Packet Routing

In Array networks

Structure and Sections

Greedy algorithms

Probabilistic analysis

Randomized algorithm

Deterministic algorithm

Off line routing

Greedy algorithms - Linear array

- Shortest path is used.
- Each packet that needs to move rightwards or leftwards does so.
- Algorithm terminates when all packets arrive at destination.
- each node can be a source and/or destination of **at most one packet** => no contention.
- Algorithm will need N-1 steps (worst case).

Greedy algorithm - 2D array

- Queuing discipline needed.

`Farthest_first()`: The packet that needs to move farthest does so.

- Basic Greedy Algorithm:
- Move on the column to the correct row.

Move on the row to the correct cell using Farthest_first.

Farthest first - 1D array

- Lemma: Consider a N node linear array in which each node contains an arbitrary number of packets, but for which there is at most one destined for each node. If edge contention is resolved via farthest first then greedy routes all the packets in N-1 steps.

Greedy algorithm - 2D array

- First phase needs $\sqrt{N} - 1$
- Second phase needs another $\sqrt{N} - 1$
- Nothing better can be asked for!

Queue size

- Source at

(1,2), (1,3),…,(1,$\sqrt{N}/3$) and

(2,1), (2,2),…,(2, 2$\sqrt{N}/3$)

- Destination

(3, $\sqrt{N}/3$), (4, $\sqrt{N}/3$),…,($\sqrt{N}$, $\sqrt{N}/3$)

- They arrive at processor at (2, $\sqrt{N}/3$) within $\sqrt{N}/3-1$ steps
- A queue of size 2 $\sqrt{N}/3 - 1$ emerges.

Greedy algorithm – Average Case

Two setups for the average case analysis.

- [Static] Each processor has a packet destined to a random processor.
- [Dynamic] At each processor a packet is generated at every step with a sufficiently small probability. Destination is random.

N packets to random destinations

- It is possible that several packets have the same destination. Worst case now could be N (though exceedingly small).
- BUT: Every packet gets to its destination within d+O(logN) steps with high probability 1-O(1/N).
- No more than 4 packets wait for the same edge with high probability.

N packets to random destinations

There is never contention for row edges.

The size of a column queue increases when more than one packets enter. Only one of them arrives over the column edge. We say the others turn. We want to estimate how many turn.

Each packet turns once along its path! At most $\sqrt{N}$ turn belong to that row and can turn. The probability that the destination is such that it turns is $1/\sqrt{N}$

Probability that r packets turn

$$\binom{\sqrt{N}}{r}\left(\frac{1}{\sqrt{N}}\right)^{r}$$

Lemma 1.6

$$\forall x, y : 0 < x < y \Rightarrow \binom{x}{y} < \left(\frac{xe}{y}\right)^{y}$$

Bound on turn probability

$$\binom{\sqrt{N}}{r}\left(\frac{1}{\sqrt{N}}\right)^r < \left(\frac{\sqrt{N}e}{r}\right)^r \left(\frac{1}{\sqrt{N}}\right)^r = \left(\frac{e}{r}\right)^r = O(1)$$

Bound on turn probability

$$r = \frac{e \log N}{\log \log N} \Rightarrow \text{queue bound is} \qquad o(N^{-2})$$

Bound queue size

- For all 4N processors with probability at most o(1/N) no queue exceeds $\dfrac{e \log N}{\log \log N}$

Packet streams

- Analysis based on steady packet streams. If they are not provided queues will be a lot smaller.
- Wide channel model assumes no queuing delays. It can be used to reason back on the standard model. When no to many packets cross a column edge in a window of time in the wide channel model, not too many cross it in the standard model.

Bound "wide channel"

- Consider the window [t+1, t+Δ]. If Q is the # packets crossing a queue from (i,j) to (i+1,j) in that window in the wide channel model then

$$Q \leq e^{(\alpha - 1 - \alpha \ln \alpha)\Delta / 2}$$

Bound "wide channel"

- Consider the window [t+1, t+Δ]. If a packet crosses the edge it must originate in the upper i rows. If it crosses the edge in time T it must be at distance T. At most 2 cells in a row are at that distance. Ergo 2i cells are eligible. This happens for all time points in the window.
- All eligible packets are 2iΔ.
- The probability that they are destined for for the remaining packets on the column j is $\sqrt{N} - i$

- Expected # packets during window:

$$\frac{2i(\sqrt{N} - i)\Delta}{\Delta} \le \frac{\Delta}{2}$$

Chernoff bounds

- Given a collection of independent Bernoulli rv's where $P[X_k] \leq P_k$

$$\Pr[X \geq \beta P] \leq e^{(1-\frac{1}{\beta}-\ln\beta)}$$

$$X = X_1 + X_2 + \ldots + X_n$$

$$P = P_1 + P_2 + \ldots + P_n$$

A Bound via Chernoff

- For $n = 2i\Delta, P_k = \frac{\sqrt{N} - i}{N}, P = \frac{2i(\sqrt{N} - i)\Delta}{N}, \beta = \frac{\alpha N}{4i(\sqrt{N} - i)}$

We get that the # packets crossing the edge Q

$$\Pr[Q \geq \alpha\Delta/2] \leq e^{(1-1/\beta - \ln\beta)\alpha P}$$

$$\leq e^{(1-1/\alpha - \ln\alpha)\alpha P}$$

Wide channel – standard model

Lemma 1.8: If a packet *p* is at distance *d* from a column edge *e* after step T and *p* crosses *e* after step T+d+δ, in the standard model then a packet must cross e at each step in the interval [T+d,T+d+ δ]

Corollary 1.9: If a packet crosses a column edge *e* at step T of the wide channel model and it crosses e at step T+ δ of the standard model, then some packet crosses *e* at every step in the interval [T,T+ δ] of the standard model

Wide channel – standard model

Lemma 1.10: For all T, Δ, x>0 if x packets cross an edge during a window of Δ steps [T+1, T+ Δ] of the standard routing model, then there is a $t \geq 0$ for which at least x+t packets cross e during the interval [T+1-t, T+ Δ] of the wide channel model.

Standard model

Lemma 1.11: The probability that $\alpha\Delta/2$ or more packets cross some column edge *e* during a window of Δ steps using the basic greedy algorithm is at most

$$O(e^{(\alpha-1-\alpha\ln\alpha)\Delta/2}) \Leftarrow 1 \le \alpha \le 2$$

Standard model

Lemma 1.12: With probability 1-O(1/N) no more than c logN packets cross nay column edge on consecutive steps of the basic greedy algorithm where $c < 9$.

Standard model

Theorem 1.13: When the basic greedy routing algorithm is used to route N packets on an $\sqrt{N} \times \sqrt{N}$ array the maximum number of packets ever queued on any edge at any time is at most four. With probability $1-O(\log^4 N/\sqrt{N})$.

- Moreover the probability that any particular packet is delayed steps is at most $O(e^{-\Delta/6})$

Dynamic routing

Model change → dynamisation.

Static routing with time barriers.

Stability issues rise!

Dynamic routing stability

Stability issues rise!

λ Probability of packet generation every step any cell.

Bisection width is $\sqrt{N}$.

$$\lambda \geq 4/\sqrt{N} \Rightarrow \frac{N}{2}\lambda\frac{1}{2} \geq \sqrt{N}$$

Ergo

$$\lambda < 4/\sqrt{N}$$

Dynamic routing stability

Theorem 1.14: If the arrival rate of packets is at most 99% of network capacity, then the probability that any particular packet is delayed Δ steps is at most $e^{-c\Delta}$ where c > 0 does not depend on N or the time the packet was generated.

Moreover in any window of T steps the maximum delay incurred by any packet is O(logT+logN) with high probability and the maximum observed queue size is

$$O\left(1+\frac{\log T}{\log N}\right)$$

Dynamic routing result

Theorem 1.14: If the arrival rate of packets is at most 99% of network capacity, then the probability that any particular packet is delayed Δ steps is at most $e^{-c\Delta}$ where c > 0 does not depend on N or the time the packet was generated.

Moreover in any window of T steps the maximum delay incurred by any packet is O(logT+logN) with high probability and the maximum observed queue size is

$$O(1+\frac{\log T}{\log N})$$

Randomized routing

Generally running time is not a problem. Queue sizes are!

Since average case is good make worst case average. This is achieved via randomization.

We will reduce queue sizes without increasing running time .

Idea: First route to a random destination then finish by routing to the final destination.

Randomized routing results

Algorithm design tools

1. Restrict choice of intermediate destinations.
2. Manage queues carefully

Best result known for one to one routing

Running time: $2\sqrt{N}+O(\log N)$

Queues: $O(1)$ whP

Result presented here

Running time: $2\sqrt{N}+o(\sqrt{N})$

Queues: $O(\log N)$ whP

Randomized routing algorithm

Phase 1 Partition each column into log N intervals. Route each packet into a random destination within its interval.

Phase 2 Each packet is routed to its correct row.

Phase 3 Each packet is routed to its correct destination within its column.

Analysis Randomized routing

Phase 1

Each partition has $\sqrt{N}/\log N$ elements. Since every node originally has one packet exactly this is the time needed.

How many packets end up in each node?

Via Chernov the probability for one cells queue Q

$$\Pr(Q > \frac{3 \ln N}{\ln \ln N}) \leq O(1/N^2)$$

Analysis Randomized routing

Phase 2

This phase needs $\sqrt{N} + o(\sqrt{N})$ steps. Contention is resolved using LIFO discipline.

Ergo once a packet starts moving it is never delayed.

We have to bound the time t a packet is delayed before moving. If the packets resides in processor (i,j) t packets are in (1,j), (2,j), …, (i,j) and pass to node (i,j+1) during 1,2,…,t of phase 2.

Chernoff II Lemma1.15

$$\alpha \le \sqrt{P} \Rightarrow$$

$$\Pr[X \ge P + \alpha\sqrt{P}] \le e^{-\alpha^2/3}$$

$$\alpha = o(P^{1/6}) \Rightarrow$$

$$\Pr[X \ge P + \alpha\sqrt{P}] = (1 + o(1))e^{-\alpha^2/2}$$

Randomized routing analysis cont.

- There are $j\sqrt{N}/\log N$ packets that could wind up in the first j nodes of row i after phase 1. Each packet does that with probability $\log N / \sqrt{N}$.
- Depending on whether $j \geq 6 \ln N$ we can use Chernoff I or II and obtain that whp $1 - O(1/N)$

$j + o(\sqrt{N})$ packets are located in (i,1)(i,2)…(i,j)!

Since the distance is at most $\sqrt{N}-j-1$ and queuing delay $j + o(\sqrt{N})$ =>

Phase 2 needs whp

$$\sqrt{N} + o(\sqrt{N})$$

Queues in randomized routing

- Queues in row edges in phase 2 do not increase. We must bound # packets at the beginning of the phase. Consider node (i,j) as a destination node for phase 2. It can be destination for at most $\sqrt{N}$ packets, which have a column j destination (because of 1-1). Each one of them has a log $N/\sqrt{N}$ change to go to row I in phase 1.
- O(log N) of the packets are sent to row i with probability $1-O(1/N^2)$
- The same happens for all N processors with probability 1-O(1/N)

Randomized routing Phase 3

- The analysis of this phase is identical as that of basic greedy routing. We begin with O(logN) packets in each processor. The # of packets do not increase during that phase.
- Every packet is in the correct column and at most one is destined for each processor. Running time is at most $\sqrt{N}-1$.

Randomized routing

- Running time is at most $2\sqrt{N} + o(\sqrt{N})$.
- Queues are $O(\log N)$