EE482B: Advanced Computer Organization Interconnection Networks

Lecture #1:	Wednesday, April 4, 2001
Lecturer:	Professor William J. Dally
Scribe:	Andrew Chang
Reviewer:	Kelly Shaw

1 Administrivia, Course Overview & Policies

This class covers the architecture and design of interconnection networks. Specific topics include network topology, routing, flow control, deadlock and livelock, router microarchitecture, network performance analysis and reliability. There is no overlap between this class and EE 482A; both EE 482A and EE 482B may be taken for credit. The estimated workload is nine (9) hours per week (3 in-class, 6 other).

Class webpage: http://cva.stanford.edu/ee482b
Detailed schedule: http://cva.stanford.edu/ee482b/schedule.html
More administrivia and policy details: http://cva.stanford.edu/ee482b/info.html

1.1 Class Staff

The staff for the class consists of:

- Lecturer - Professor William J. Dally (billd@csl.stanford.edu)
- Teaching Assistant - Kelly Shaw (kashaw@cva.stanford.edu)
- Support Staff - Pamela Elliott (pamela@csl.stanford.edu)

1.2 Class Time & Location

Both the time and location of the class have changed due to a conflict with EE 384Y (Packet Switch Architectures II). All subsequent classes will meet from **11:00AM-12:15PM** Mondays and Wednesdays in **Skilling 193**.

1.3 Requirements

Grades in the class are based on six components:

1. Three homework assignments **(15%)**.

   Note: teams of up to four may collaborate on each homework assignment. Each group should submit only one homework, listing group members and also acknowledging any additional outside assistance.

2. In class midterm exam on May 9 covering material up to and including the May 7 lecture. **(20%)**.

3. Class Participation **(10%)**.

4. Lecture Scribing **(5%)**

   Each student must be the class scribe (or one of a team of scribes) for at least one lecture. See Kelly Shaw to sign-up for a scribing slot.

   Scribing details: **http://cva.stanford.edu/ee482b/scribe.html**.

5. Research paper **(20%)**

   The paper guidelines will be issued on April 11. Each paper will survey a specific topic in the field and report back to the class. The research paper is due May 16.

6. Project **(30%)**

   The project this year is similar style and scope to the project assigned in EE 482A. Students will pick a research topic from a list of possible projects (or propose their own, after consultation with the teaching staff). After identifying the research topic, a team of up to four will conduct a self-contained research project with the goal of achieving publishable results. The project will have three milestones/checkpoints: proposal (May 14), midpoint check (May 21) and final submission (due date June 4).

1.4 Collaboration Policy

In this class, four of the six requirements (homework, scribing, research paper and project) may be completed as a collaborative effort with other students (stated limit is four, but five is possible with approval from teaching staff). In all cases the team of students should make one submission and **explicitly acknowledge and attribute by name any and all outside assistance.**

1.5 SITN and Remote Student Policy

This class is taught with **NO** delay. Due dates are identical for both on-campus and remote students. The method for remote students to gain participation points is currently TBD, please contact Professor Dally to work out a procedure. All remote students in the Bay Area are expected to show-up on campus for the May 9 Midterm exam. The class is available through Stanford Online and it is also shown on Stanford cable (tape delayed - channel E2 (54) from 6:45pm - 8:00pm).

1.6 Text

The main text for the class is Professor Dally's manuscript in preparation. The first four chapters were available in class and are on the webpage. Please note that these pre-release pages are copyright protected (despite their lack of explicit copyright notice). While the class is self-contained, students who wish to have a supplemental reference should use the text *Interconnection Networks* by Duato, Yalamanchili, and Ni. Note, however, that this book is currently out of print. Professor Dally suggested looking on eBay. [scribe's note - a recent check showed no volumes for sale on eBay, best bet is borrow/buy from a friend].

1.7 Miscellaneous

The following additional announcements were made:

1. All students must subscribe to the class mailing list

   email majordomo@lists.stanford.edu and include 'subscribe ee482b-students' in the body

2. All students must stop by Kelly Shaw's office during office hours to have their picture taken. All pictures will be posted on the class webpage to allow Professor Dally to begin to associate names with faces and to facilitate interaction between students in the class.

3. A grader is needed for the class. In exchange for grading, the student will receive a perfect homework score on all three homeworks as well as pay. If you are interested please contact Kelly Shaw.

2 Three Key Questions Answered By Class

There are three major questions answered by this class:

1. What is an interconnection network?

2. Where are interconnection networks used?

3. Why are interconnection networks important?

2.1 What is an interconnection network?

An interconnection network connects a set of input terminal nodes to a set of output terminal nodes. A terminal node is either the source or sink of a message. A route starts from one terminal node, runs through the interconnection network and ends in another terminal node. Switching nodes reside within an interconnection network.

Interconnection networks exist on many scales. This course focuses on small scale interconnection networks (within a board, or a cabinet or a machine room). Other courses explore LAN's and WAN's. Significantly different constraints exist on interconnection networks at various scales, resulting in very different design choices.

2.2 Where are interconnection networks used?

Interconnection networks have a wide range of applications. Within computer systems, they are used to connect processors to memory, and I/O controllers to I/O devices. They are also used to connect input ports to output ports in communication switches and network routers. Historically each of the above applications employed a simple multi-drop bus. However, more and more systems are switching to high performance point-to-point interconnections to meet the steadily increasing performance requirements enabled by semiconductor scaling.

2.3 Why are interconnection networks important?

Interconnection networks are important because they transport data between system components, and, thus are, by definition, on the critical path.

2.3.1 Example Applications

Professor Dally provided three example uses of interconnection networks: processor-memory, packet switch fabrics, and I/O systems.

2.3.2 Processor-Memories Interconnect

The key characteristics of the Processor & Memories application are that latency is critical, the system is self-throttling, a significant difference between average and peak bandwidth exist and no packets can be dropped. Processors & Memory is a demanding application of interconnection networks.

There are two basic approaches to interconnecting processors and memory: "Dance Hall Archiecture" and "Combined Architecture". The "Dance Hall Architecture" places processors on one side and memory on the other side of the interconnect. Processors then send packets (read/write requests) through the interconnection network to the memory

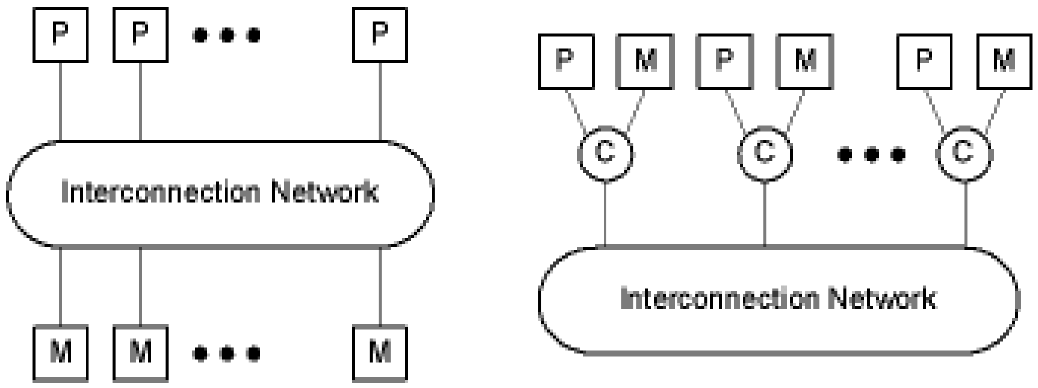

Figure 1: Processor and Memory Interconnect

Parameter	Value
Processor Ports	1-2048
Memory Ports	0-4096
Peak Bandwidth	4GBytes/s
Average Bandwidth	100MBytes/s
Message Payload Size	64 or 512 bits
Latency Sensitivity	100ns
Congestion Control	Self Throttled
Message Loss	Unacceptable
Traffic Patterns	Arbitrary

Table 1: Parameters of Processor-Memory Interconnection Networks

and the memory returns data back through the network. In this architecture, all processors are equally distant from each piece of memory. An alternate approach is the "combined architecture". In this approach, some memory is "local" to the processor while most memory is "remote". The processor can access local memory without going through the interconnection network. Most modern systems employ this integrated-node approach. Both configurations are shown in Figure 1 above and also in Figure 1.2 of the text

Note: As an interconnection network designer, the important parameters are how many ports and how much traffic each generates and not the detailed nature of each port (ie. whether an Alpha or a Pentium IV microprocessor is located at the port).

The typical parameters for a Processor-Memory interconnection network are provided above as well as in Table 1.1 of the text.

There can be a wide difference between the peak and average bandwidth seen by the processor-memory interconnection network. We can take advantage of this disparity

Parameter	Value
Device Ports	1-1024
Memory Ports	1-64
Peak Bandwidth	100MBytes/s
Average Bandwidth	5MBytes/s
Packet Payload Size	32Kbits
Latency Sensitivity	100μs
Congestion Control	Self Throttled
Packet Loss	Unacceptable[1]
Traffic Patterns	Arbitrary

Table 2: Parameters of I/O Interconnection Networks

by using concentrators to share bandwidth and reduce design costs by provisioning for a new peak bandwidth that is closer to the average bandwidth. However, we can not just design for average bandwidth since bandwidth directly effects *serialization latency* (see Section 6.0 for definitions), one of the three components of overall latency. The latency of the interconnect is critical as processors will stall, wasting valuable computing cycles, waiting for the return of memory requests. Also, we can not drop any packets as a processor will hang and potentially crash the system if it never receives a reply for its request.

2.3.3 Packet Switching Fabric

The key characteristics of the Packet Switching Fabric application are that latency **does not** matter, the system is **not** self-throttling, average and peak bandwidth are not very different, and packets can be dropped. Packet switching fabric is a demanding application of interconnection networks. The typical parameters for a Packet Switching Fabric interconnect are provided above in Table 2 (and in Table 1.3 of the text).

As noted above, there is much smaller variation between average and peak bandwidth requirements in the packet switch application. New traffic continues to arrive regularly at the input ports. Thus, concentrators are not an effective solution.

2.3.4 I/O Systems

The key characteristics of the I/O system application are that latency does not matter, the system is self-throttling, relatively low bandwidth requirements and packets can **not** be dropped. I/O systems are a relatively un-demanding application of interconnection networks. Even in I/O systems, multi-drop busses are starting to become an insufficient solution. Infiniband seeks to address this by providing a specification for I/O interconnection.

3 Professor Dally's Background

Professor Dally has designed a number of interconnection networks and provided a brief overview of some of his past projects.

3.1 Mars Router

The Mars router is a 16x16 bit sliced crossbar designed as a part of a hardware accelerator for logic simulation at Bell Labs in 1984 and fabricated in 2.5μm CMOS. Notable characteristics: crossbar bitsliced 2bits/chip, 100MHz operating frequency, source routed, and arbiters employed within the crossbar instead use of centralized scheduling.

3.2 Torus Routing Chip

The Torus Routing Chip (TRC) was designed and built at Caltech in 1985. The TRC implemented a 2-D torus network, contained 8bit channels and operated at 20MHz. It demonstrated both worm-hole flow control and virtual channels. Internally, the 5-input controllers (X+,X-, Y+,Y-, to/from node) were connected via a crossbar.

3.3 Message Driven Processor

The Message Driven Processor (MDP) was designed at MIT (completed in 1991) and fabricated in 1.2μm CMOS. The MDP integrated processor, memory and a router on a single chip. The integral router enabled the construction of a 3-D mesh network with 18bit channels and operating at 32MHz.

3.4 J-Machine & Cray T3D

The J-Machine is a 1024-node multicomputer created from an 8x8x16 mesh of MDP's and built at MIT in 1991. The importance of pins and packaging was one of the key lessons from this project. The use of elastomeric connectors to provide vertical interconnects between printed circuit boards reduced the maximum wire length in the system to 5-inches and enabled the packaging of the complete 1024-node system in a 2'x2'x1' volume.

The Cray T3D was introduced in 1993 and employed a 3-D torus network with dimension-ordered routing that was very similar to the J-Machine's network. It used 16-bit channels, featured four virtual channels per physical channel, was implemented in ECL gate arrays and operated at 150MHz.

3.5 Reliable Router & Cray T3E

The Reliable Router (RR) chip was designed at MIT in 1995 and fabricated in 0.8μm CMOS. The RR implemented a 2-D torus with 32b channels operating at 200MHz. It included both adaptive routing, using Duato's algorithm, and fault tolerance, through the Unique Token Protocol (UTP). The RR employed link level retry and the UTP guaranteed that no duplication occurred. In addition, the RR employed simultaneous bi-directional signalling and a low-latency plesiochronous synchronization mechanism.

The Cray T3E, introduced in 1996, employed a 3-D torus network that was very similar to the Reliable Router.

3.6 Avici TSR

After completing his Ph.D., Larry Dennison, the lead graduate student in the Reliable Router project, founded Avici Systems in 1996. Avici's first product is the Terabit Switch Router (TSR). The TSR supports up to 2240 OC-48 (2.5Gb/s) or 560 OC-192 (10Gb/s) connections. The TSR implements a 3-D torus, employs randomized oblivious routing. Within a line module the design is partitioned into router and traffic manager (Professor Dally noted that in retrospect, this was not the best choice). The design is implemented with 6 gate-arrays. The TSR uses 24bit channels organized as 4b bitslices.

4 The Basics of Interconnection Networks

4.1 An Analogy

All networks can be described in terms of their topology, routing and flow control. A useful analogy to interconnection networks is planning a cross-country road trip.

- Topology (get a map) - An interconnection network consists of two sets: channels (C), which carry traffic and have a certain bandwidth - roads in the map analogy; and nodes (N), which provide connections between channels and provide storage - junctions with parking lots in the map analogy. The "parking lots" are needed to provide storage when channels are unavailable.

- Routing (find some path) - A path is a specific ordered set of channels

  P = (C1, C2, C3, ... C*); The destination of C2 is src of C3.

- Flow control (what to do with traffic jams) - A packet is too large a unit to meter. Instead, a packet is broken down into flow control units (flits), where only the first flit carries the address. The head flit leads and leaves state in each routing junction. The other flits are merely associated with the flit ahead of them. The advantage of metering flits instead of packets is reduced storage requirements for the junctions and a reduced probability of collisions.

  Three common flow control mechanisms are:

  1. mis-routing (fancy term for being impatient - "if I'm moving it must be progress")

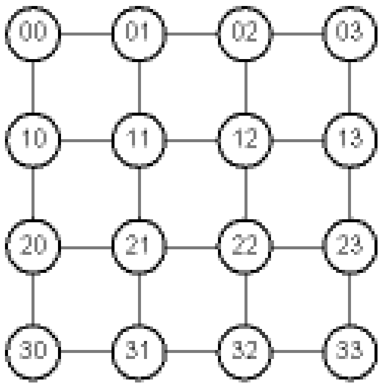

Figure 2: 4-ary 2-mesh

When mis-routing, one may route in a direction other than the intended direction, even if the selected direction moves further away instead of toward the final destination. The Connection Machine employed mis-routing when resources aren't free.

2. dropping (throw away the packet - this is generally bad)
3. buffering (park and wait)

Most of the flow control discussion will be focussed on how to allocate buffers (many schemes exist).

4.2 Example Topology

A 4-ary 2-mesh is shown above (and in Figure 1.3a of the text). Each edge in the figure represents two channels - one in each direction. This is a radix-4, 2-dimensional network and consists of 16 nodes organized in a 4x4 mesh. There are a total of 96 channels (4x8 + 8x6 + 4x4). In contrast, a 4x4 torus would have 128 (16x8) channels due to the additional "wrap-around" channels.

4.3 Route Diversity

Route diversity means that there are multiple paths from an input to a selected output and is a characteristic of a good topology. The selected topology determines the amount of route diversity. The chosen routing algorithm determines the extent to which the available route diversity is exploited. The ability to select alternate paths to complete a route provides fault tolerance, enables traffic to be carried despite local congestion and facilitates load balancing by "spreading out" hot spots.

Number of input ports	64	
Number of output ports	64	
Port width	2	bits
Port data rate	1	GHz
Port duty factor	1.0	
Traffic pattern	random	
Packet size	4-64	bytes
Packet dropping is acceptable		

Table 3: Specifications for our example network

Signals per Chip	150	
Chip Cost	$200	
Chip pin bandwidth	1	Gb/s
Signals per Board	750	
Board Cost	$200	
Signals per Cable	80	
Cable Cost	$50	
Cable length	4	m at 1Gb/s

Table 4: Constraints for our example network

5 A Simple Design Example

Professor Dally selected a simple 4-ary 3-fly (radix-4, 3-stage butterfly) network as an initial illustrative example. Two motivations exist for this selection - the design is easy to explain and it is easy to improve upon the performance of this network with our subsequent designs. In general, the selection of a topology is a careful balance of mapping the design specifications onto the technology constraints.

The specifications and constraints for this design are provided below in Tables 3 & 4 (and in Tables 2.1 and 2.2 of the text respectively). The number of pins per chip (150) is the key constraint determining partitioning. Also, the network employs dropping flow control.

The topology ("map") of the network is illustrated below (and in Figure 2.2 of the text). Due to the selection of a butterfly network, the 1GByte/s required by the input and output ports combined with the duty factor of 1.0 (ie. all the time), results in a maximum bandwidth for each individual network channel of only 0.25GBytes/s. Due to the selection of dropping flow control we need a large speed-up. We will find that in a three stage network with dropping flow control our peak throughput will be only 25% of capacity. A speed-up of four (4) would, in theory, allow us to meet the 1GByte/s requirement however, we select a speed-up of eight (8) as it allows additional headroom and reduces the variance in latency experienced by different packets.

With speed-up (S) of eight, and 150 signals per chip, we get 16 signals per channel

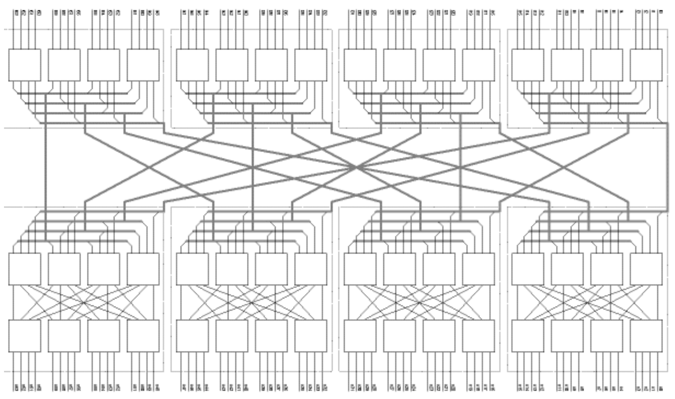

Figure 3: 4-ary 3-fly

(rounding to nearest multiple of 2). With 750 channels per board and 16 signals per channel, we can get 46 channels per board. Again rounding to the nearest multiple of 2, we get 32 channels per board (ie. 4 chips). The node degree (d) is nine (150/16) and the degree of the network (k) is four (d/2).

5.1 Routing

The selected network is destination tag routed. This means that "the bits of destination address are used to select the output port at each stage of the network". A property of k-ary n-fly networks is that the route through the interconnect network is completely determined by the choice of destination and, thus, butterfly networks have *no* path diversity. This is one weakness of k-ary n-fly networks as path diversity (ie. multiple available routes from a source to a specific destination) is advantageous for load balance, congestion and fault tolerance.

Routing within a k-ary n-fly network occurs as follows: the first stage routes packets to the correct Nth of the network, subsequent stages continue to route the packet to the correct Nth within their section, with N always equal to the radix. For example, in a 4-ary 3-fly, the first stage directs traffic to the correct quarter, the second stage further directs traffic to the correct 16th (1/4 x 1/4) within the network, and, finally, the third stage sends the packet to it's final destination - the unique 64th of the network.

As an example, suppose that one would like to route from node 42 to node 21 (6'b01_10_10 to 6'b01_01_01). Assuming the output ports of each switch are numbered 0, 1, 2, and 3, the destination tag, 6'b01_01_01 dictates that the packet use the "1" port in each of the three stages of the butterfly.

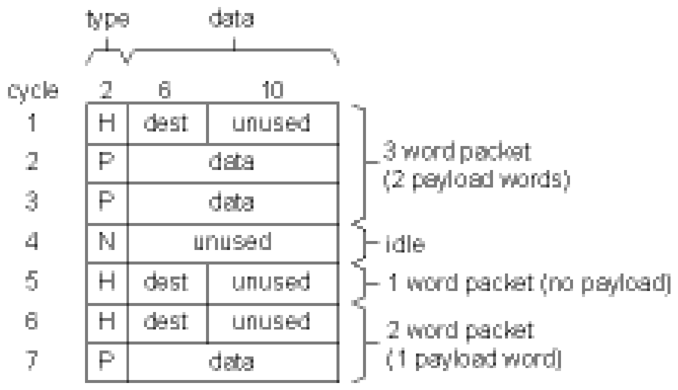

Figure 4: Basic packet format

5.2 Flow Control

Basically, flow control determines what action occurs when two packets collide. A very simplistic approach is that one wins and the rest are dropped and must be retransmitted. However, before we can determine if two packets collide, we need to define the structure of a packet. The network transports physical digits or *phits*. A packet consists of a set of phits. A basic packet format is shown in below in Figure 4 (as well as in Figure 2.3 of the text). The payload in this format is 16 bits wide, but the packet adds another two bits to enable the encoding of head (H), payload (P) and null (N) phit. Tail is not explicitly coded, instead it is implied by a P phit followed by either an H phit or a N phit.

5.3 Putting It All Together

The block diagram of the simple router is presented in Figure 5 (Figure 2.4 from the text). The basic operation of this router is as follows. Phits arrive and are clocked into the input latch and then distributed to all four multiplexors. At each multiplexor, an allocator examines the type of each phit and sets it's switch based on next hop field of any seen head flit. Enabled packets are then sent to the shifter which rotates the address in the header flit to discard the just-used bits and align the address in the head flit for the following stage. Once a channel is allocated it is held for the duration of a packet. Thus, the allocator also includes state bits that remain set while payload phits are being presented to the input port. When more than one input port requests an output port, the allocator relies on a fixed priority encoder to determine the winner. The schematics for the described allocator is shown in Figure 6 (Figures 2.5 from the text). Also, the corresponding verilog is show in Figure 2.6 in the text.

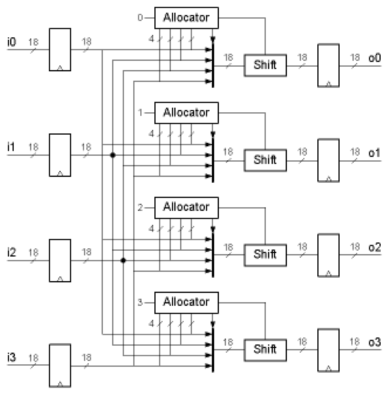

Figure 5: Block Diagram of simple router

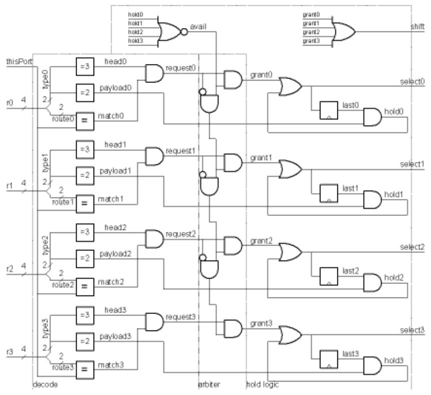

Figure 6: Schematics of Allocator

5.4 Performance

The performance of an interconnection network can be characterized by its throughput and latency. Figures 7 and 8 below (Figures 2.8 and 2.9 in the text) show the behavior of our simple example.

The Figure 7 shows how throughput changes as we increased offered traffic from zero to one. If we did not drop packets the throughput curve would be a straight line. But since we have selected dropping flow control, as we increase offered traffic, packets are dropped and then must be re-transmitted. Because our speed-up is 8, our offered traffic can never exceed 1/8. At input of 1/8, we yield less than 1/8, the difference will be re-transmitted and the steady-state will occur when the output is 1/8 and the input is approximately 1/5. The saturation point occurs when only 25% of offerred traffic get through with the rest colliding requiring retransmission. The shown behavior is completely unacceptable as most of the cost of the interconnection network is in the bandwidth (channels). In the example configuration these are severely under utilized as they operate at only 25%

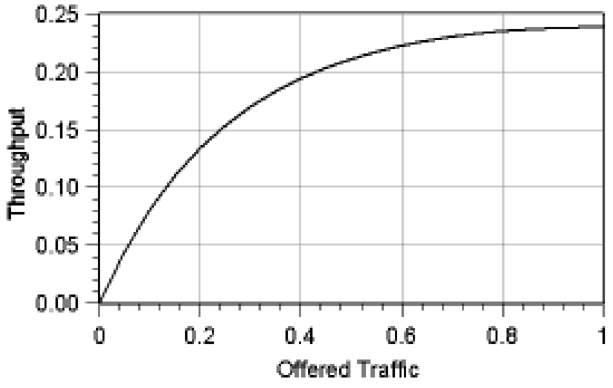

Figure 7: Throughput

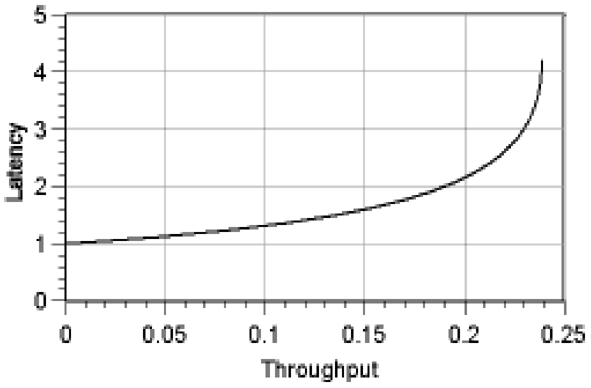

Figure 8: Latency

of capacity. In subsequent portions of the class, we will explore how to increase the utilization to over 90% by adding additional design complexity.

Examining Figure 8 shows that as we approach the saturation (peak) throughput, our latency goes to infinity. With light traffic, the latency is just the minimal number of hops to traverse the router and the network. As traffic increases, a percentage of the packets collide and require retransmission. In our example the peak throughput is 0.25 and as we approach this condition, (where only 25% of the packets get through and 75% of the packets require retransmission) the latency approaches infinity. However, if we back-off slightly from this case, we find that the latency bound is approximately four times that of the zero load (minimal traffic case).

Note, with a properly designed network whose channel utilization approaches 100%, the saturation throughput also approaches 1 (instead of the 0.25 for our example).

6 Definitions

- Ideal Throughput(Θ_{ideal}) : Defined as the ratio between the network channel bandwidth to maximum channel load.
- Non-blocking: A circuit property, a path can always be established from any free input to any free output port, regardless of the configuration of other paths in network. Behavior similar to a fully connected crossbar.
- Non-interference: Any input with available bandwidth can send packets to any output with available bandwidth without interference from packets to other, possibly oversubscribed, ports.
- Route Diversity: Multiple routes from a input port to desired output port. Important for both load balancing and fault tolerance. offered traffic (γ_{max}).
- Speed-up (S): Defined as the ratio of (Θ_{ideal}) to the offered traffic (γ_{max}).
- Self-throttling: The sources of traffic will stop injecting new messages when the interconnection network is congested. Keeps network from becoming unstable.
- Serialization Latency (T_s): Ratio of length of message (L - in bits) to bandwidth (b - in bits/second) of the channel used for transmission.
- Wormhole routing(flow control): Messages are segmented into flow-control units (flits). Flits move from node to node through the network. When a message is blocked, flits of the message are buffered in their current node.