Survey of routing algorithms for efficient design of Router in NOC

M.Deivakani

Department of ECE,
PSNA College of Engineering & Technology,
Dindigul,TamilNadu.
mdeivakani82@gmail.com

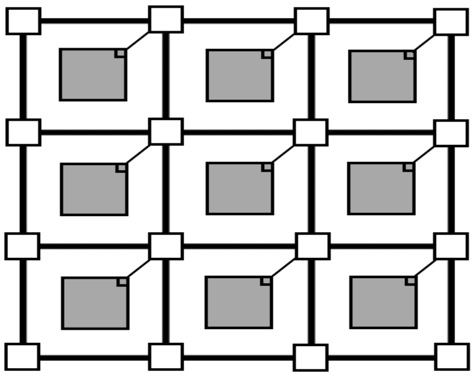
Dr.D.Shanthi

Department of CSE,
PSNA College of Engineering & Technology,
Dindigul, TamilNadu.
dshan71@gmail.com

***Abstract*— With the need to integrate an increasing variety of Semiconductor Intellectual Property (IP) blocks in a single System on Chip (SoC), communication management becomes critical when highly diversified functions with varying latency and bandwidth requirements must be supported in an interconnect fabric. NoC (Network on Chip) technology simplifies the hardware required for switching and routing functions, allowing SoCs with NoC interconnect fabrics to reach higher operating frequencies. Routers in NOC need to have some information about network status in order to make decisions regarding how and where to send packets.In this paper we discusses various efficient routing algorithms to design router in NOC**

Keywords*—Routing Algorithm,NOC,Router,Low power network on chip

NoC technology is often called “a front-end solution to a back-end problem.” As semiconductor transistor dimensions shrink and increasing amounts of IP block functions are added to a chip, the physical infrastructure that carries data on the chip and guarantees quality of service begins to crumble. Many of today's systems-on-chip are too complex to utilize a traditional hierarchal bus or crossbar interconnects approach.



.

Fig.1.Network On Chip

A. Router

A **router** is a networking device, commonly specialized hardware, that forwards data packets Between computer networks. This creates an overlay internetwork, as a router is connected to two or more data lines from different networks. When a data packet comes in one of the lines, the router reads the address information in the packet to determine its ultimate destination. Then, using information in its routing table or routing policy, it directs the packet to the next network on its journey. Routers perform the "traffic directing" functions on the Internet. A data packet is typically forwarded from one router to another through the networks that constitute the internetwork until it reaches its destination node. A router is always configured with some default route. A default route tells the router where to forward a packet if there is no route found for specific destination. In case there are multiple path exists to reach the same destination, router can make decision based on the following information:

- Hop Count
- Bandwidth
- Metric
- Prefix-length
- Delay

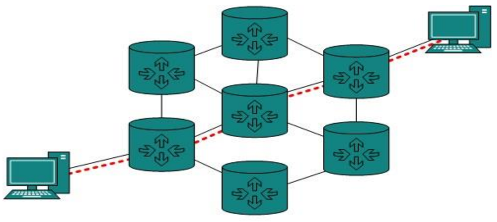


Fig.1 Routing

(i)Performance Issues of routing algorithm in Router

1. **Correctness**: The algorithm must deliver every packet to its ultimate destination

2. Complexity: The algorithm for the computation of the tables must use as few messages, time, and storage as possible

3. Efficiency: The algorithm must send packets through good paths

4. Robustness: In the case of a topological change, the algorithm updates the routing tables appropriately

5. Fairness: The algorithm must provide service to every user in the same degree

(ii)Good paths

1. Minimum hop: The cost of a path is the number of hops

2. Shortest path: Each channel has a non-negative cost –the path cost is the sum of the cost of the edges. Packets are routed along shortest paths.

3. Minimum delay/congestion: The bandwidth of a path is the minimum among the bandwidths of the channels on that path.

4. Most robust path: Given the probability of packet drops in each channel, packets are to be routed along the most reliable path

There are many routing algorithms which are used to determine the path, load, and distance over the network traffic. In this paper, a routing algorithm is taken as a tool for making an analysis over the research work done in network algorithms. Routing algorithms are classified as adoptive routing algorithms and non adoptive routing algorithms. An adoptive routing algorithm is an algorithm in which the network path can changes their routing ways s in accordance to the changes taken place in the network topology and in the traffic. It is having a dynamic routing table in which it sends data over the network. Distance vector routing algorithm, link state routing algorithm, distributed routing algorithms are comes under the category of adoptive routing algorithms. The non adoptive routing algorithms are the algorithms in which it follows a static routing table for the data to allow transmission over the network. This algorithm does not adjust with the current

Many research papers have reviewed and analyzed the performance of the adoptive routing algorithms and the non adoptive routing algorithms. Most probably, the adoptive routing algorithms give better efficiency in packet delivery ratio, throughput, routing overhead etc.

1)Classification of Routing Algorithms

The routing algorithms may be classified as follows:

I. **Adaptive Routing Algorithm:** These algorithms change their routing decisions to reflect changes in the topology and in traffic as well. These get their routing information from adjacent routers or from all routers. The optimization parameters are the distance, number of hops and estimated transit time. This can be further classified as follows:

   1. **Centralized:** In this type some central node in the network gets entire information about the network topology, about the traffic and about other nodes. This then transmits this information to the respective routers. The advantage of this is that only one node is required to keep the information. The disadvantage is that if the central node goes down the entire network is down, i.e. single point of failure.

   2. **Isolated:** In this method the node decides the routing without seeking information from other nodes. The sending node does not know about the status of a particular link. The disadvantage is that the packet may be send through a congested route resulting in a delay. Some examples of this type of algorithm for routing are:

      ▪ **Hot Potato:** When a packet comes to a node, it tries to get rid of it as fast as it can, by putting it on the shortest output queue without regard to where that link leads. A variation of this algorithm is to combine static routing with the hot potato algorithm. When a packet arrives, the routing algorithm takes into account both the static weights of the links and the queue lengths.

      ▪ **Backward Learning:** In this method the routing tables at each node gets modified by information from the incoming packets. One way to implement backward learning is to include the identity of the source node in each packet, together with a hop counter that is incremented on each hop. When a node receives a packet in a particular line, it notes down the number of hops it has taken to reach it from the source node. If the previous value of hop count stored in the node is better than the current one then nothing is done but if the current value is better then the value is updated for future use. The problem with this is that when the best route goes down then it cannot recall the second best route to a particular node. Hence all the nodes have to forget the stored informations periodically and start all over again.

   3. **Distributed:** In this the node receives information from its neighbouring nodes and then takes the decision about which way to send the packet. The disadvantage is that if in between the the interval it receives information and sends the paket something changes then the packet may be delayed.

II. **Non-Adaptive Routing Algorithm:** These algorithms do not base their routing decisions on measurements and estimates of the current traffic and topology. Instead the route to be taken in going from one node to the other is computed in advance, off-line, and downloaded to the routers when the network is booted. This is also known as static routing. This can be further classified as:

a. **Flooding:** Flooding adapts the technique in which every incoming packet is sent on every outgoing line except the one on which it arrived. One problem with this method is that packets may go in a loop. As a result of this a node may receive several copies of a particular packet which is undesirable. Some techniques adapted to overcome these problems are as follows:

Sequence Numbers: Every packet is given a sequence number. When a node receives the packet it sees its source address and sequence number. If the node finds that it has sent the same packet earlier then it will not transmit the packet and will just discard it.

Hop Count: Every packet has a hop count associated with it. This is decremented(or incremented) by one by each node which sees it. When the hop count becomes zero(or a maximum possible value) the packet is dropped.

Spanning Tree: The packet is sent only on those links that lead to the destination by constructing a spanning tree routed at the source. This avoids loops in transmission but is possible only when all the intermediate nodes have knowledge of the network topology.

Flooding is not practical for general kinds of applications. But in cases where high degree of robustness is desired such as in military applications, flooding is of great help.

Random Walk: In this method a packet is sent by the node to one of its neighbours randomly. This algorithm is highly robust. When the network is highly interconnected, this algorithm has the property of making excellent use of alternative routes. It is usually implemented by sending the packet onto the least queued link.

II. Dijkstra's Algorithm

Dijkstra's algorithm is an iterative algorithm that provides us with the shortest path from one particular starting node to all other nodes in the graph. Again this is similar to the results of a breadth first search. Initialization:

Notation:

•c(x,y): link cost from node x to y; = ∞ if not direct Neighbors

• D(v): current value of cost of path from source to dest. v

• p(v): predecessor node along path from source to v

• N': set of nodes whose least cost path definitively known

Initialization:

```
N' = {u}
for all nodes v
if v adjacent to u
then D(v) = c(u,v)
else D(v) = ∞
Loop
find w not in N' such that D(w) is a minimum
add w to N'
update D(v) for all v adjacent to w and not in N' :
D(v) = min( D(v), D(w) + c(w,v) )
new cost to v is either old cost to v or known
shortest path cost to w plus cost from w to v */
 until all nodes in N'
```

III. The Floyd Warshall Algorithm

This algorithm iterates on the set of nodes that can be used as intermediate nodes on paths. This set grows from a single node (say node 1) at start to finally all the nodes of the graph. At each iteration, we find the shortest path using given set of nodes as intermediate nodes, so that finally all the shortest paths are obtained.

Notation

$D_{i,j}[n]$ = Length of shortest path between the nodes i and j using only the nodes 1,2,....n as intermediate nodes.

Initial Condition

$D_{i,j}[0] = d_{i,j}$ for all nodes i,j .

Algorithm

Initially, n = 0. At each iteration, add next node to n. i.e. For n=1,2,.....N-1 ,

$$D_{i,j}[n+1] = \min \{ D_{i,j}[n],\ D_{i,n+1}[n] + D_{n+1,j}[n] \}$$

Principle

Suppose the shortest path between i and j using nodes 1,2,...n is known. Now, if node n+1 is allowed to be an intermediate node, then the shortest path under new conditions either passes through node n+1 or it doesn't. If it does not pass through the node n+1, then $D_{i,j}[n+1]$ is same as $D_{i,j}[n]$. Else, we find the cost of the new route, which is obtained from the sum, $D_{i,n+1}[n] + D_{n+1,j}[n]$. So we take the minimum of these two cases at each step. After adding all the nodes to the set of intermediate nodes, we obtain the shortest paths between all pairs of nodes together. The complexity of Floyd-Warshall algorithm is $O(N^3)$.

IV. Distance Vector Routing

The basic idea: look at costs that your neighbors are advertising to get a packet to a destination; select the neighbor whose advertised cost, added with the cost to get to that neighbor, is lowest. You also need to advertise your costs to your neighbors too. The cost is usually delay time

• Formally, each router maintains two vectors called delay and successor node, so router i has

$$Di = [di1 \cdots diN]^T \text{ and } Si = [si1 \cdots siN]^T$$

where dij is current estimate of minimum delay from router i to j (dii = 0), N is number of routers in network, and sij is next node in current minimum-delay route from i to j

• Periodically, each router exchanges its delay vector with all its neighbors, and on the basis of all incoming delay vectors, router k updates both its vectors: The distance vector routing has some problems: responds slowly to congestion and delay increases, and has been replaced by link state routing

V. Link State Routing

The idea behind the link state routing is simple: each router must do the followings

– Find out its neighbors and get their network addresses

– Measure the delay or cost to each of its neighbors

– Construct a link state packet to tell all it has just learnt

– Send this packet to all the other routers

– Find the shortest path to every other router, i.e. a sink tree

A router knows its network interfaces. just sends a HELLO packet on each point-to-point link, and the router at the other end must reply telling who it is with its unique network address Situation is complicated with a “broadcast” LAN,i.e. two or more routers are connected by a LAN: solution is to consider such a LAN as a node itself. To know the link cost simply send an ECHO packet to the neighbor, measure the round-trip delay and divide it by two, and this will give a reasonable estimate of the actual delay

a. Build link state packets: After collecting information needed, each router builds link state(LSP) packet with its identity, sequence number and age (used in distributing), followed by list of neighbors (identity and link cost), e.g. When to build link state packets: once an hour is often enough

b.Distribute link state packets: Each router maintains list of (source, seq.number) pairs they saw. When a LSP arrives, it is checked against the list. If it is new, it is forwarded on all lines except the one it arrived on; if it is a duplicate, it is discarded. To safeguard against old data, link down etc., age is decremented once a second and every time it is forwarded by a router. When the age reaches zero, the LSP is discarded

VI. CONCLUSION

The small size of Network on Chip circuits sets special requirements for all operations. The network technology of the Internet is very hard to straightly shrink to the NoC so the technologies should be specially adapted to the NoC. The routing algorithms presented in this report are difficult to be set in the order of superiority. Different applications need different routing algorithms. While some algorithm is suitable to one system, another algorithm works better in some other system. However, it can be generalized that in most of the cases a simple algorithm suits to simple systems while complex algorithms fit to more complex systems. Big network traffic amounts in wide complex systems need efficient traffic equalization and congestion avoidance while the most significant features in smaller systems are the low energy consumption and low latency.

REFERENCES

[1] H. Kariniemi, J. Nurmi: Arbitration and Routing Schemes for On-chip Packet Networks. Interconnect-Centric Design for Advanced SoC and NoC (toim: J. Nurmi, H. Tenhunen, J. Isoaho & A. Jantsch), Kluwer Academic Publishers, 2004, pages: 253–282.

[2] M. Ali, M. Welzl, S. Hellebrand: *A Dynamic Routing Mechanism for Network on Chip*. 23rd NORCHIP Conference, 21–22 November 2005,

[3] W. Dally and B. Towles, 2001, Route packets, not wires:on-chip interconnection networks, in DAC, pp. 684–689.

[4] C. J. Glass, L. M. Ni, “The turn model for adaptive routing”, Proceedings of the International Symposium on Computer Architecture (ISCA), pp.278-287, 1992.

[5] R. V. Boppana, S. Chalasani, “A framework for designing deadlock-free wormhole routing algorithms”, IEEE Transactions on Parallel and Distributed Systems (TPDS), 7(2): 169-183, 1996.

[6] A. Singh, W. J. Dally, A. K. Gupta, B. Towles,“GOAL: A Load-balanced Adaptive Routing Algorithm for Torus Networks”, in Proc. of the International Symp. on Comp. Arch., pp. 194-205, June, 2003.

[7] Terry T. Ye, Luca Benini, Giovanni De Micheli,”Analysis of Power Consumption on Switch Fabricsin Network Routers”, In Proceedings of DAC, 2002

[8] Jipeng Zhou, F. C. M. Lau, “Adaptive fault-tolerant wormhole routing with two virtual channels in 2D meshes”, 7th International Symposium on Parallel Architectures, Algorithms and Networks, 2004. 10-12 May 2004. pp: 142 – 148.

[9] J. Nurmi: Network-on-Chip: A New Paradigm for System-on-Chip Design Proceedings 2005 International Symposium on System-on-Chip, 15–17 November 2005, pages: 2–6.