Dynamic Packet Routing on Arrays with Bounded Buffers

Andrei Z. Broder[1], Alan M. Frieze[2]*, and Eli Upfal[3]

[1] Digital Systems Research Center, 130 Lytton Avenue, Palo Alto, CA 94301, USA.
[2] Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh PA15213, USA
[3] IBM Almaden Research Center, San Jose, CA 95120, USA
and
Department of Applied Mathematics, The Weizmann Institute of Science Rehovot, Israel.
broder@pa.dec.com, af1p@andrew.cmu.edu, eli@wisdom.weizmann.ac.il

Abstract. We study the performance of packet routing on arrays (or meshes) with bounded buffers in the routing switches, assuming that new packets are continuously inserted at all the nodes. We give the first routing algorithm on this topology that is stable under an injection rate within a constant factor of the hardware bandwidth. Unlike previous results, our algorithm does not require the global synchronization of the insertion times or the retraction and reinsertion of excessively delayed messages and our analysis holds for a broad range of packet generation stochastic distributions. This result represents a new application of a general technique for the design and analysis of dynamic algorithms that we first presented in [Broder et al., FOCS 96, pp. 390–399.].

1 Introduction

The rigorous analysis of the dynamic performance of routing algorithms is one of the most challenging current goals in the study of communication networks. So far, most theoretical work on this area has focused on static routing: A set of packets is injected into the system at time 0, and the routing algorithm is measured by the time it takes to deliver all these packets to their destinations, assuming that no new packets are injected into the system in the meantime (see Leighton [8] for an extensive survey). In practice however, networks are rarely used in this "batch" mode. Most real-life networks operate in a *dynamic* mode whereby new packets are continuously injected into the system. Each processor usually controls only the rate at which it injects its own packets and has only a limited knowledge of the global state.

This situation is better modeled by a stochastic paradigm whereby packets are continuously injected into the system according to some inter-arrival distribution, and the routing algorithm is evaluated according to its long term behavior.

* Research supported in part by NSF Grant CCRR-9530974

In particular, quantities of interest are the maximum arrival rate for which the system is stable (that is, the arrival rate that ensures that the expected number of packets waiting in queues does not grow with time), and the expected time a packet spends in the system in the steady state. The performance of a dynamic algorithm is a function of the inter-arrival distribution. The goal is to develop algorithms that perform close to optimal for any inter-arrival distribution.

Several recent articles have addressed the dynamic routing problem, in the context of packet routing on arrays [7, 6, 9], on the hypercube and the butterfly [12] and general networks [11]. The analysis in all these works assumes a Poisson injection rate and requires unbounded queues in the routing switches (though some works give a high probability bound on the size of the queue used [7, 6]). Unbounded queues allow the application of some tools from queuing theory (see [4, 5]) and help reduce the correlation between events in the system, thus simplifying the analysis at the cost of a less realistic model. Clearly bounded buffers in the routing switches is a setting that most accurately models real networks.

A general technique for the design and analysis of dynamic packet routing algorithms has been developed in [1]. The crux of that work is a general theorem showing that any communication scheme (a routing algorithm and a network) that satisfies a given set of conditions, defined only with respect to a *finite history*, is stable up to a certain inter-arrival rate. Thus, the analysis of the long term behavior of a dynamic algorithm is reduced to a simpler question of analyzing a finite execution of the algorithm. Furthermore, this technique also gives a bound on the expected routing time in the stable state. The theorem applies to any inter-arrival distribution: the stability results and the expected routing time of a packet inside the network depend only on the inter-arrival rate. The waiting time in queues depends on the inter-arrival distribution and an explicit relation is given in the main theorem of [1].

To apply the general technique one needs to present an algorithm whose performance analysis (on finite segments) satisfies the conditions of the theorem. Several applications of the general technique to routing algorithms for low diameter networks such as the butterfly have been demonstrated in [1]. Here we present the first application to arrays: we consider an $n \times n$ mesh of routing switches with bounded buffers. Each routing switch node is connected to a processor. A processor has its own queue for the packets generated by it that are waiting to enter the network. We can assume this processor queue to be unbounded. (De facto, this queue is finite and when it becomes full the processor stops generating new packets.) Other packets pass only through the routing switch. The routing switch has a *routing queue* stored in a bounded buffer where packets waiting to be routed are placed. For simplicity we assume that packets have random destinations. This assumption can be relaxed as long as no destination is overloaded with packets.

Theorem 1. *There is a packet routing algorithm for the $n \times n$ mesh with bounded buffers that is stable for any inter-arrival distribution with expectation at least Cn for some fixed constant C. The expected time a packet spends in the network*

is $O(n)$. In the case of Poisson arrival (geometric inter-arrival distribution) the expected time the packet spends in the queue is also $O(n)$.

Since the distance between most pairs of nodes on the mesh is $\Omega(n)$ the above theorem is clearly optimal up to constant factors.

Leighton [7] has studied this problem and obtained similar results provided that the buffers in the routing switches are *unbounded.* More precisely, Leighton's algorithm ensures that at any *fixed* time, with high probability, no routing queue has more than 4 packets. However, for any sufficiently long execution the maximum size of any queue exceeds any given bound. Our results build on Leighton's analysis, by augmenting his algorithm with a simple flow control mechanism, which ensures that every routing queue is bounded at all times, and thus only finite buffers are needed.

In a different direction, several recent works studied the dynamic performance of deflection (hot potato) routing. In deflection routing packets in the networks always move and there are no buffers in the routing switches. However, the analysis in these works requires either strong synchronization between processors in timing injections of new packets to the network [3, 10], or a mechanism to retract and reinsert excessively delayed packets [2].

2 The General Technique

We outline here the general technique developed in [1]. The setting is as follows: we are given a routing algorithm $\mathcal{A}$ acting on a **synchronous** network $\Gamma(n)$ with n inputs and n outputs. Each input receives new packets with inter-arrival distribution $\mathcal{F}$. The packets are placed into an unbounded FIFO queue at the input node. Packets have an output destination chosen independently and uniformly at random. When a packet reaches the top of its queue, we call it *active.* At some point after becoming active, the packet is removed from its queue and eventually routed to its destination.

We are interested in determining under which conditions the queuing system is *ergodic* (or stable), that is, under which conditions the expected length of the input queues is bounded as $t \to \infty$. To this purpose we have to study the *inter-departure* time, which is the interval from when a packet becomes active until it leaves the queue, and the next packet in line (if any) becomes active. Besides stability, we are also interested in the expected time a packet spends in the queue, and the expected time it spends in the network.

Since the inter-arrival times are independent, if the inter-departure times are also independent, then each queue can simply be viewed as a G/G/1 system and the stability condition would trivially be that the inter-departure rate exceeds the inter-arrival rate. However the usual situation is that there are complex interactions among packets during routing and thus the inter-departure times are highly dependent and hard to analyze.

The following theorem defines a set of relatively simple sufficient conditions such that if the routing algorithm satisfies them, then the system is stable up to

a certain inter-arrival rate and we can bound the expected time a packet spends in the queue and in the network.

Theorem 2. *Assume that the randomized routing algorithm $\mathcal{A}$ acting on the network $\Gamma(n)$ is characterized by four parameters a, b, m, and T, where a and b are constants, m and T might depend on n, and $m/T < 1$, and that it satisfies the following conditions:*

(1) *Every packet is delivered at most $O(n^a)$ steps after becoming active.*
(2) *There exists an event $\mathcal{E}_\tau$ (where τ is a time) with the following properties:*
   (a) *$\mathcal{E}_\tau$ depends only on random choices made in the open interval $(\tau - n^b, \tau + n^b)$. (These choices are: random destinations chosen by the packets that became active in the interval, random arrivals in this interval, and random coin flips made by the algorithm.)*
   (b) *$\neg\mathcal{E}_\tau$ implies that any packet that at time τ was among the first m packets in its queue, is delivered before time $\tau + T$.*
   (c) *For any fixed time τ the probability $\Pr(\mathcal{E}_\tau)$ is bounded by*

$$\Pr(\mathcal{E}_\tau) \leq \frac{(m/T)^7}{n^{2a+2b+3}} .$$

If there exists a positive constant ϵ such that the inter-arrival distribution $\mathcal{F}$ has an inter-arrival rate smaller than $(1-\epsilon)m/T$, then the system is stable and the time a packet spends in the input queue is bounded by $O(T) + f(\frac{T}{m})$, where f is a function that depends only on $\mathcal{F}$ and not on the routing process. (For "reasonable" distributions such as Poisson $f(\frac{T}{m}) = O(T/m)$). Furthermore the average time elapsed since a packet becomes active until it is delivered is also $O(T)$. □

3 The Main Result

3.1 The Algorithm

The description of the algorithm has three components: path selection; routing switch policy; and flow control.

Path selection. A packet takes the shortest one-bend route from origin to destination. First (left or right) on its origin row to its destination column, then up or down on that column to the packet's destination.

Switch policy. We assume that a switch can receive up to four packets per step, one from each incoming edge, and send four packets per step, one through each outgoing edge. A switch maintains a buffer for each outgoing edge. When there is a space in a buffer the switch receives packets to that buffer according to the following rule: A packet is *old* if it has spent at least Kn steps in the network, where K is a sufficiently large constant ($K \geq 2e^2$) will suffice. Old packets have higher priority. Among the old packets the oldest packet has the highest priority. Old packets of the same age are dealt with in lexicographic order of pair (origin,destination). Among packets that are not old, the packet that has to travel farthest has the highest priority.

Admission control. The algorithm uses a token based admission control mechanism. Each input has one token. A token can be in one of three modes: *enabled*, *used* or *suspended*. Initially all the tokens are in enabled mode. To inject a packet into the network the input needs an enabled token. The packet at the top of the processor queue is sent together with the enabled token to its destination, provided there is room in the edge buffer which will receive it. When a packet is delivered the mode of its token switches to used mode and the token (acknowledgment) is returned to the input node where it came from. We use a separate network to route tokens back to their sources and the analysis of this routing mirrors that of the main network. Details are left to the full paper. Let t_s be the last time a given token was sent with a packet, and let t_r be the last time it returns to its input node.

If $t_r - t_s \leq 2Kn$ then the token becomes enabled again at time $t_r + 2Kn + Z$, where Z is a random number chosen uniformly from $[0, Kn]$. If $t_r - t_s > 2Kn$ then the token mode is switched to suspended mode until time $t_r + 6n^7 + Z$ steps, then it is switched again to enabled mode, where Z is chosen as above.

This flow mechanism guarantees that an input cannot inject more than one packet within each interval of $2Kn$ steps, and that the input does not inject new packets when the network is congested. Furthermore, observe that the probability that a token becomes enabled at any fixed time is at most $1/(Kn)$.

3.2 Analysis of the Algorithm

In this section we give an analysis of the performance of the above algorithm for finite intervals showing that the algorithm satisfies the requirements of the general technique (Thm. 2). This will prove Thm. 1.

We will assume Poisson arrivals i.e. at each step, a packet arrives with probability $1/(Cn)$. The general case can be handled as in [1]. The following lemma satisfies requirement (1) of the general theorem:

Lemma 3. *Under this protocol, no packet takes more than $S = 7n^7$ steps to reach its destination, once it has become active and at most $2n^5$ steps once it has obtained an enabled token.*

Proof. Consider a packet P. Let P_0 be its predecessor in the queue. P_0 does not leave the queue until it has an enabled token. At that time there are no more than n^2 other packets in the network. Consider the progress of the highest priority old packet Π in the network. If Π is moving along a column then it moves at every time step. If it is moving along a row, then it could fail to move because further along that row there is contention for a column buffer. Π waits at most $Kn + n^2$ steps before making another move. This is because the packet waiting to move along the column in question will have become old and it will be the oldest packet trying to get into the column edge buffer. Thus after $Kn + Kn^2 + n^3$ steps Π will have reached its destination column and will reach its final destination within a further n steps. So after at most $Kn^3 + Kn^4 + n^5$ steps, P_0 will be the highest priority packet in the network and will be delivered

within a further $Kn + Kn^2 + n^3$ steps. Thus P_0 gets to its destination at most $Kn+Kn^2+(K+1)n^3+Kn^4+n^5 \leq 2n^5$ steps after leaving the queue. The used token comes back after at most another $2n^5$ steps and after at most $6n^7 + Kn$ steps is re-activated. Finally, after at most another $2n^5$ steps the packet P is delivered. The sum of these delays is less than $7n^7$. □

Next we turn to the definition of the event $\mathcal{E}_\tau$, and to the probabilistic analysis of the algorithm's performance in finite intervals. Our analysis is based on the the technique in [7] as described in [8] [Sect. 1.7.2]. As in [7] we relate the execution of the algorithm to an artificial execution on a *wide-channel* model in which an arbitrary number of packets can traverse an edge at any step, and no packet is ever delayed. We assume that execution on the wide-channel starts at time τ.

Let c and q be constants to be defined later, let

$$d_0 = (c+3)\log n + \log(2K) \ .$$

This serves as a suitable high probability upper bound on the delay of any given packet. Define the events $\mathcal{E}_\tau$ as follows:

$\mathcal{E}_\tau$ is the event that at least one of the following occurred:

(1) There is an old packet in the system at any time in the interval $[\tau, \tau + Kn]$.
(2) There is a row edge e, a $t \geq 0$, and an interval $[t_0, t_0+t+d_0] \subseteq [\tau, \tau + Kn]$, such that $t + d_0$ packets traverse edge e in that interval in the wide-channel model.
(3) There is a column edge e, a $t \geq 0$, and an interval $[t_0, t_0 + t + 2d_0] \subseteq [\tau, \tau + Kn]$, such that $t + d_0$ packets traverse edge e in that interval in the wide-channel model.
(4) A routing buffer has q packets in some step in the interval $[\tau, \tau + Kn]$.

We say that a packet was delayed d steps in traversing an edge if there is a d steps gap between the time it traverses the edge in the wide-channel model and the time it traverses the edge in the standard model. Leighton's analysis in [7] is based on the following fact (see Cor. 1.9 and Lemma 1.10 in [8]): If buffers are unbounded and the farthest to go packet always has the highest priority, then a packet is delayed d steps in traversing a row edge e only if there is an interval of $t + d$ steps such that $t + d$ packets crossed edge e in that interval in the wide-channel model. Similarly, if a packet is delayed d steps in crossing a column edge e then, assuming no packet is delayed more than d_0 steps on a row, there is an interval of $t + d_0 + d$ steps in which $t + d$ packets cross edge e in the wide-channel model (see [8] for a detailed proof). Thus we have the following corollary that satisfies requirement (2)(b) in the general theorem:

Corollary 4. *The event $\neg\mathcal{E}_\tau$ implies that any packet with an enabled token at time τ is delivered within the next $2n + 2d_0 \leq Kn$ steps.*

Proof. Any packet with an enabled token is delivered within $2n$ steps in the wide-channel model. In the standard model its additional delay is at most $2d_0$. □

Next we bound the probability of the event $\mathcal{E}_\tau$. Assume first that old packets are removed from the system the moment they become old, and all buffers are unbounded.

For an edge e and an interval $[t_1, t_2] \subseteq [\tau, \tau + Kn]$ let $\mathcal{E}_0(e, t_1, t_2, r)$ be the event that in the wide-channel model r packets cross e during time interval $[t_1, t_2]$. Note that every token is used at most once in the interval.

Case 1. e is a row edge:

$$\Pr(\mathcal{E}_0(e, t_1, t_2, r)) \leq \binom{n}{r} \left(\frac{t_2 - t_1}{Kn}\right)^r \leq \left(\frac{e(t_2 - t_1)}{rK}\right)^r .$$

(The nodes on the row under consideration have a total of n tokens. Each token is used at most once in the interval. Choose r of them to transport the packets of interest. The probability that a token becomes enabled at any fixed time is at most $1/(Kn)$.)

Case 2. e is a column edge:

$$\Pr(\mathcal{E}_0(e, t_1, t_2, r)) \leq \binom{n^2}{r} \left(\frac{t_2 - t_1}{Kn^2}\right)^r \leq \left(\frac{e(t_2 - t_1)}{rK}\right)^r .$$

(There is a total of n^2 tokens. Each token is used at most once in the interval. Choose r of them to transport the packets of interest. The probability that a token becomes enabled at any fixed time is at most $1/(Kn)$ and the probability that the uses a particular column is $1/n$.)

Thus, under the assumption that no old packets are present in the interval $[\tau, \tau + Kn]$ the probability that there is a row edge e, a $t \geq 0$, and an interval $[t_0, t_0 + t + d_0] \subseteq [\tau, \tau + Kn]$, such that $t + d_0$ packets traverse edge e in that interval in the wide-channel model is bounded by

$$Kn^3 \sum_{t \geq 0} \left(\frac{e(d_0 + t)}{(d_0 + t)K}\right)^{d_0 + t} \leq 2Kn^3 \left(\frac{e}{K}\right)^{d_0} \leq n^{-c}$$

for $K \geq e^2$. (There are Kn possible values for t_0 and n^2 edges.)

Similarly, under the same assumptions, the probability that there is a column edge e, a $t \geq 0$, and an interval $[t_0, t_0 + t + 2d_0] \subseteq [\tau, \tau + Kn]$, such that $t + d_0$ packets traverse edge e in that interval in the wide-channel model is bounded by

$$Kn^3 \sum_{t \geq 0} \left(\frac{e(2d_0 + t)}{(d_0 + t)K}\right)^{d_0 + t} \leq Kn^3 \sum_{t \geq 0} \left(\frac{2e}{K}\right)^{d_0 + t} \leq 2Kn^3 \left(\frac{2e}{K}\right)^{d_0} \leq n^{-c} ,$$

provided that $K \geq 2e^2$.

Remark 5. It follows (see [8, Sect. 1.7.2, Lemma 1.10]) that with probability at least $1 - 2n^{-c}$, for every edge e and time interval I of d_0 steps, there is a step in I in which e is empty.

Now we show that the assumptions that no old packets are present in the interval $[\tau, \tau + Kn]$ holds with high probability.

Consider an interval $[\tau', \tau' + 2Kn]$. By definition, the only packets that can become old in the interval $[\tau' + Kn, \tau' + 2Kn]$ had enabled tokens at time τ'. Thus, in view of the discussion above, the probability that any packet becomes old in $[\tau' + Kn, \tau' + 2Kn]$ given that there were no old packets present in the interval $[\tau', \tau' + Kn]$ is at most $2n^{-c}$.

Hence if the interval $[\tau - \sigma, \tau + Kn]$ contains any subinterval of length at least Kn when no old messages are present, we can generously bound the probability that any old messages are present in the interval $[\tau, \tau + Kn]$, by $2(\sigma + Kn)n^{-c}$.

Take $\sigma = n^2 S + kN$. Then if a packet from a particular source becomes old at some time in $[\tau - \sigma, \tau + Kn]$, it is delivered within S steps, and its token is not enabled again until after time $\tau + Kn$. Hence there is an interval of length at least $\sigma - n^2 S = kN$ when certainly no old messages are present, and thus with high probability no old messages are present in $[\tau, \tau + Kn]$.

Next we bound the probability that any buffer is full in the interval $[\tau - \sigma, \tau + Kn]$. From Remark 5 we can assume that some row edge has q packets in its buffer only if there is a window of d_0 steps in which some input tries to inject at least q packets. The probability of this is at most

$$(\sigma + Kn)n^2 \binom{d_0}{q} \left(\frac{1}{Cn}\right)^2 = O(n^{-c})$$

for sufficiently large q.

From Remark 5 we can further assume that a column edge has q packets in its buffer only if there is a window of d_0 steps in which q packets turn at e, Assuming no row delay of d_0 or more, the probability of this is bounded by

$$(\sigma + Kn)n^2 \binom{n}{q} \left(\frac{2d_0}{Kn^2}\right)^q = O(n^{-c})$$

for sufficiently large q (see [8, Sect. 1.7.2, Thm. 1.13]). (The factor $(\sigma + Kn)n^2$ bounds the number of (interval $I = [t_1, t_2]$, edge e) pairs. There are $\binom{n}{q}$ choices of token. There is a probability $1/n$ that its destination uses e. There is a probability of at most $2d_0/(Kn)$ that it becomes enabled at a time which means it would cross e during $[t_1 - d_0, t_2]$ in the wide-channel model.)

Now choose a, b, and c such that

$$\begin{aligned} n^a &\geq S = 2((K+2)n + n^2) \ , \\ n^b &\geq \sigma = n^2 S + Kn \ , \\ c &\geq 2a + 2b + 11 \ . \end{aligned}$$

Then summarizing the discussion above

$$\Pr(\mathcal{E}_\tau) \leq n^{-(2a+2b+3)} .$$

Finally we observe that in evaluating the probability of the event $\mathcal{E}_\tau$ we only used events in the interval $[\tau - \sigma, \tau + Kn]$ and thus requirement (2)(a) in the general theorem is satisfied for n^b. Thus we showed that all the condition of Thm. 2 are satisfied for $T = 2Kn$ and $m = 1$. □

4 Conclusion

We have shown how to combine our general technique, first presented in [1], with that of Leighton's analysis [8] of a greedy routing algorithm on an $n \times n$ mesh. The results are optimal up to constant factors.

References

1. A. Z. Broder, A. M. Frieze, and E. Upfal. A general approach to dynamic packet routing with bounded buffers. *Proceedings of the 37th IEEE Symp. on Foundations of Computer Science.* Burlington, 1996, pp. 390–399.
2. A. Z. Broder and E. Upfal. Dynamic deflection routing on arrays. *Proceedings of the 28th ACM Symp. on Theory of Computing.* Philadelphia, 1996, pp. 348–355.
3. U. Feige. "Nonmonotonic phenomena in packet routing", manuscript, march 1997.
4. M. Harcol-Balter and P. Black. Queuing analysis of oblivious packet routing networks. *Procs. of the 5th Annual ACM-SIAM Symp. on Discrete Algorithms.* Pages 583–592, 1994.
5. M. Harcol-Balter and D. Wolf. Bounding delays in packet-routing networks. *Procs. of the 27th Annual ACM Symp. on Theory of Computing,* 1995, pp. 248–257.
6. N. Kahale and T. Leighton. Greedy dynamic routing on arrays. *Procs. of the 6th Annual ACM-SIAM Symp. on Discrete Algorithms.* Pages 558–566, 1995.
7. F. T. Leighton. Average case analysis of greedy routing algorithms on arrays. *Procs. of the Second Annual ACM Symp. on Parallel Algorithms and Architectures.* Pages 2–10, 1990.
8. F. T. Leighton. *Introduction to Parallel Algorithms and Architectures.* Morgan-Kaufmann, San Mateo, CA 1992.
9. M. Mitzenmacher. Bounds on the greedy algorithms for array networks. *Procs. of the 6th Annual ACM Symp. on Parallel Algorithms and Architectures.* Pages 346–353, 1994.
10. T. Rubshtein. A Dynamic Hot Potato Routing Algorithm for the Torus. M.Sc. thesis, The Weizmann Institute, 1996.
11. C. Scheideler and B. Voecking Universal continuous routing strategies. SPAA '96.
12. G. D. Stamoulis and J. N. Tsitsiklis. The efficiency of greedy routing in hypercubes and butterflies. *Procs. of the 6th Annual ACM Symp. on Parallel Algorithms and Architectures.* Pages 346–353, 1994.