3-cube, giving eight-way contention. The 10-cube gives 16-way contention. So with the increase in the dimension of the cube, we can expect the throughput per terminal of the network to halve. The delays for the systematic traffic are expected to double with the increase in dimension of the cube.

Results for the binary n–cube

Table 7.1 Random traffic on the n–cube

Network size	Mean delay	Max delay	Throughput(%)
64	48	322	78.8
256	59	546	70.2
1024	64	655	71.1

Table 7.2 Systematic traffic on the n–cube

Network size	Mean delay	Max delay	Throughput(%)
64	188	376	25.1
256	383	1618	12.5
1024	722	3679	6.3

Table 7.3 Universal Routing on the n–cube

Network size	Mean delay	Max delay	Throughput(%)
64	59	260	84.7
256	80	514	67.5
1024	91	605	71.7

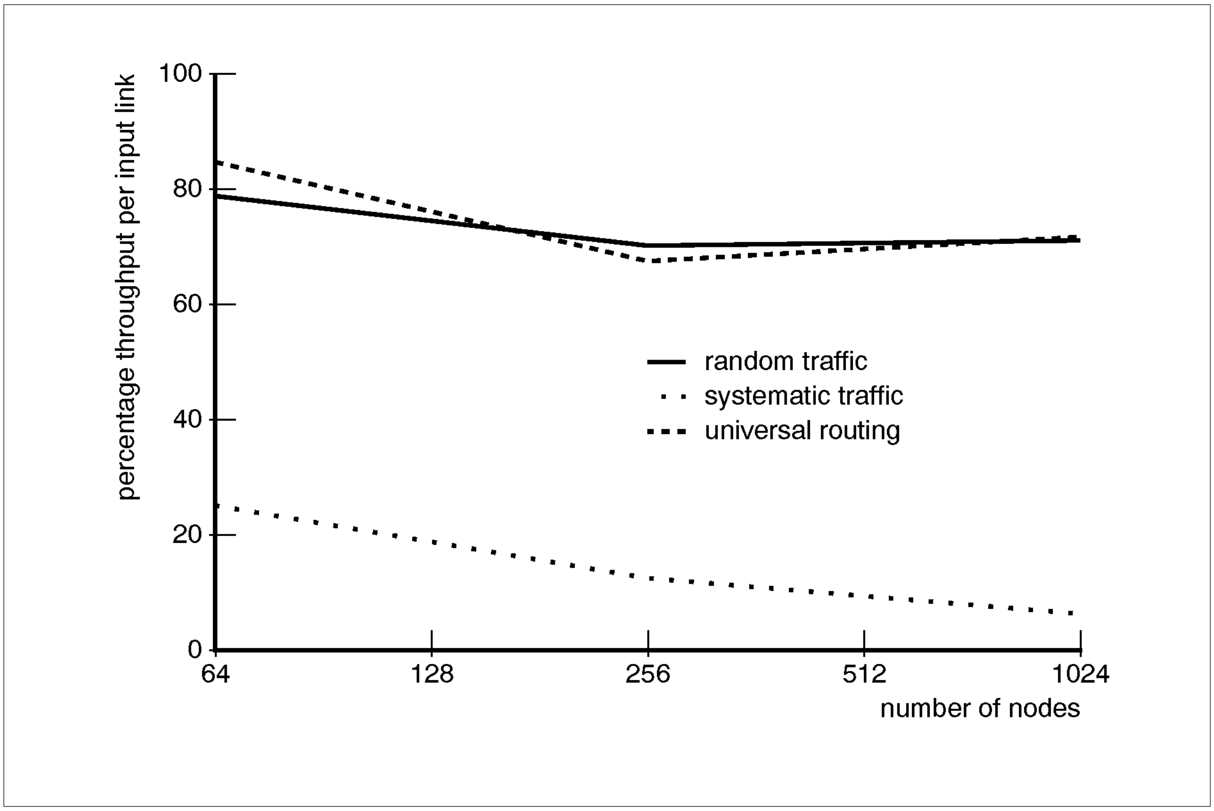

Figure 7.5 Throughput varying with network size on the n–cube

Discussion

The continuous random traffic shows the throughput and delay to scale, as predicted. Universal routing has the effect of adjusting the nature of the systematic permutation towards that of the random traffic. The variation of throughput with network size is due to variation within the random number generator.

The results show the behavior of the systematic permutation to be as expected, with a large increase in delay and a large decrease in throughput for an increase in network size. Note the relative decrease in throughput as network size increases. For the 6-cube, throughput is about one third of that for random traffic, the 8-cube reduces to one sixth of the random traffic throughput, and the 10-cube to a mere twelfth of the random traffic throughput.

As an aside, there is an interesting aspect of the delay figures. Comparing the random traffic with the universal routing shows that the universal routing does not double the delays. This is counter-intuitive, as the universal routing sends messages, on average, twice as far. However, this anomaly is explained by the nature of random traffic. As noted earlier, random traffic will send several packets to the same destination. This is a major cause for delay for the random traffic. However, the universal routing on the systematic traffic does not cause the same destination contention (and does not cause contention at the randomly chosen node because random headers are removed at each link in the switch).

7.6.2 The two–dimensional grid

The systematic traffic pattern

On a grid, a block move provides the permutation on which to base the systematic traffic. The grid is divided into four sets of nodes, with the nodes being bisected in both the x and y directions. The top left corner is translated onto the bottom right, and vice versa. This means that messages are delayed in both the y and x direction when travelling to their destination. Note that the four separate block moves are independent of each other.

The amount of contention doubles in both the x and y direction with an increase in network size. This suggests that throughput will decrease by a factor of 4, and that the average delay will at least double with each increase in network size.

Results for the 2–D grid

Table 7.4 Random traffic on the 2–D grid

Network size	Mean delay	Max delay	Throughput(%)
64	116	1135	34.2
256	223	4442	17.5
1024	336	19937	7.9

Table 7.5 Systematic traffic on the 2–D grid

Network size	Mean delay	Max delay	Throughput(%)
64	302	1311	12.6
256	861	7126	3.1
1024	1916	36833	0.8

Table 7.6 Universal Routing on the 2–D grid

Network size	Mean delay	Max delay	Throughput(%)
64	187	1095	21.9
256	368	2178	11.2
1024	826	4725	5.1

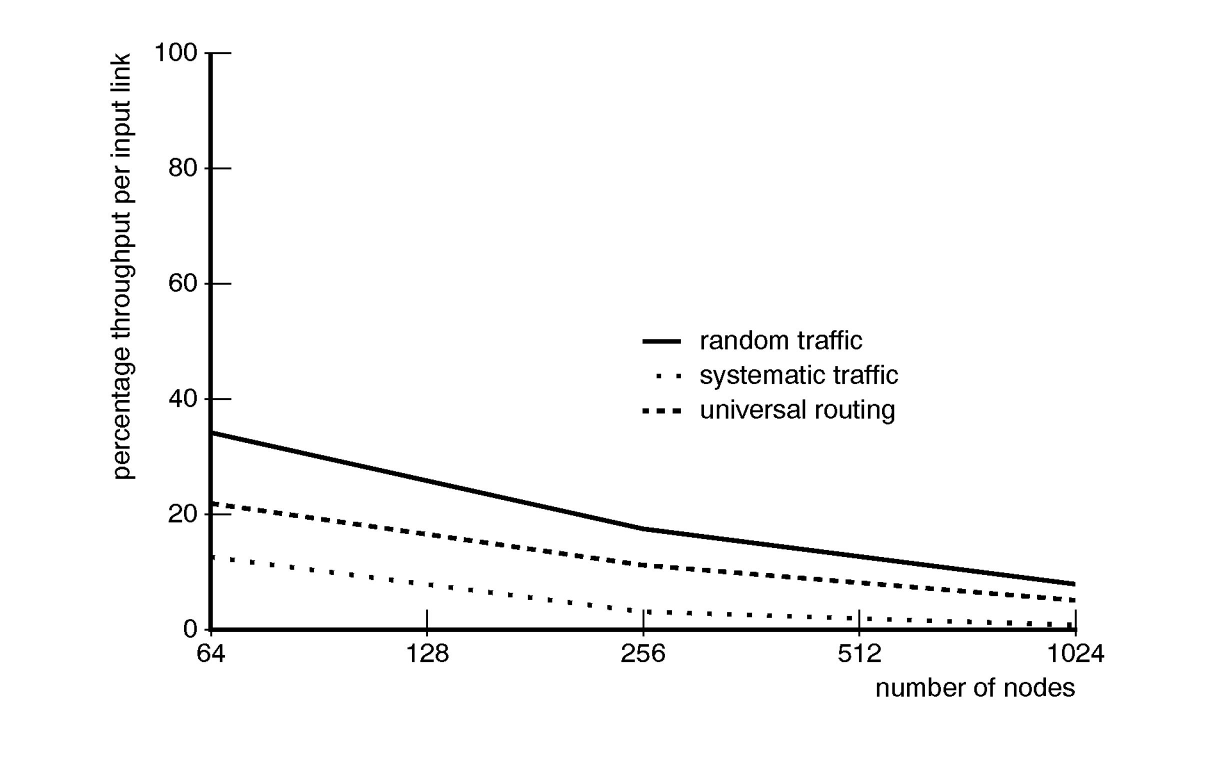

Figure 7.6 Throughput varying with network size on the 2–D grid

Discussion

The continuous random traffic shows that throughput per node degrades with increasing network size. This is to be expected, as the grid does not increase network capacity at a suitable rate. The delay increases quickly with network size.

Systematic traffic shows that the throughput and delay on a grid can both be affected considerably by the traffic pattern. Again, the throughput per terminal decreases with the network size. Universal routing pulls the behavior back towards the random traffic, providing similar scalability in both throughput and delay. Throughput is now limited only by the overall capacity of the network. For the grid, the universal routing takes longer than the random traffic, as expected.

7.6.3 Indirect multistage networks

The systematic traffic pattern

The systematic traffic pattern is built upon a very straightforward permutation. In each case, the source node adds a particular value (modulo the number of nodes), to its own identity number. This number is chosen so that all traffic is routed through a single mid-layer switch. Note that there is no contention within the switch, as messages contend for the links. However, this ensures a large amount of contention for both the inbound and the outbound links of that switch.

Consider these patterns compared to random traffic. For the 64-way network, shown in figure 7.2, traffic all goes via the top switch on the right hand side. As there are two central switches, this can be expected to reduce bandwidth by about a half compared to the random traffic, as only one half of the links out of the left hand side are used.

Results for the indirect multistage networks

Table 7.7 Random traffic on the MIN

Network size	Mean delay	Max delay	Throughput(%)
64	36	442	56.8
256	46	512	48.8
1024	78	1078	30.0

Table 7.8 Systematic traffic on the MIN

Network size	Mean delay	Max delay	Throughput(%)
64	44	44	36.0
256	204	364	8.6
1024	408	622	4.4

Table 7.9 Universal Routing on the MIN

Network size	Mean delay	Max delay	Throughput(%)
64	48	228	41.7
256	46	284	47.6
1024	111	926	21.3

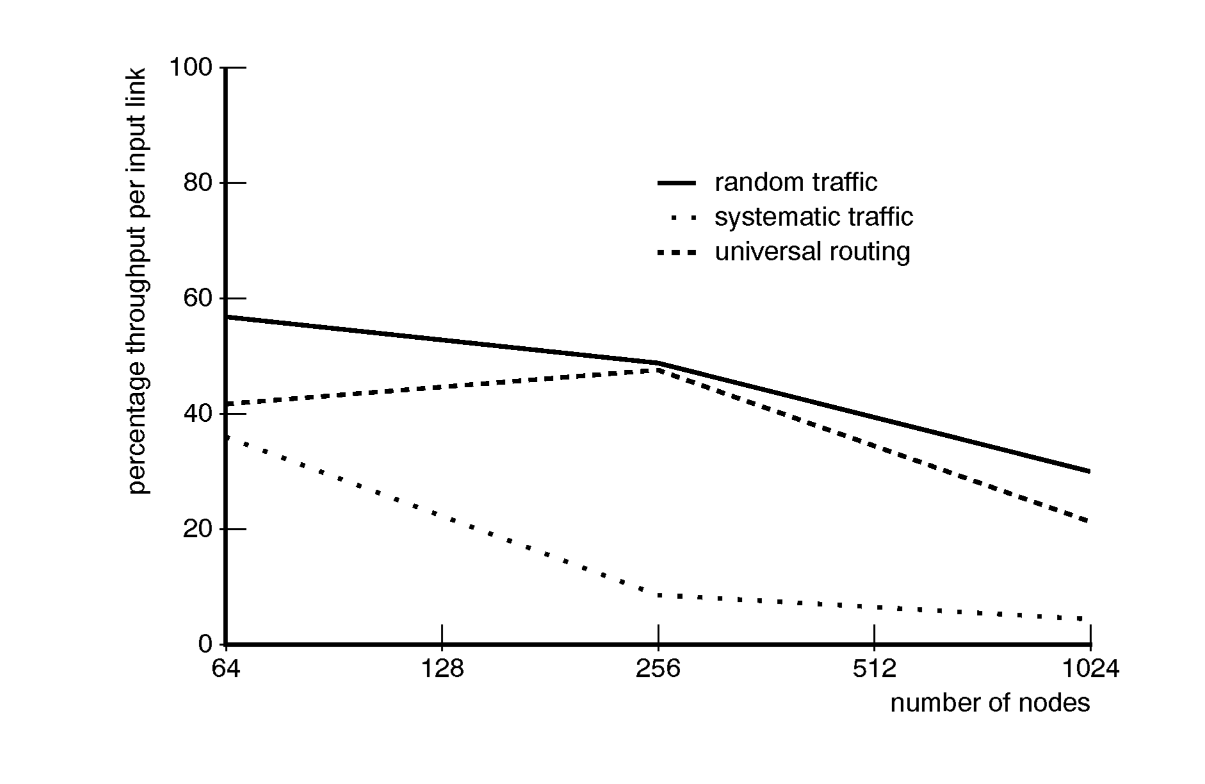

Figure 7.7 Throughput varying with network size on the indirect multistage networks

Discussion

Random traffic on the indirect multistage network shows that in the low-cost networks considered the throughput per node degrades with network size. However, the number of input links per switch can be altered, and the "centre" of the network made a lot more richly connected. This will improve the scalability characteristics.

Systematic traffic patterns show that the indirect networks have traffic patterns which can severely affect bandwidth and delay, and once again universal routing will overcome these problems. The universal routing graphs do not look smooth because the structure of the networks varies.

7.6.4 Scalability

The networks examined are all appealing for varying reasons, theoretical or practical. The hypercube satisfies the requirement for constant throughput from a node as the network size increases, whereas the grid and indirect multistage networks tail-off in throughput as the network size increases. For the grid, using up to 4 such networks in parallel would not give the throughput of a single link to a cube structure, for networks over 256 nodes. The indirect multistage networks could be replicated to provide this throughput. Note that 4-way replication of the 1024-node network gives a total throughput from the processor similar to the throughput from a single link which is available from a cube. These approximate calculations assume that traffic is split optimally over the parallel networks.

On all of the networks, universal routing removes the varying delays due to traffic pattern contention. In each case, it provides a means of taking advantage of the bandwidth inherent in the network structure.

7.6.5 Is this good use of link bandwidth?

One of the disadvantages of universal routing is the additional link bandwidth which is required. For instance, on the n-cube, the number of links required is doubled. This raises the issue as to whether these extra links are being well used. If they were used instead to ‘fatten’ the original cube structure, would deterministic routing provide a better solution?

If the links were doubled then the throughput could be doubled for deterministic routing which used both available paths optimally. However, even doubling the throughput on the cubes does not bring the systematic traffic throughput close to that of universal routing. This suggests that universal routing does not only give scalability, but also makes good use of link bandwidth.

For the indirect networks no extra links are used, and on the grid 1.5 times as many links are used. These factors also show that using the links for universal routing is preferable to extra links and deterministic routing on these structures.

7.7 Performance Predictability

The previous results show that universal routing can improve the throughput and bandwidth scalability of a network. In this section, universal routing is shown to improve the predictability of the network also.

We investigate the 8-cube. Each node in the network sends to a distinct destination node, i.e. the traffic pattern is a permutation. If each node creates twenty packets to the same destination, the resulting traffic pattern is called a 20-permutation.

The underlying permutation is the perfect shuffle, which is obtained by deriving the destination node number by rotating the bits of the source node number by a particular amount. A rotation of 1 gives a 2-way shuffle, the rotate of 2 gives a 4-way shuffle, and so on. The rotation was varied from 0 to the cube dimension and the time measured for a 20-permutation to complete using both deterministic and universal routing.

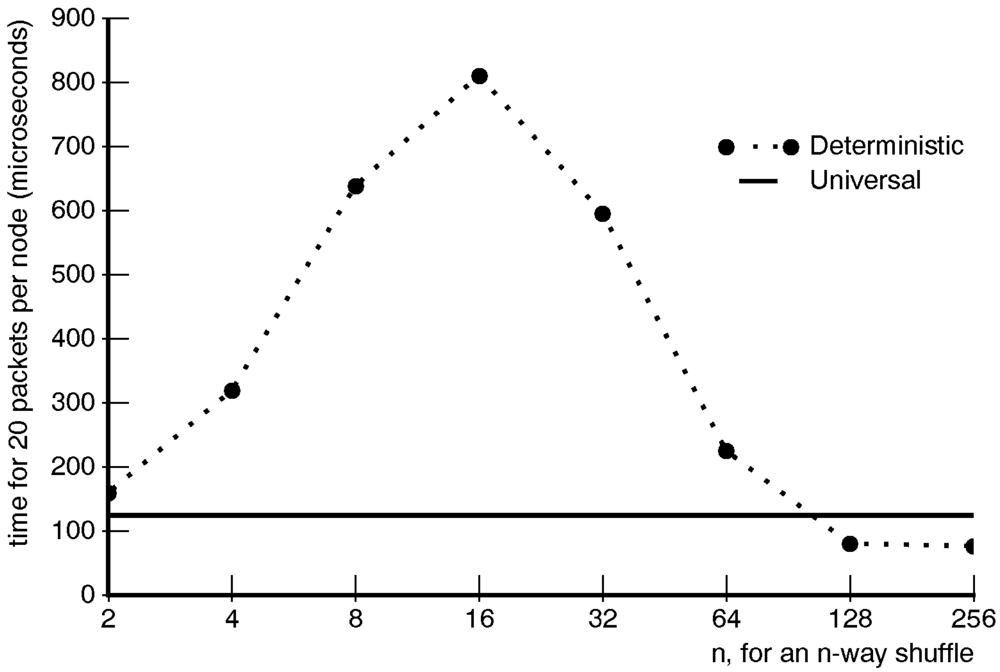

Figure 7.8 The variation of time taken to finish with the degree of shuffle

The results for the 8–cube are shown in figure 7.8. This shows that the deterministic routing gives a wide variation in run-time. For instance, changing to an 8-way shuffle rather than a 4-way shuffle increases the network delivery time by a factor of 2. With universal routing the time taken remains approximately constant (a representative value is shown). This is a major advantage, since calculating a bound on the run-time requires the worst case to be taken into account.

Again, the extra links for universal routing could be used for deterministic routing and provide extra bandwidth to allow the permutation to finish in about half of the time. However, the variability remains, and most of the deterministic routing cases would still be worse than the universal routing.

7.8 Conclusions

In this chapter we have examined communication networks which can now be built from state-of-the-art VLSI technology. Each of the networks investigated has been shown to have a systematic traffic pattern which severely effects its performance. The detrimental effect of this pattern grows with increasing network size.

The inherent scalability of the networks have been illustrated by the use of random traffic patterns. The use of universal routing provides scalability similar to that of random traffic patterns, for the systematic traffic patterns. Results have highlighted the unpredictable nature of deterministic routing, and shown that the use of links for universal routing restores predictability and the scalability inherent to the network structure.