Performance Analysis (cont'd); Reliability

Lecture #13: Monday, May 21, 2001
Lecturer: Professor Bill Dally
Scribe: Melvyn Lim, TJ Leong
Reviewer: Kelly Shaw

1 Performance Analysis (cont'd)

1.1 Dropping Flow Control

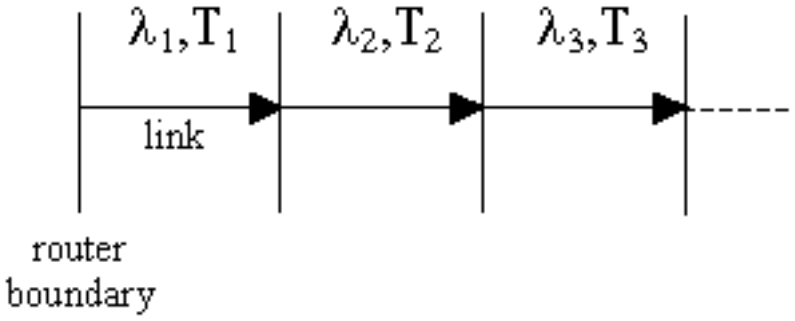

Figure 1

Figure 1 shows one path of a network without feedback. At each stage (link), there is a service time, T_i, and a traffic rate, λ_i. Since packets are dropped along the path in dropping flow control, T_i is constant at each stage while λ_i decreases along the path in the direction of transport. The duty factor of traffic at stage i, ρ_i, is $\lambda_i T_i$. We can normalize the duty factor to T such that $\rho_i = \lambda_i$.

When a packet arrives at a router and tries to get the outgoing channel, it could be blocked in one of two ways. First, it is blocked if the body flit of another packet is already on the outgoing channel. This is shown in Figure 2, where packets are assumed to be L flits long. This blocking happens if any body flit - each packet has L-1 body flits - of any packet - there are packets from the other k-1 inputs of the router - are on the outgoing channel, which has duty factor ρ_{i+1}. The probability of being blocked by the body flit of another packet is:

$$P_{BB} = \rho_{i+1}(\tfrac{k-1}{k})(\tfrac{L-1}{L})$$

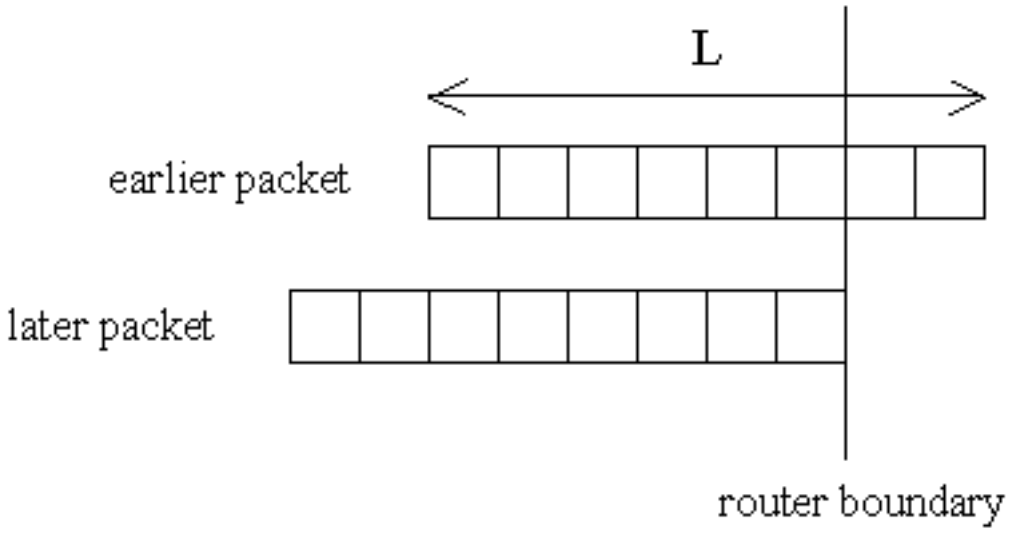

Figure 2

Second, the packet is blocked if it tries to get the outgoing channel at the same time as one or more other packets and loses the contention. For contention to take place, the outgoing channel must be in the idle state. This happens when the outgoing channel is not occupied by a packet (i.e. not blocked by a body flit), with probability $1-P_{BB}$. Assuming (pessimistically) that the packet always loses contention, it would not be blocked by another packet only if it was the only one to have requested the outgoing channel at that instant, with probability $1-(1-\frac{\rho_i}{k})^{k-1}$. Thus, the probability of being blocked by the head flit of another packet is:

$$P_{HB} = [1 - P_{BB}][1 - (1 - \tfrac{\rho_i}{k})^{k-1}]$$

The total probability of a packet being blocked is the sum of P_{BB} and P_{HB}. Hence, the duty factor of the next stage channel is:

$$\rho_{i+1} = \rho_i(1 - P_{BB} - P_{HB})$$

Note that for very long packets, the $\frac{L-1}{L}$ term can be eliminated from P_{BB}, and P_{HB} can be ignored. For a packet that is only 1 flit long, $\frac{L-1}{L} = 0$ and therefore P_{BB} can be ignored.

1.2 Infinite Queues

Suppose there is a multi-hop network with infinite queues at every stage. The traffic rate, λ_d, and service time, T_d, remain constant at every stage. The only quantity of concern is the wait time of a packet in a queue. To calculate this, we need to know the occupancy of the queue. We make approximations on the arrival rates and service

times of the queue. If the arrival intervals and service times associated with the queue are exponentially distributed, we get an M/M/1 queue, and the mean wait time for the queue is $\frac{\rho}{1-\rho}T$. This gives an exponentially increasing wait time as a function of ρ. If latency needs to be minimized, the network would have to be operated at a low traffic rate. In this approximation, wait times are exponentially distributed.

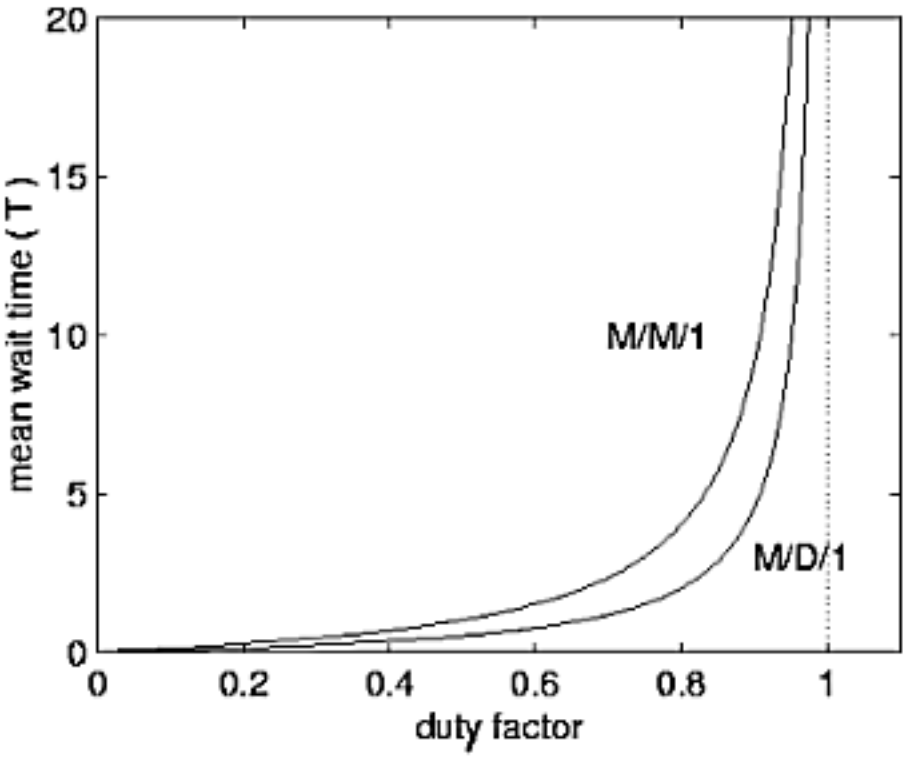

Figure 3

In many networks, service times are not exponentially distributed. For example, memory access times are usually of a fixed length. This results in a deterministic service time distribution, which gives an M/D/1 queue. The mean wait time for an M/D/1 queue is $\frac{\rho}{2(1-\rho)}T$. The mean wait time (in multiples of T) as a function of ρ is shown for both M/M/1 and M/D/1 queues in Figure 3. In general, an M/G/1 queue has a mean wait time of $\frac{\rho}{2(1-\rho)}(1+c_b^2)T$, where c_b is variance. If buffers are assumed to have infinite capacity, latency can be bounded by controlling the load (ρ) or by using static scheduling.

1.3 Finite Queues

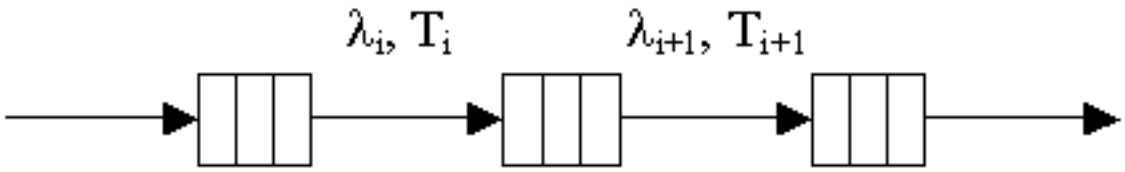

Figure 4

Suppose a queue has enough storage for only 3 packets as shown in Figure 4. It cannot be modeled as an M/M/1 queue because there is a nontrivial probability that the queue can get full, blocking subsequent incoming packets. The mean wait time for the queue and the probability of the queue filling up can be analyzed using a Markov chain of the queue size, shown in Figure 5. The queue length increases if a packet arrives, with probability λ, which is the arrival rate. The queue decreases if a packet is serviced (packets are not dropped in this model), with probability $\frac{1}{T_{i+1}}$, which is the service rate. Note that when the queue is full (state 3), the probability of increasing in length is folded back into the probability of decreasing in length.

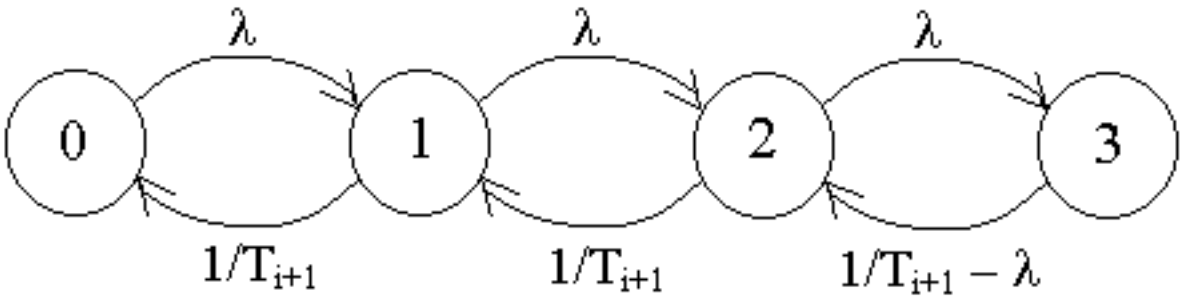

Figure 5

To compute the probabilities of being in a state, we assume that the "chance" of being in state 0, q_0 is some value, say 1. (This will be normalized later.) For i=1,...,n-1, where n is the maximum queue length, $q_i = \lambda T_{i+1} q_{i-1}$. For i=n (or 3 in this case), $q_n = \frac{\lambda}{\frac{1}{T_{i+1}} - \lambda} q_{n-1}$. The probability of being in a state i is:

$$P_i = \frac{q_i}{\sum_{i=0}^{n} q_i}$$

The service time at stage i, T_i is given by the service time at stage i+1, T_{i+1}, plus some increase in latency caused by blocking at stage i+1 due to a full queue at stage i+1. This increase in latency is given by the probability of blocking (i.e. the queue at stage i+1 is full) times the average time it takes the blockage to clear, which is $\frac{1}{2}T_{i+1}$. Therefore, $T_i = T_{i+1} + P_n\frac{T_{i+1}}{2}$.

1.4 Virtual Channel Queues

Suppose a network has links with 3 virtual channels per link, and hence 3 queues associated with each link in a router. This is shown in Figure 6, with the grayed cells representing filled buffers. The upstream router is on the left and the downstream router is on the right. The link will block if all 3 virtual channel queues in the downstream router are full. There are packets from 3 virtual channels in the upstream router contending for the link, thus the service time (of the virtual channel queues in the upstream router) tends to be longer than if there were only 1 virtual channel. This longer service time is given by T_i multiplied by a factor $\overline{j_{max}}$, where j is the number of occupied virtual channels, j_{max} is the maximum value of j along a route, and $\overline{j_{max}}$ is the average of the maximum values over many routes.

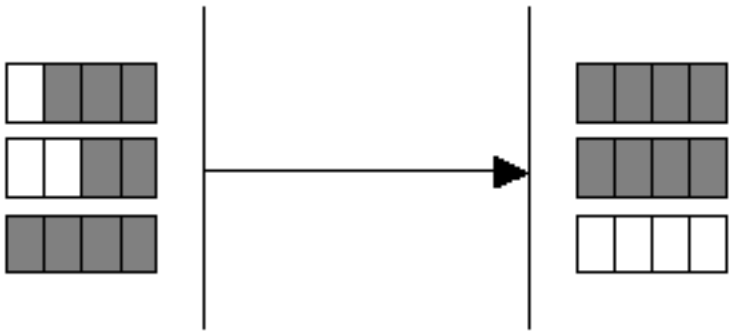

Figure 6

1.5 Aside: "Fairness is a Bad Idea"

As an aside from performance analysis, we discuss why fair queueing is not good for virtual channels. Suppose there are 2 packets, A and B, each comprising 5 flits, both arriving at a router and going on the same outgoing channel. (Let AAAAA represent packet A and BBBBB represent packet B.) With fair queueing, the sequence of flits seen on the channel is ABABABABAB. Both packets are delayed by almost the maximum amount and both tie up their virtual channels by almost the maximum amount. If a "winner take all" approach is taken, where packet A gets transmitted entirely first then packet B gets transmitted, the sequence of flits seen on the channel is AAAAABBBBB.

Packet B gets to the next router at the same time (cycle 10) as before but packet A gets there 4 cycles earlier than before. Also, packet A's virtual channel gets released 4 cycles earlier than before. This approach optimizes for bandwidth. Subsequently, packet A keeps getting ahead of B at every hop along the path, in effect delaying only the losers instead of everyone.

1.6 Cyclic Networks

Our analysis so far has dealt with networks that have clearly defined start and stop points. However, this is not the case for cyclic networks. To make sense of cyclic networks, we will have to modify our analysis method slightly and take into consideration traffic going into and out of the network. A unidirectional k-node ring example, shown in Figure 7, was given where traffic takes $\frac{k}{2}$ hops on average.

We consider each node to have an input channel from the preceding node, an output channel to the subsequent node, an entry point from outside the network to the node, and an exit point from the network, as shown in Figure 8. Traffic either exits at the node or moves on to the next node in the ring. The corresponding fraction of traffic is shown next to each arrow depicting traffic flow through the node. There is traffic coming into the node with some input service time T_{in} and rate λ_{in}, and exiting with T_{out}, λ_{out}. Traffic flowing to the next node in the ring has service time T_{int}. Using this model, we can solve for T_{int}. Interested parties may see Prof Dally for his notes on how to work out such an example.

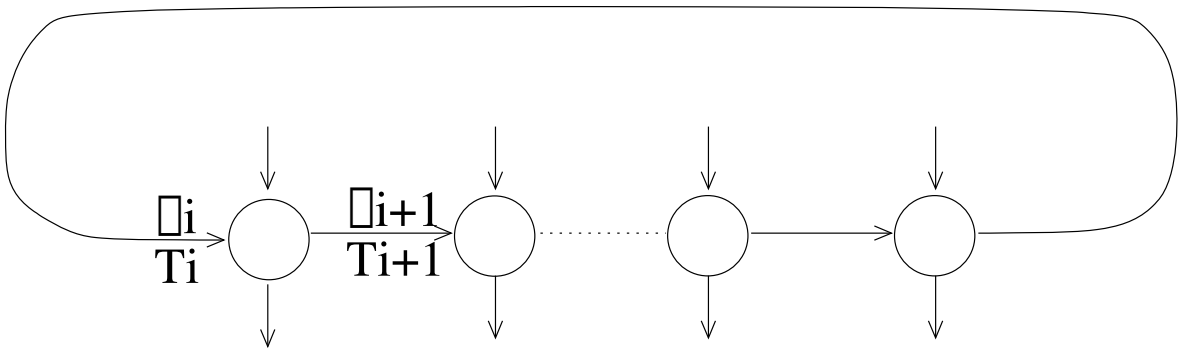

Figure 7

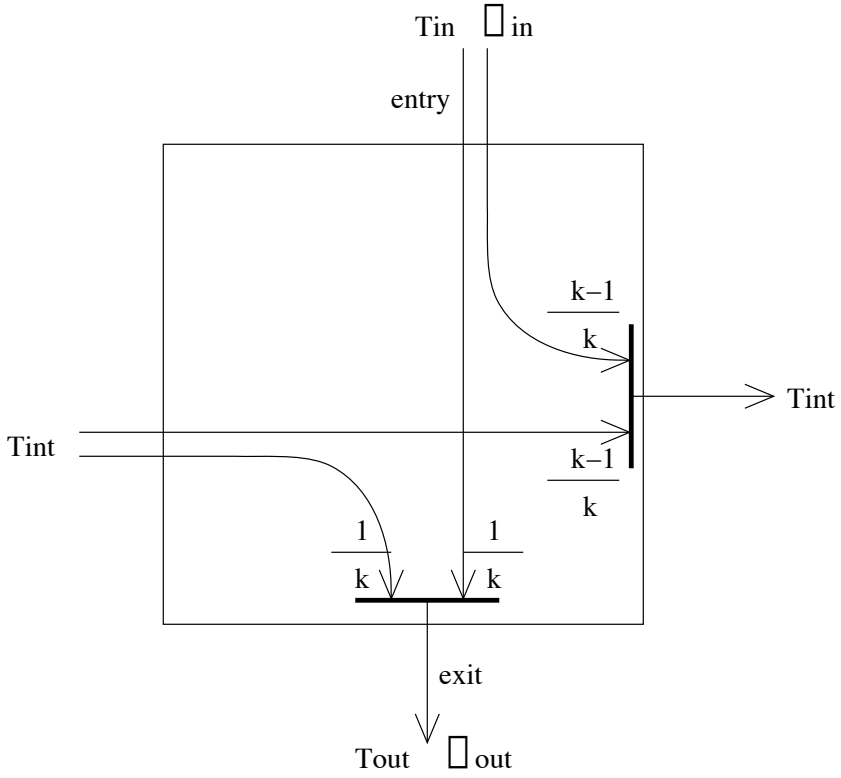

Figure 8

1.7 Common Simulation Pitfalls

Analysis is useful when dealing with ideal networks. However, it becomes difficult to perform analysis when non-ideal situations are encountered. Simulation and numerical calculation provide alternatives to analysis.

Some things to keep in mind when running simulations: 1. Give the simulator time to warm up. Data collected when the simulator is just starting up is often not meaningful because the queues in the networks are empty. Allow the queue lengths to reach steady state before recording data. 2. Watch for cases where you are attempting to measure something in steady state that is not stable. 3. Be careful to include the source queueing time when determining the network delay. Do not only consider the time that the packet is in the actual network.

2 Reliability

2.1 Definitions

There is a demand for reliable systems that can run for years with a very small probability of an error occurring. Reliability has to be built from the ground up as it is difficult to

add this feature to a system in the middle of its lifetime.

The reliability of a system at time t is the probability that the system is working at time t given that it was working at time 0, assuming no repairs were carried out.

$$R(t) = P(\text{working at } t \text{ — working at } t = 0)$$

Typically, we have a system that is working at time t and which subsequently fails with an exponentially distributed failure rate as shown in Figure 9.

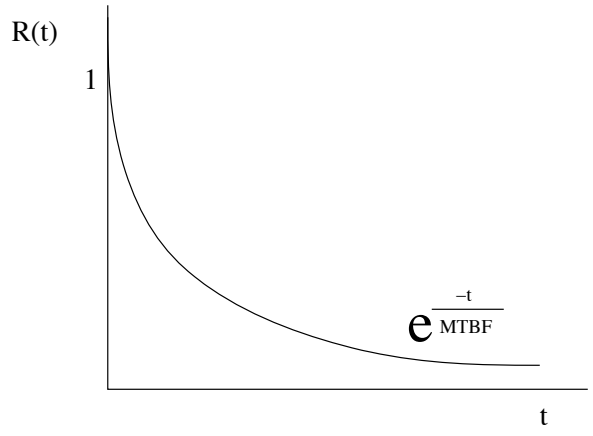

Figure 9

Real reliability curves, however, tend to dip in the early lifetime of the system, have a middle stage where the reliability levels out, and then dip again towards the latter part of their lifetime again, like in Figure 10. The Mean Time Between Failure (MTBF) is not a constant factor and usually looks like the bathtub curve in Figure 11. The high initial MTBF can be attributed to imperfect testing of the manufactured systems which is usually gotten rid of through the use of a burn-in. Generally people test to get rid of early failure rate, changing the curve so that the new location of its start is at A. The increased MTBF towards the end is a result of wear over an extended period of time.

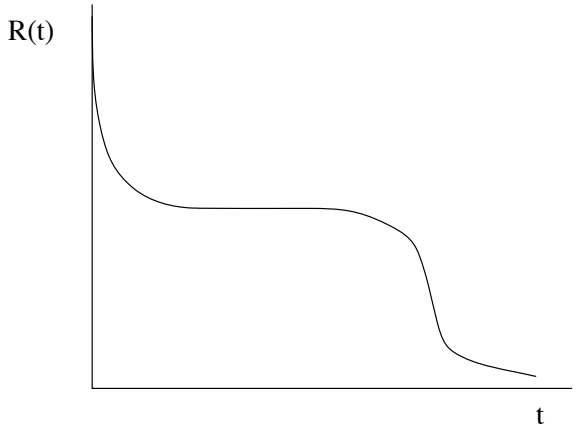

Figure 10

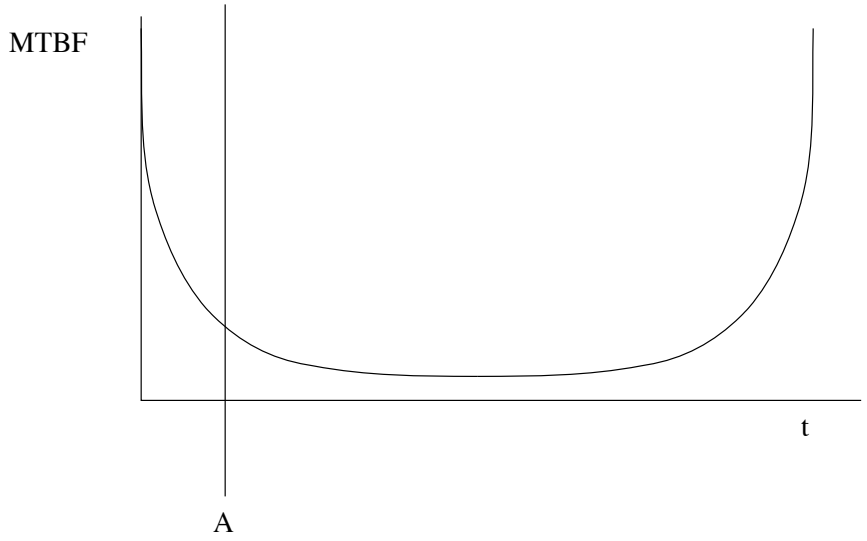

Figure 11

Availability is the percentage of time a system is working assuming repair, or simply P(working). The system is assumed to be working for a period of time equivalent to MTBF (or τ_F, the time constant of failure), and under repair for Mean Time To Repair (MTTR), so the availability is the ratio of these two metrics:

$$A = \frac{MTBF}{MTBF + MTTR}$$

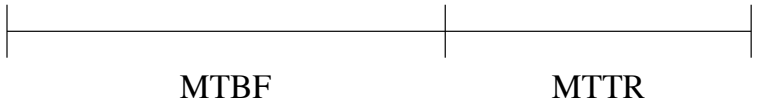

Figure 12

To build highly available systems, reduce MTTR by making repair instantaneous and having recovery system to get things up and running immediately.

A fault is the cause of a failure. Examples of faults that can occur are power outages, operator error, noise, open and short circuits, and parameter drift. Instead of considering every fault that could possibly occur in a system, engineers often make use of fault models to approximate faults and evaluate techniques to handle them.

2.2 Dealing With Faults

There are three basic steps to be taken in dealing with faults.

1. Detect the fault has occurred.
2. Contain the fault.
3. Reconfigure the system.

2.3 Fault Scenarios

There are several fault scenarios that we can consider, as numbered in Figure 13.

1. Bit flip in Body flit
2. Bit flip in Head flit
3. Bit flip in Channel Memory (Buffers)
4. Bit flip in Channel State
5. Hard failure of Channel
6. Hard failure of Node

Events 1, 2, 3, and 4 are transient failures, while events 5 and 6 are hard failures.

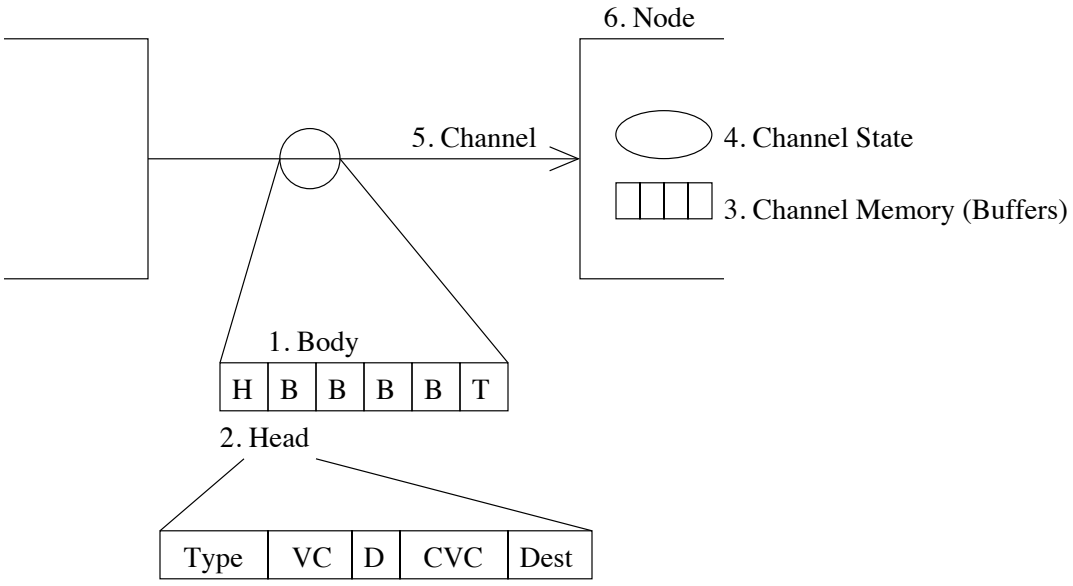

Figure 13

1. Bit flip in Body flit

(a) Detect - We can put the CRC at the flit level or at the packet level. The advantage detecting faults at the flit level is that the flit in error is not propagated, whereas if detection is done at the packet level, we are not able to detect the fault until the entire packet has been received, so the flit in error is propagated downstream. However, detecting faults at the flit level incurs more overhead.

(b) Contain - If fault detection is done at the flit level, the fault is contained by not propagating the flit in question to the next link. If fault detection is done at the packet level, the fault is contained by not propagating the erroneous packet out of the destination node.

(c) Reconfigure - This is achieved by retransmitting at the link level, if only one link has bits in error, or by retransmitting from the source, if the entire path is in error.

2.4 An Example Involving Errors

Assuming a Bit Error Rate $BER = 10^{-15}$ and with links going at $b = 10^9$ b/s, and considering a topology with 2K nodes that each have 6 channels that are 8b wide (like a 3-D torus), we find that the aggregate bandwidth of the network is 10^14 b/s. This means we only have to wait 10 s before we get an error!

To achieve an availability specification of 99.99%, assuming a repair time of 1 day (10^5 s), we need the system to stay up for 10^9 s, and require a BER of 10^-23.