Dynamic Analysis

Static (batch) processes:

- Input is fixed at the beginning of the computation.
- Goal: Minimize the time till termination.

Dynamic Processes:

- Input is continuously injected to the system.
- Goal: Characterize the long term (steady state) performance of the process.

Static vs. Dynamic Packet Routing

Static Routing:

A set of communication requests are injected at time 0.

Goal: Minimize the time to satisfy all requests.

Benchmark: Time to route an arbitrary permutation.

Dynamic Routing:

New Packets are continuously injected to the system.

Benchmark: steady state analysis.

Packet Routing on a Ring

N-node one-directional ring.

In each step, each node generates a new packet with probability λ.

Packets have random destinations (chosen uniformly at random from the N nodes).

An edge can transmit one packet per step.

Stability

$W_{avg}(t) =$ expected waiting time of packets that entered the system at time t.

$L(t) =$ the number of packets in queues at the end of step t.

A system is stable if both $E[W_{avg}(t)]$ and $E[L(t)]$ are bounded with respect to t.

Upper bound on λ

Theorem 1. *The system cannot be stable for* $\lambda > \frac{2}{N}$.

Proof. The average route of a packet is $N/2$.

There are no more than N packet transitions per step.

To keep the system stable a process can insert new packet every $N/2$ steps on the average. $\square$

Lower bound on λ

Theorem 2. *A routing algorithm with farthest-first priority is stable for any* $\lambda < 2/N$.

Packet r was delayed t steps in crossing edge e if

1. r traversed edge e at step $t + k$.

2. e is the kth edge in the route of r.

Lemma 1. *If a packet r was delayed t steps in crossing edge e, then there is an interval of t steps such that some packet crossed edge e in each step of this interval.*

Proof. Because of the farthest-first priority scheme any packet that delayed r must also cross edge e.

We say that packet s delayed packet r at time τ if s moves at that step, r does not move, and all the packets between s and r do not move.

A packet can delay r only once.

All the packets that delayed r are moving in front of r, thus crossing e in the interval $[k, \ldots, t-1]$. $\square$

The wide-channel model

In the wide-channel model packets are never delayed.

A packet crosses edge e at time k if e is the kth edge on its route.

Lemma 2. *There is a constant ϵ such that for a sufficiently large t_0 the probability that a given packet is delayed by at least t_0 steps is bounded by $e^{-\epsilon t_0}$.*

Proof. If a packet was delayed t steps then there is an edge e such that the packet was delayed t steps in crossing that edge.

This implies that there is an interval of t steps such that at least t packets crossed edge e in that interval in the wide-channel model.

The expected number of packets that cross e in the wide-channel at a given step is bounded by

$$\sum_{i=1}^{N} \frac{\lambda(N-i)}{N} < \frac{2}{N}\frac{N(N-1)}{2N} = \gamma < 1$$

Chernoff bound gives the result. $\square$

Theorem 3. *For any $\lambda < 2/N$ the system is stable and the expected time a packet is delivered is $O(N)$.*

Proof. The route length is bounded by N.

Let r.v. X denote the delay.

Then

$$E[X] = \sum_{t \geq 1} \Pr(X \geq t)$$

$$= t_0 + \sum_{t > t_0} e^{-\epsilon t} = O(1)$$

$\square$

Theorem 4. *With probability* $1-1/N$ *a given packet is not delayed more than* $O(\log N)$ *steps.*

Theorem 5. *With probability* $1-1/N$ *a given queue at a given time step has no more than 3 elements.*

Dynamic Circuit Switching

2N-input/output $\log N$-dimensional butterfly network.

Each input has N queues for N possible destinations.

Each input generates in each step a new request with probability p_0. Request have random output destinations.

Algorithm

In each step each input tries to establish a path to its (random) destination.

In order to route a message from input i to output j, the unique greedy path from i to j is reserved.

If two paths try to use the same edge - one path is canceled (request returns to queue).

Lemma 3. *If input v tries to establish a path at a given iteration, the probability that the path is established is at least $1/\log N$.*

Proof.

Consider the case in which all inputs try to establish paths, each to a random destination.

Let p_i be the probability that an edge e at level i is used by a path. Then,

$$p_{i+1} = 2\frac{p_i}{2} - \frac{p_i^2}{4}$$

$$p_i \sim \frac{4}{i+4}$$

$$p_{\log N+1} \sim \frac{4}{\log N}$$

Thus the expected number of packets that will reach their destinations is at least $N/\log N$. Thus

$1/\log N$ of the paths are established. By symmetry a given request is established with probability at least $1/\log N$. $\square$

Theorem 6. *The system is stable if arrival rate to the inputs is less than* $1/\log N$*.*

Theorem 7. *The expected wait in the queue is* $O(N \log N)$*.*