Recurrence equation as basis for designing hot-potato routing protocols

Ville Leppänen and Martti Penttonen

Abstract: *In this work we demonstrate the use of generating functions for solving recurrence equations, which can occur e.g. in the analysis of routing protocols. We prove an exact bound for the expectation of packets reaching their target in a torus, when the routing protocol makes the packets to favor with probability p in each node a diagonal route to their target. Higher probability increases expectation of successful packets. We consider designing routing protocols according to the scenario and compare them to known results.*

Key words: *Computer Systems and Technologies, hot-potato routing, torus network.*

INTRODUCTION

Often we can analyze the probability (or complexity) of a complex phenomenon recursively: Assuming we can reduce any instance of the problem to a "smaller" instance of the problem, and we can evaluate the probability of the reduction, and we can also evaluate the probability of the "small" problem instances, then we can evaluate the probability of any problem instance. However, even though such a group of recurrence equations can be used for the computation of probabilities, solving these equations and providing a direct formula for the probability may prove out to be a very difficult task.

In this work we show, how generating functions can be used to solve a recurrence equation. We consider designing hot-potato routing algorithms that are could be characterized by the following recurrence equation

$$P(k,l)=\begin{cases} P(l,k), \forall k,l \\ \frac{1}{2}P(k-1,l)+\frac{1}{2}P(k,l-1), k=l>0 \\ pP(k-1,l)+(1-p)P(k,l-1), k>l>0 \\ pP(k-1,l), k>l=0 \\ 1, k=l=0 \end{cases} \qquad (1)$$

Hot-potato routing on torus

A routing strategy used to resolve output port contention problem in packet-switched interconnection networks is the *hot-potato* or *deflection* routing strategy. In the hot-potato routing all entering packets must leave at the next step - i.e. packets cannot be buffered as in the store-and-forward routing strategy. In general, in each node the out-degree must be at least the in-degree, the output port contention must be resolved somehow. If there are multiple packets preferring the same output port, the routing strategy must select at most one for each out-going link.

An out-going link is *good for a packet*, if it takes the packet closer to its destination. Other links are *bad* for that packet. Notice that several links might be good for a packet. A packet is said to *deflect* at a node, if the routing strategy assigns it to a bad out-going link. See e.g. [10] for definitions and a survey of hot-potato routing techniques and results.

Consider a two-dimensional *n x n* one-way torus as shown in Figure 1. Let the lower left corner have position (0,0). By *one-way property* we mean that links are directed as in Figure 1. When routing a packet e.g. from (k,l) to (0,0), it means that there are several shortest paths of length k+l in the *k x l* rectangular area between the nodes, but it is also possible for a packet to drop out of the rectangular area - each drop out increases the route length by *n*. The only bad links are those through which a packet drops out.

To avoid drop outs, it would seem wise to try to stay as far away from the edges as possible - i.e. to try to get close to the diagonal (see Figure 1) and then follow a diagonal route to the target. The Equation 1 characterizes the success probability of routing in a one-way two-dimensional grid, when packets tend to choose, with bias *p*, a diagonal route from node (k,l) to (0,0). Note that due to one-way property the success probability is less than 1, because the packets may drop out.

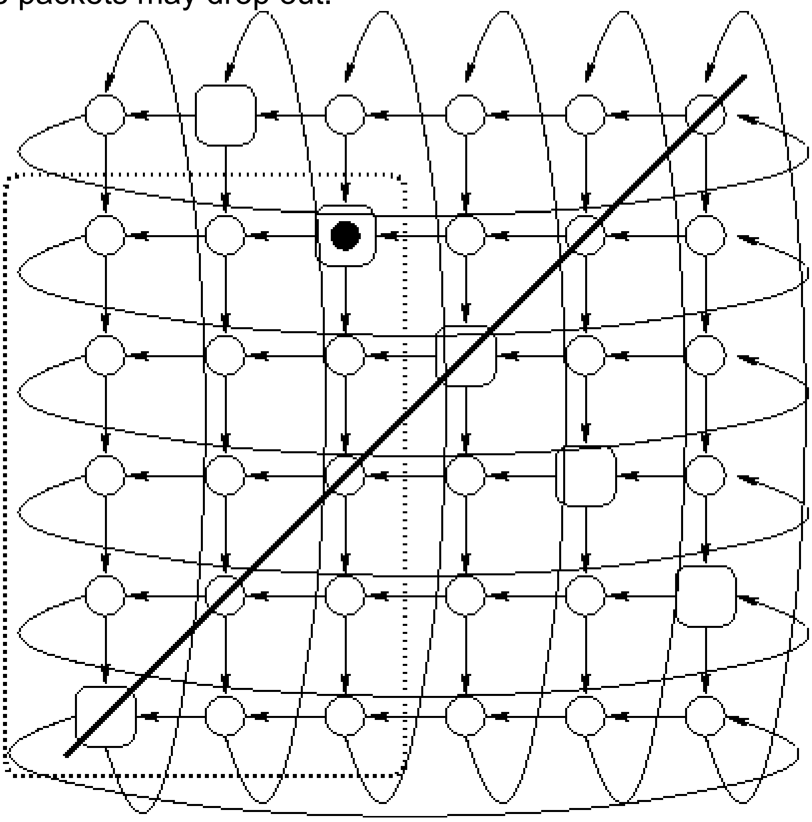

Figure 1. A 6 x 6 torus. Lower left corner is (0,0). Dotted area represents the rectangular area for packet targeted to (0,0) from node (2,4). Diagonal with respect to target node (0,0) is also shown.

Observe that from the view point of each target node (lower left corner in Figure 1) the approaching packets form an approaching front. In one-way torus, only the packets in a front at distance *d* can affect the route of a packet at distance *d*.

Results

Constructing routing algorithms for meshes and tori is studied in several papers. The problems studied as well as assumptions concerning the routing machinery differ quite a lot, see [4]. In [3,5,7,11], routing on various sparse meshes and tori are studied, but assumptions differ so much that fair comparison is not possible. Many of the results are asymptotic, although recently also analysis of the exact cost is done experimentally as well as analytically [1,2,5].

In this paper, we present a novel analysis of a recurrence equation, which can be used to characterize certain routing algorithms. Our analysis gives the exact expected routing cost. In Section 2 we present the analysis, and then in Section 3 we consider constructing routing algorithms that could be characterized with Equation 1.

ANALYSIS OF RECURRENCE EQUATION

Let *p* be a constant such that $0 \le p \le 1$ and let *P(k,l)* be a function defined by the recurrence Equation 1. Can we estimate the values of *P(k,l)*? When *p*=3/4, denote

$Q_k = P(k,k)$, and calculate $Q_0 = 1$, $Q_1 = 3/4$, $Q_2 = 45/64$, $Q_3 = 702/1024$, …, Q_k appears to approach 2/3. Indeed, this is the case. We can prove generally

Theorem 1.

(a) $\lim_{k\to\infty} P(k,k) = \dfrac{1-2(1-p)+\sqrt{1-4p(1-p)}}{2p}$

(b) $\lim_{n\to\infty} \dfrac{1}{n}\sum_{i+j=n} P(i,j) = \dfrac{1-p}{p}(1-2(1-p)+\sqrt{1-4p(1-p)})$

To prove (a), denote *q=1-p* and let $Q_k = P(k,k)$. When computing Q_k recursively, computation starts with a ``diagonal'' number $Q_k$ and ends at the ``diagonal'' number Q_0, and there may be other ``diagonal'' numbers Q_j, 0<j<k, in intermediate phases of the computation.

Assume we already know Q_j, for all j<k. We reduce the computation of Q_k to these values as follows. Consider paths from node (k,k) to node (0,0) in integer grid $\{(i,j)\,|\,0 \le i, j \le k\}$. Now

$$Q_k = 2\sum_{j=0}^{k-1} C_j \frac{1}{2} q^j p^{j+1} Q_{k-j-1.}$$

In this formula, 2 comes from the fact that on a path form (k,k) to (0,0) we return first to the diagonal point (k-j-1,k-j-1) either from the upper side of the diagonal or from the lower side of the diagonal. Each term in the sum corresponds to the first point on the diagonal (k,k), (k-1,k-1), …,(1,1), (0,0), where we first come after leaving (k,k). C_j is the number of paths from (k,k) to (k-j-1, k-j-1) that do not contain any diagonal points (k-1, k-1), …, (k-j,k-j). It is easy to see that among the edges of each path leading from (k,k) to (k-j-1, k-j-1), there is one that leaves (k,k) with probability 1/2, there are *j* edges that diverge from the diagonal with probability *q*, and there are *j+1* edges that converge towards the diagonal with probability *p*. Finally, the probability of reaching (0,0) from (k-j-1,k-j-1) is Q_{k-j-1}. The numbers C_j are known in literature as *Catalan numbers*, and we know

Lemma 1. [6] $C(z) = \sum_{i=0}^{\infty} C_i z^i = \dfrac{1-\sqrt{1-4z}}{2z}$

Analogously, denote $Q(z) = Q_0 + Q_1 z + Q_2 z^2 + \dots$. By grouping factors in Q_k, we have

$$Q_k = p\sum_{j=0}^{k-1} C_j (pq)^j Q_{k-j-1} = p[C(pqz)Q(z)]_{k-1},$$

where $[F(z)G(z)]_k$ is the multiplier of z^k in the product of polynomials *F(z)* and *G(z)*. Hence, we have $Q_0 = 1$ and $[Q(z)]_k z^k = p[C(pqz)Q(z)]_{k-1} z^k$, for *k>0*. By adding columnwise we get $Q(z) = 1 + pC(pqz)Q(z)z = p\dfrac{1-\sqrt{1-4pqz}}{2pqz}Q(z)z = 1 + \dfrac{1-\sqrt{1-4pqz}}{2q}Q(z).$

From this we can solve $Q(z) = \dfrac{2q}{2q-1+\sqrt{1-4pqz}} = \dfrac{1-2q+\sqrt{1-4pqz}}{2p(1-z)} = \dfrac{1}{1-z}R(z),$ where

$$R(z) = \frac{1-2q+\sqrt{1-4pqz}}{2p}.$$

Observe now that $1/(1-z) = 1 + z + z^2 + \dots$ and denote $R(z) = R_0 + R_1 z + R_2 z^2 + \dots$. Then

$$Q_k = \left[(1+z+z^2+...)(R_0+R_1z+R_2z^2+...)\right]_k = R_0+R_1+R_2+....$$

Hence, $\lim_{k\to\infty} Q_k = R_0+R_1+R_2+... = \dfrac{1-2q+\sqrt{1-4pqz}}{2p}$.

To prove part (b) of the theorem, denote $S_n = \sum_{i+j=n} P(i,j)$. By recursion rule,

$$S_n = P(o,n)+P(1,n-1)+...+P(\frac{n}{2}-1,\frac{n}{2}+1)+P(\frac{n}{2},\frac{n}{2})+P(\frac{n}{2}+1,\frac{n}{2}-1)+...+P(n-1,1)+P(n,0)$$

$$= pP(0,n-1)+(1-p)P(0,n-1)+pP(1,n-2)+...+(1-p)P(\frac{n}{2}-2,\frac{n}{2}+1)+pP(\frac{n}{2}-1,\frac{n}{2})$$

$$+\frac{1}{2}P(\frac{n}{2}-1,\frac{n}{2})+\frac{1}{2}P(\frac{n}{2},\frac{n}{2}-1)+pP(\frac{n}{2},\frac{n}{2}-1)+(1-p)P(\frac{n}{2}+1,\frac{n}{2}-2)+...$$

$$+pP(n-2,1)+(1-p)P(n-1,0)+pP(n-1,0)$$

$$= S_{n-1}+(1-p)P(\frac{n}{2}-1,\frac{n}{2})+(1-p)P(\frac{n}{2},\frac{n}{2}-1) = S_{n-1}+2(1-p)Q_{n/2}$$

for even *n*. The same argument applies for odd *n*. Hence,

$$S_n = 2(1-p)Q_0+...+2(1-p)Q_{n/2}$$

By $\lim_{n\to\infty} Q_n = \dfrac{1-2q+\sqrt{1-4pq}}{2p}$ we get $\lim_{n\to\infty}\dfrac{1}{n}S_n = \dfrac{1-p}{p}(1-2q+\sqrt{1-4pq})$. Q.E.D.

SYNTHESIS OF ROUTING ALGORITHMS

Consider that *P(k,l)* denotes the expectation for a packet to reach its destination that is *k* vertical and *l* horizontal steps away from the current position in a 2-dimensional one-way torus. In that case, Theorem 1b gives an expectation for a (randomly chosen) packet to reach its destination that is *n* steps away.

As a special case of Equation 1, consider

If we could construct a hot-potato routing protocol that is always able to forward with probability 3/4 a packet toward the diagonal of its destination node, then by Theorem 1b the expectation for an arbitrary packet to reach its destination is 1/3.

Notice also that if *p*=1/2 in Equation 1, then by Theorem 1b the expectation for a packet to reach its target approaches 0. I.e. arbitrarily forwarding the packets is not useful.

Next, assume that the torus is sparse - i.e. only the nodes marked with square are nodes (processors) that can produce and consume packets. Routing on such sparse meshes and tori are studied in [3,5,7,11].

Randomized diagonal routing

Consider first a simple *diagonal protocol P1* in one-way sparse torus, where packets always aim toward the diagonal leading to its target and priority is assigned randomly to the packets. With probability 1/2 a packet can choose the link in the node.

In "fresh" average case routing situation, when target addresses are almost random, even in the other case, when the other packet chooses first, our packet has equal chances to become forwarded toward the diagonal. Thus, initially the setting works with respect to Equation 1. When the routing proceeds, the packets that interact with each other are less and less random - their targets are closer to each other and the risk that forced move leads farther away from the diagonal increases. Thus, applying *p*=3/4 for Equation 1 gives upper bound for the efficiency of *P1*. Our experiments [5] confirm this.

How the protocol *P1* could be improved? Basically we need to find ways to restrict other packets ability to push packets over the edge of rectangular area. There are two ways to do this: (1) Improve to protocol so that for each packet there is a smaller set of packets that can drop it out the rectangular area, or (2) make the set of packets moving in the routing machinery more homogeneous. We present a protocol for both cases.

Improved diagonal routing

Next consider a protocol P2, where packets again always aim toward the diagonal, and higher priority is given to the packet that is (1) closer to its target and (2) farther away from the diagonal leading to its target. In a tie situation, priority is assigned randomly.

When modeling P2 with Equation 1, the initial situation appears to be the same as in protocol P1. However, rest of the routing process cannot be modeled with Equation 1, since the probability of staying within the rectangular area is now much more predictable. A crucial observation is that a packet on the edge of rectangular area gets pushed over the edge if only if both packets are aiming at the same target (and a packet once pushed over the edge is not able to push more packets over the edge).

This alone guarantees that if many packets try to get to the same target, one of them is successful. Consider the *n* packets that are being sent to their random destinations at a given moment of time. The probability that none of the *n* packets tries to arrive at a fixed target is $(1-1/n)^n \to 1/e$. Hence, with probability *1-1/e* there are packets coming to this destination, and at least one of them is successful. Thus, the expectation for a packet to succeed is at least *(e-1)/2e*.

The protocol P2 is actually better that the lower limit suggests, since in many case two packets arrive to a given target. Analyzing the situation would be easier, if we additionally define (protocol P3) that packets are not allowed to cross over the diagonal. In that case we can conclude that in random case there is at least one packet arriving from the left side to the target with probability *1-1/e*. The same holds for the other side. Thus, the expectation for a packet to succeed is $(e-1)/e \approx 0.63$.

It is interesting to observe that similar results [5] were proven for anti-diagonal routing protocol: A protocol were packets all the time try to ``walk'' along the edges of rectangular area.

Homogeneous set

The other way to decrease the number of packets able to push a packet is to assume that the initial distribution of packets is by no means random. This is possible, if the nodes have a lot of packets to send and it is possible to maintain some kind of sorted order of them.

Consider a case, where each source sends two packets whose target is *k+l* steps away - first is such that it needs to go *k* steps horizontally and *l* steps vertically, and vice versa for the other packet. In such case, it is quite easy to see that even the protocol P1 moves the packets so that they all reach their diagonal after |*k-l*| steps, and after that they move without any conflicts (almost along the diagonal) to their target. The situation can be modeled with Equation 1 - using probability *p*=1.0, which by Theorem 1b gives expectation 1 for a packet to reach its target.

A natural relaxation of this setting is to assume that each source only sends packets that are targeted to an area of size $\Delta > 1$ for both sides of each sender (in the above case $\Delta = 1$). This allows us to construct settings for which the probability *p* in Equation 1 varies between 0.75 and 1.0.

CONCLUSIONS AND FUTURE WORK

We have proved a solution for a recurrence equation that can be used to model several routing situations in tori networks. We used the recurrence equation as a basis for designing routing protocols and situations. Considering the equation gives hints how to improve routing.

Extending the results to two-way sparse tori is straightforward. As a future work we suggest extending the results to routing situations in ordinary tori.

REFERENCES

[1] C. Bartzis, I. Caragiannis, C. Kaklamanis, I. Vergados. Experimental Evaluation of Hot-Potato Routing Algorithms on 2-Dimensional Processor Arrays. In *Proceedings of Euro-Par 2000*, pages 877 - 881, 2000.

[2] C. Busch, M. Herlihy, R. Wattenhofer. Randomized Greedy Hot-Potato Routing. In *Proceedings of 11th annual ACM-SIAM Symposium on Discrete Algorithms*, pages 458 - 466, 2000.

[3] B.S. Chelebus, A. Czumaj, L. Gasieniec, M. Kowaluk, W. Plandowski. Parallel Alternating-Direction Access Machine. In *Proceedings of Mathematical Foundations of Computer Science, LNCS 1113*, pages 267 - 279, 1996.

[4] M.D. Grammatikakis, D.F. Hsu, M. Kraetzl, J.F. Sibeyn. Packet Routing in Fixed-Connection Networks: A Survey. *Journal of Parallel and Distributed Computing*, 54(2):77 - 132, 1998.

[5] R. Honkanen, V. Leppänen, M. Penttonen. Hot-Potato Routing Algorithms for Sparse Optical Torus. In *Proceedings of International Conference on Parallel Processing*, pages 302 -- 307, 2001.

[6] D.E.Knuth. The Art of Computer Programming, Vol. 1. Addison-Wesley, 1968.

[7] V. Leppänen and M. Penttonen. Work-Optimal Simulation of PRAM Models on Meshes. *Nordic Journal on Computing*, 2(1):51 -- 69, 1995.

[8] R. Motwani and P. Raghavan. Randomized Algorithms. Cambridge University Press, 1995.

[9] H. Schröder, O. Sykora, and I. Vrto. Optimal Embedding of a Toroidal Array in a Linear Array. In *Foundations of Computation Theory*, pages 390 - 394, 1991.

[10] A. Schuster. Bounds and Analysis Techniques for Greedy Hot-Potato Routing, Chapter 11, pages 283 - 354. Academic Publishers, 1997.

[11] J. Sibeyn. Solving Fundamental Problems on Sparse-Meshes. In *Scandinavian Workshop on Algorithm Theory, SWAT'98, LNCS 1432*, pages 288 -- 299, 1998.

ABOUT THE AUTHORS

Prof. Ville Leppänen, PhD, Department of Information Technology, University of Turku, Phone: +358 2 333 8676, E-mail: Ville.Leppanen@it.utu.fi.

Prof. Martti Penttonen, PhD, Department of Computer Science, University of Kuopio, Phone: +358 17 16 2560, E-mail: Martti.Penttonen@cs.uku.fi.