Latency Characteristics of Wormhole Routed Networks

No Author Given

No Institute Given

Abstract. It has been the custom in performance analysis to assume a fixed delay through the routers of the interconnection network independent of router complexity and network traffic. However it is acknowledged that traffic in the network arising from parallel algorithms has a significant effect on network performance.
In this work we look at some of the features of router design and the effect of traffic on network latency. We introduce the concept of fixed and variable delays within a router and study their effect on delay in wormhole routed interconnection networks. Based on this study we develop expressions to more accurately reflect network latency in a loaded network.

1 Introduction

The switching method describes the process of setting up a physical path through the routers — switching modules, of the interconnection network of a parallel computer. In packet switching, no physical path is set up between the source and destination. A message is first split up into packets and sent a 'hop' at a time from source to destination. The switching technique describes the flow and buffering of packets as they move from source to destination. Wormhole routing is a type of switching technique where a packet is first split into smaller units, flits, before transmission.

Figure 1 presents a simplified picture of wormhole routing. It demonstrates the pipelined effect of wormhole routed packet transmission in an unloaded network or a loaded network where a packet has not encountered contention.

It is often said that the first flit encounters the delay at each node (t_{node}) and on each channel (t_{wire}) whereas the delay of the following flits is that of the final hop. This assumption is often used in wormhole routing analysis where network latency may be expressed

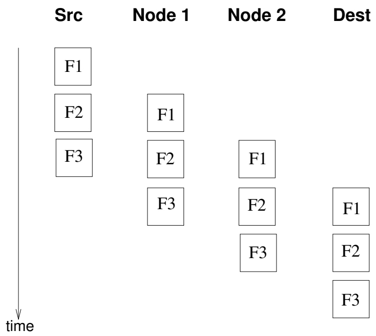

Fig. 1. Wormhole Routing - 3 flit packet

as:

$$network - latency = \frac{F}{W} \times (D+1) \times t_{node} + (D + (\frac{L}{W} - 1)) \times t_{wire} \quad (1)$$

where F is the flit length, W is the channel bandwidth, D is the network diameter and L is the packet length.

This paper is arranged as follows. In section 2 we consider the components of network latency in wormhole routed networks and develop what we have termed our base expression. The following sections — sections 3, 4 and 5 expand the definition of network latency to reflect relevant router features and traffic. A summary and a description of future work required is given in section 6.

2 Base expression for Wormhole Routing

In this section we develop our base expression for wormhole routing, building on the expression given in equation 1.

Theorem 1. *Delays encountered between the source and destination processors and their local routers are non negligible.*

Earlier analysis of network latency often included only the communication delay across the interconnection network [Aga91]. However, as described in [BP93] the contribution of injection (at the source) and consumption (at the destination) delays to end-to-end delay may be substantial especially at the destination where bottlenecks may appear.

Theorem 2. *The node delay encountered by the head of a packet includes decision delays, not encountered by the flits of the body.*

There is an important distinction between the node delay encountered by the header (t_{node}) and that encountered by each of the remaining flits (t_{prop}). Since the decision delay is only encountered by the header then the delay encountered by the header is greater than that of the other flits [BP93]. It can be shown [Dal90] that this can have a large effect in high diameter networks with short messages. The effect is, off course, dependent on the difference in size of the respective delays. However, for long messages the effect may be quite insignificant. As such, it is chosen to separate these two delays and model the size of the packet in terms of the number of flits in the header and body respectively.

Theorem 3. *The delays incurred at the source and destination routers are not necessarily equivalent to the delays incurred at the intermediate nodes.*

An important field of ongoing research in this area is work towards solving the communication bottleneck problem. A large part of which is acknowledged to come from the software level. One aim is to move some of the software functionality into hardware at the router. Implementation in hardware may result in increased throughput but increase the complexity at the source and destination and as such increase the delays at these routers.

One of the arbitration methods implemented in routers is a priority arbitration scheme which gives priority to traffic between the router and it's local processor over traffic that is passing through the router. As such, the delays incurred at the source and destination may differ from that of the intermediate nodes.

Theorem 4. *Wire delay is made up of not only the propagation delay at each stage but the delays through the sending and receiving circuitry as well.*

With present technology and for the foreseeable future, node delay is expected to dominate wire delay [Aga91]. However, as technological advances give rise to decreased feature size with a proportional increase in switching speeds this will result in an increase in wire delay in relation to node delay. Also, the scaling of the sending and receiving circuitry will be limited by the speed of the communication medium and therefore increase the importance of the sending and receiving delays to switching delays [MT90].

From these theorems we develop our base equation for Network latency in a wormhole routed network.

Definition 5. Network Latency with respect to wormhole routed networks, assuming infinite buffering is represented by:

$$Network - latency - WH = t_{inj} + t_{header} + t_{body} + t_{cons} \tag{2}$$

$$\begin{aligned} t_{header} = (h * \frac{F}{W}) * (t_{src} + ((D-1) * t_{node}) + t_{dest}) + \\ ((D + (h-1)) * t_{wire}) \end{aligned} \tag{3}$$

$$t_{body} = (L - h) * \frac{F}{W} * (t_{wire} + t_{prop}) \tag{4}$$

$$t_{wire} = t_{send} + t_{phy} + t_{recv} \tag{5}$$

where h represents the number of flits in the header, t_{inj} represents the injection channel delay, $_{cons}$ represents the consumption channel delay, t_{send} is the delay in the sending circuitry, t_{phy} is the wire propagation delay and t_{recv} is the delay in the receiving circuitry.

3 Decision and Queueing

Theorem 6. *The delay encountered at a given router is made up of a fixed part and a variable part.*

The fixed delay is dependent on the design of the router and independent of network loading whereas the variable delay is made up of queueing and arbitration delays which are dependent on the network loading as well as some of the design issues. The size of the fixed delay is dependent on whether the router is the source, destination or intermediate node on the packet's path.

In the case of the packet header, a decision and arbitration for output channel is required at each node so that the header creates a path through the network for the remaining flits to follow. As such, a variable delay is incurred at each node on route to the destination.

The size of the variable delay depends on the amount of queueing or arbitration delays at a router which is in turn dependent on the amount of traffic competing within the node at a particular point in time. The effect of the variable delay on the overall path delay is a compounded delay made up of elements from each node on the path. As such, a constant cannot realistically represent this part of the delay in a loaded network. Definition 6 redefines equation 3 in the light of fixed and flexible delays.

Definition 7. The delay experienced by the head of a packet in a wormhole routed network taking into account the effect of variable delays on the path to the destination, can be expressed as:

$$t_{header-var} = (h * \frac{F}{W}) * (t_{src-fx} + ((D-1) * t_{node-fx}) + t_{dest-fx} + \sum_{0 \leq i \leq D} t_{var}) + (D + (h-1)) * t_{wire})) \tag{6}$$

where $t_{src\text{-}fx}$, $t_{node\text{-}fx}$ and $t_{dest\text{-}fx}$ are the fixed delays incurred at the source, intermediate and destination nodes respectively and t_{var} is the variable delay incurred by the header at either an intermediate, source or destination node.

4 Output Buffering

In this section we consider the effect of buffering on wormhole routing analysis, limiting the description to the case of one-flit buffers.

Definition 8. For pipelined transmission, assuming that a one-flit buffer is available at each output channel, the following condition must be satisfied.

$$(t_{Fix} + t_{Flex})_{src} \geq (t_{Fix} + t_{Flex})_{int1} \geq (t_{Fix} + t_{Flex})_{int2} \cdots \geq (t_{Fix} + t_{Flex})_{dest} \tag{7}$$

where t_{Fix} is the fixed delay and t_{Flex} the variable delay encountered at a node and t_{intX} represents an intermediate node in the network.

Theorem 9. *The variable delay encountered by the body of the packet does not contribute to network latency.*

Expression 7 restricts movement of the flits to a pipelined fashion, that it that a flit cannot overtake a preceding flit of the same packet in the network. This highlights the situation that although body of the packet is not involved in decision or queueing delays they may not progress faster than the header and as such may require to wait at a node until the header completes. As such the body encounters both a fixed and variable delay where the variable delay reflects the waiting time for the header to complete at the current node.

As soon as the header moves, assuming that the fixed delay of the flits of the body is less than the node delay, all the flits of the body follow and as such the delay of the header is not compounded by the number of flits in the packet and as such effect of this delay on the flits is hidden by the pipelining effect. When the header reaches the destination, it can no longer add an extra delay to the following flits and as such the delays incurred by the flits at the final hop include only the fixed part of t_{prop} and as such equation 4 holds. However, it should be borne in mind that although the latency expression for the body is unaffected, the body will reside in the network longer due to the delay at the header and as such will effect contention and thus throughput in the network.

The ideal case is off-course that $t_{Fix} + t_{Flex}$ of node $\mathbf{i}$ is equivalent to $t_{Fix} + t_{Flex}$ of node $\mathbf{i+1}$, similar to a regular pipeline where an efficient pipeline is one in which each task is of equal length such that no task needs to wait on the proceeding task to complete before it can continue and on the other hand that a task is ready to continue as soon as the proceeding task continues i.e. keeps the pipeline full.

5 Network Control

An important feature missing from the previous discussion is control. Without some form of control in the network, the sending node has no way of knowing if the receiving node is ready to receive. Control is usually implemented as request/acknowledge lines or control codes. In this work we will consider the case of request/acknowledge lines and assume that buffering is provided as one-flit output buffers.

Theorem 10. *The delay of request signalling is overlapped with data transfer whereas acknowledge signals are pipelined back along the data path, compounding the delay at each hop.*

We assume that a request line is driven high after data is placed on the outgoing channel, that is after data is valid. Therefore, when the request is received at the neighbouring node the data is already available at the input. When the request is accepted, the data may be either buffered at the input or traverses the router, buffering at the output. In this work we assume output buffering.

As Req signals are sent parallel to the data, no additional delay is encountered by the req. On acceptance of the data by node(i+1), an acknowledge signal is returned to the previous node(i). This frees node(i)'s output buffer and allows it to accept data from node(i-1).

At node(i+1) processing of the input data continues while the acknowledgement signal is sent to node(i). As such, the signal is again overlapped by data movement and is such no additional delay is incurred at node(i+1) due to the acknowledgement signal. However, the time taken for the acknowledgement to arrive at node(i) delays acceptance of data from node(i-1) at node(i).

The time taken from sending to receipt of an acknowledgement can be estimated as t_{wire}. Therefore, the data at node(i-1) is delayed by t_{wire}. However, although data is pipelined through the network i.e. forwards, acknowledgements are pipelined backwards along the path of transmission. The delay is accumulated along the path and is therefore dependent on the path length.

Theorem 11. *In wormhole routing, in the presence of one-flit output buffers and request/acknowledge signalling, each of the flits of the body contribute more than the delay of the final hop to the network delay.*

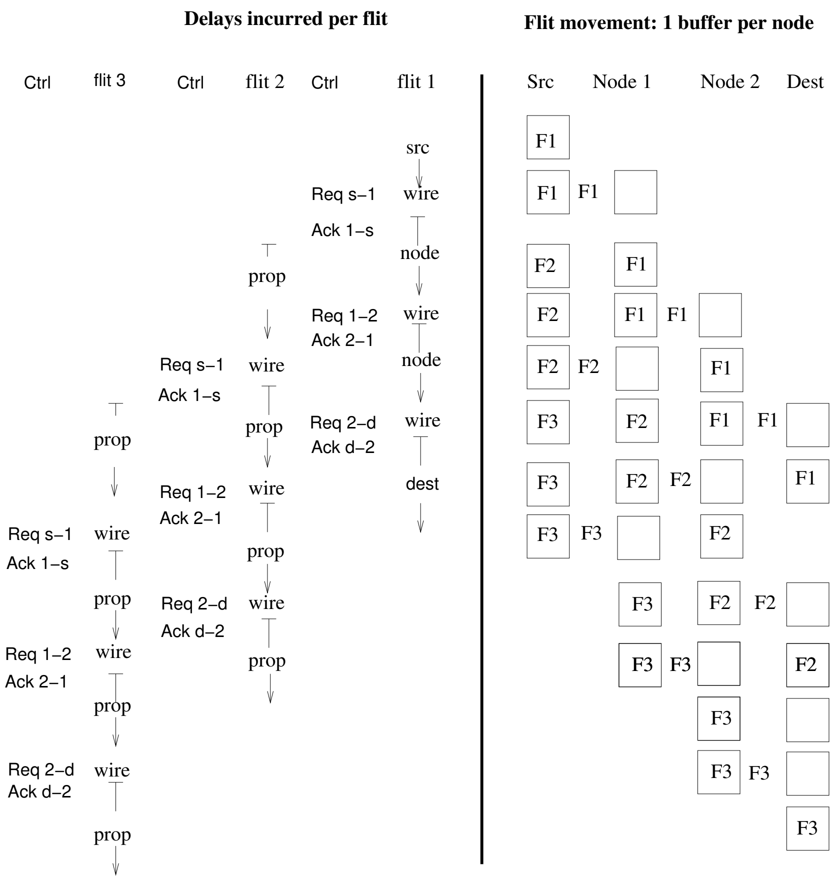

Fig. 2. Wormhole Routing - One flit buffer, 3 flit packet

From figure 2 it can be seen that as flit it i is being processed at the destination, Node 2 receives ack from the destination, thus freeing Node 2's buffer and allowing Node 1 to forward flit *i+1* to Node 2. Flit *i* completes after t_{dest} or t_{prop} depending on whether it is the head or one of the following flits. Flit *i+1* on the other hand, must wait on the ack before it can send a request to Node 2, be processed at Node 2 and proceed to the destination. As such flit *i+1* encounters 1 t_{wire} for the ack, 2 t_{wire} for transmission between Node 1-2 and Node 2-dest, and 2 t_{prop} for the processing at Node 2 and the destination. As such flit *i+1* is delayed $3 \times t_{wire} + t_{prop}$ with respect to flit *i*.

Definition 12 refines equation 4 to reflect the effect of control signalling, for the case of 1-flit buffering at each node, assuming that the time taken for an acknowledgement to be received by a preceding node is t_{wire}

Definition 12. The delay experienced by the body of the packet in a wormhole-routed network using req/ack signalling and 1-bit output buffers is expressed as:

$$t_{body-1flit} = (L - h) * \frac{F}{W} * (3 * t_{wire} + t_{prop}) \tag{8}$$

6 Summary and Future Work

In this work we have studied the features of wormhole routed networks and in particular the design of the router to see which components should be included in an expression for latency such that performance analysis models can be enhanced. We have considered buffering, control, decision and queueing features and their effect on the expression for latency in a loaded network. Changes in the latency equations result in variations in contention patterns seen by the model. The resulting model enables a more realistic analysis of wormhole routed networks in the light of network traffic.

There are, however, many unsolved problems and thus the equations presented require further refining. The work presented has limited the buffer size to a one-flit buffer. We are currently investigating the effect of various buffer sizes, aiming to represent buffer size as a component of the latency expression. Also consideration has only been given to request/acknowledge signalling. The effect of control codes should also be analysed and if possible refine the expression for this case.

// 7 Acknowledgements

I would like to thank Lasse Natvig for his continuing support and advise in this ongoing project.

References

[Aga91] A. Agarwal. Limits on Interconnection Network Performance. *IEEE Trans. on Parallel and Distributed Systems*, 2(4), Oct. 1991.

[BP93] S. Balakrishnan and D. K. Panda. Impact of Multiple Consumption Channels on Wormhole Routed k-ary n-cube Networks. In *Proc. of the Intern. Parallel Processing Symposium '93*, pp 163–167, 1993.

[Dal90] W. J. Dally. *Network and Processor Architecture for Message-Driven Computers.* MorganKaufmann Publishers, Inc., 1990.

[MT90] J. E. Midwinter and M. G. Taylor. Optoelectronic Interconnects in VLSI : The Reality of Optical Computing. *IEEE LCS*, pp 40–46, May 1990.