A Succinct Story of Compact Stuff

by Cyril Gavoille

FRAIGNIAUD Workshop

[**F**undamental **R**esearch and **A**lgorithmic **I**nnovations in **G**raphs, **N**etworks and the **I**nternet, with **A**pplications and **U**pcoming **D**irections]

Paris – IRIF

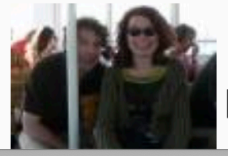
Novembre 28th, 2022

Pierre FRAIGNIAUD

Current and Former PhD Students

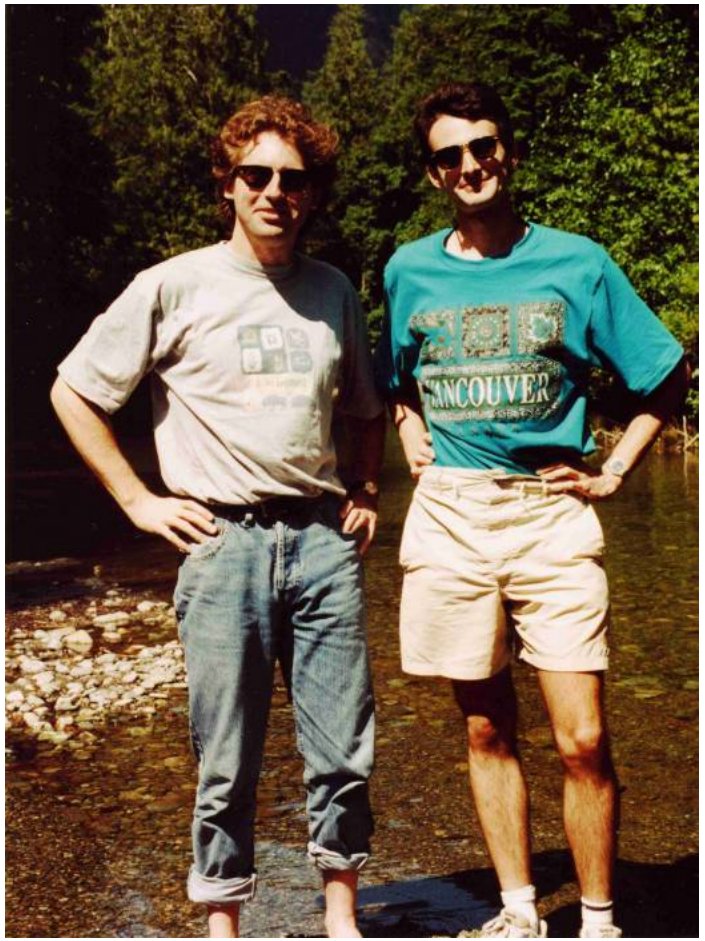
[1996] Cyril Gavoille (Professeur, University of Bordeaux)

[1996] Eric Fleury (Professeur, ENS Lyon)



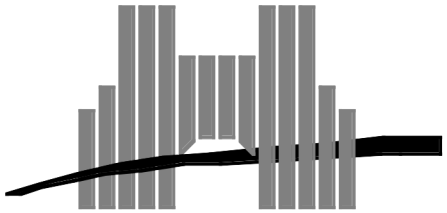
[1997] Sandrine Vial (Maitresse de Conférence, University of Versailles)

The Origins



1994 – Vancouver (CAN)

1991 - Master’s internship (M. Gastaldo, P. Fraigniaud)



Ecole Normale Supérieure de Lyon

46, Allée de l'Italie, 69364 Lyon Cedex 07, France
Téléphone : +33 72 72 80 00; Télécopieur :+33 72 72 80 80
Adresses électroniques :
lip@frensl61.bitnet; lip@lip.ens-lyon.fr (uucp)

Laboratoire de l'Informatique du Parallélisme

Ecole Normale Supérieure de Lyon
Institut IMAG
Unité de Recherche Associée au CNRS n° 1398

1991 - Master’s internship (M. Gastaldo, P. Fraigniaud)

Evaluation of the MasPar Performances

Cyril GAVOILLE

Training Course for a Masters in Computer Science

Claude Bernard University

- 1991 -

Produced within the LIP

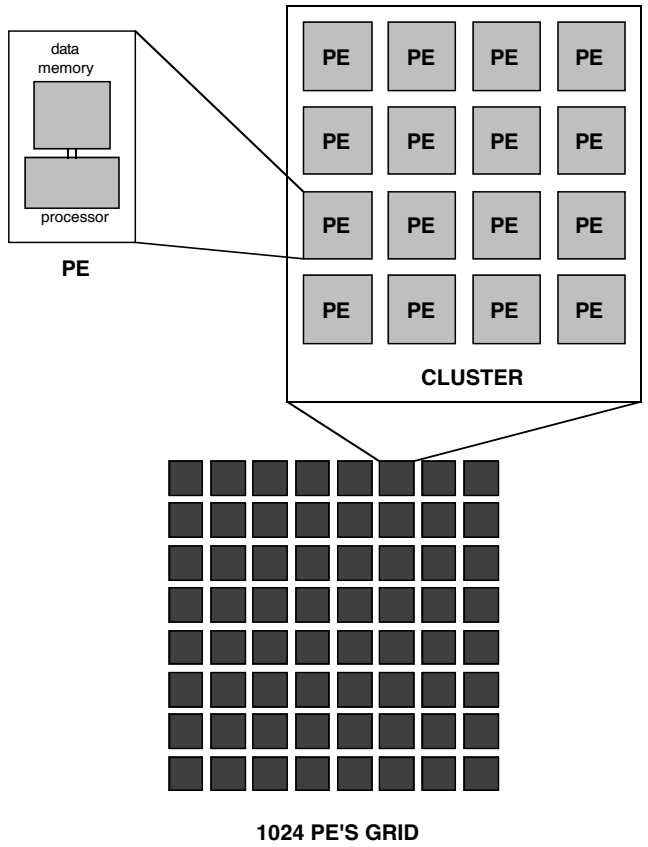


Figure 2.1: Processors grid regrouped in clusters

1991 - Master’s internship (M. Gastaldo, P. Fraigniaud)

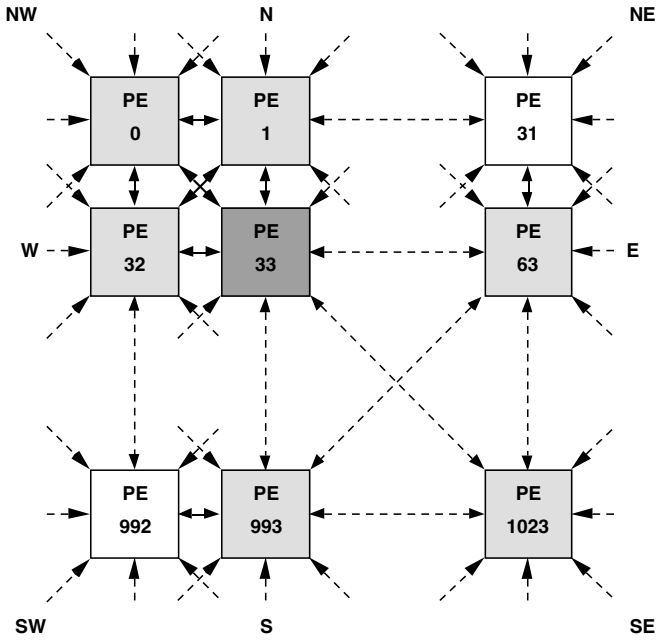


Figure 2.2: The physical connections between PE's and the 8 directions of the X-Net communications

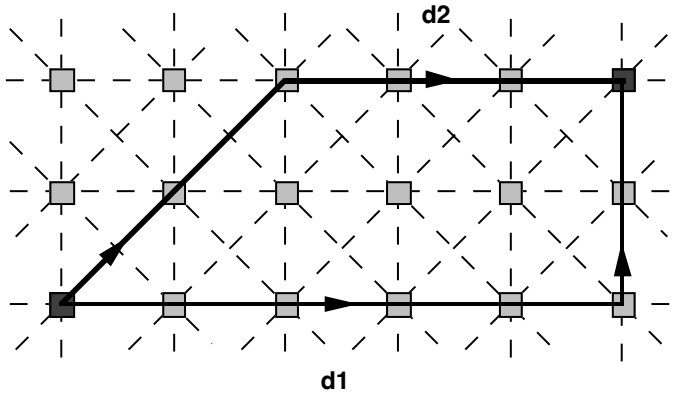


Figure 4.5c: Two distance models (`d1` and `d2`) for `xsend` and `xfetch` communications.

`xsend` and `xfetch` communication Results

Just as the `xnet` communications, the `xsend` and `xfetch` instructions depend on the distances, which are `dy` and `dx`. They also depend on the number of transferred bits (see the size of the simple types of the variables table). We will also notice the importance of left-values (that is `xsend`). We can also say that the times depend on the parameter *type* (`ss`, `ps`, `sp`, or `pp`; see syntax of these communications). These times decrease when the pointers (`scr` and `dest`) are in the ACU memory. When all the variable addresses are the same, the pointer is then situated in the ACU and the sequencer can then order a transfer of bytes directly to all the PE's. But if the variable addresses are not the same for all PE's, the sequencer can only send, to the PE's, instructions for adresses decoding and then send transfer instructions. So then the PE's have to perform more instructions.

Results on X-Net type communications

An X-Net communication used in *left-value* is always quicker than the same instruction *right-value*. Performances depend neither on direction nor on the

Interval Routing Time

1995 – Luminy (FRA)

1992-93 – Master2's internship

Stage de DEA

Routage par Intervalles

Cyril Gavoille
LIP
Ecole Normale Supérieure de Lyon
69364 Lyon Cedex 07, France
Cyril.Gavoille@lip.ens-lyon.fr
Responsable : Pierre Fraigniaud

28 juin 1993

Résumé

Dans ce rapport, nous présentons une classification des fonctions de routage par intervalles. Nous construisons alors une hiérarchie de graphes induite par les propriétés des fonctions de routages par intervalles de ces graphes. Le but de cette étude étant de répondre à des questions du type : "quels sont les graphes qui admettent telle propriété sur la fonction de routage par intervalles ?" Cette construction est partiellement obtenue à partir de résultats connus dans la littérature, mais également à partir de résultats originaux présentés dans ce rapport. Enfin, nous étudions les cas de graphes particuliers comme la Grille, l'Hypercube, le *CCC* ou le Shuffle-Exchange.

1 Introduction

1.1 Généralités sur le routage

Le routage consiste à faire transiter un message dans un réseau, d'un processeur à un autre (communication point à point), via éventuellement d'autres processeurs intermédiaires. Le choix de la *route* à suivre est déterminé par chacun des processeurs impliqués dans la communication à l'aide

1993 - Interval routing (PhD starts)

M.D. May and P.W. Thompson. Transputers and routers: Components for concurrent machines. In *Transputers and Routers: Function, Performance and Applications*, INMOS Ltd., April 1990

1 Transputers and Routers: Components for Concurrent Machines

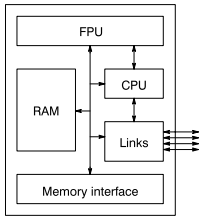
1.1 Introduction

This chapter describes an architecture for concurrent machines constructed from two types of component: 'transputers' and 'routers'. In subsequent chapters we consider the details of these two components, and show the architecture can be adapted to include other types of component.

A transputer is a complete microcomputer integrated in a single VLSI chip. Each transputer has a number of communication links, allowing transputers to be interconnected to form concurrent processing systems. The transputer instruction set contains instructions to send and receive messages through these links, minimizing delays in inter-transputer communication. Transputers can be directly connected to form specialised networks, or can be interconnected via routing chips. Routing chips are VLSI building blocks for interconnection networks: they can support system-wide message routing at high throughput and low delay.

1.2 Transputers

VLSI technology enables a complete computer to be constructed on a single silicon chip. The INMOS T800 transputer [1], integrates a central processor, a floating point unit, four kilobytes of static random access memory plus an interface for external memory, and a communications system onto a chip about 1 square centimetre in area.



T800 Transputer

As a microcomputer, the transputer is unusual in that it has the ability to communicate with other transputers via its communication links; this enables transputers to be connected together to construct multiprocessor systems to tackle specific problems. The transputer is also unusual in that it has the ability to execute many software processes, sharing its time between them automati-

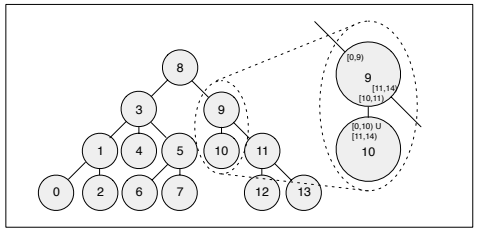


Figure 1.8 A Tree with Interval Labelling

Hypercubes can be labelled

The labelling of the hypercube follows the construction given for the deadlock free routing algorithm. In combining the two order-n hypercubes H_1 and H_2, the transputers in H_1 are labelled $0, \ldots, 2^n-1$ and those in H_2 are labelled $2^n, \ldots 2^{n+1}-1$. The link from each node h_1 in H_1 to the corresponding node h_2 in H_2 is labelled with the interval $[2^n, \ldots 2^{n+1})$ at h_1, and with $[0, \ldots 2^n)$ at h_2. This inductively constructs a hypercube together with the deadlock-free routing algorithm described above.

Arrays can be labelled

The labelling for an array follows the construction of the deadlock free routing algorithm. An n-dimensional array is composed of m arrays of dimension $n-1$, with m corresponding nodes (one from each $n-1$ dimensional array) joined to form a line. If each of the $n-1$ dimensional arrays has p nodes, the nodes in the $n-1$ dimensional arrays are numbered $0, \ldots, p-1; p, \ldots, 2p-1; \ldots; (m-1)p, \ldots, mp-1$. On every line the link joining the i^{th} node to the $(i+1)^{th}$ node is labelled $[ip, \ldots, mp)$ and the link to the $(i-1)^{th}$ node is labelled $[0, \ldots, (i-1)p)$. This inductively labels an array to route packets according to the deadlock free algorithm described above. An example is shown in figure 1.9. This shows the labels assigned to each node, and the intervals assigned to the links of one of the nodes.

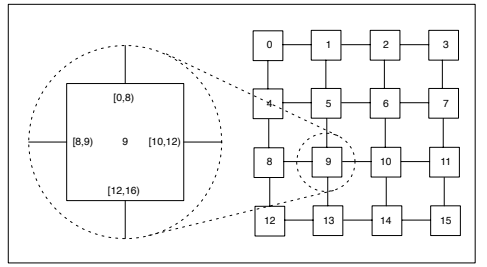


Figure 1.9 An Array with Interval Labelling

Labelling arbitrary networks

The above labelings provide optimal routing, so that each packet takes one of the shortest paths to its destination. It can easily be shown [6] that any network can be labelled so as to provide deadlock free routing; it is only necessary to construct a spanning tree and label it as described above. This may produce a non-optimal routing which cannot exploit all of the links present in the network as a whole. Optimal labelings are known for all of the networks shown below:

trees
hypercubes
arrays
multi-stage networks
butterfly networks
rings[3]

In high performance embedded applications (or in reconfigurable computers) specialised networks are often used to minimize interconnect costs or to avoid the need for message routing. In these systems, a non-optimal labelling can be used to provide low-speed system-wide communications such as would be needed for system configuration and monitoring.

1.5.2 Header Deletion

The main disadvantages of the interval labelling system are that it does not permit arbitrary routes through a network, and it does not allow a message to be routed through a series of networks. These problems can be overcome by a simple extension: *header deletion*. Any link of a router can be set to delete the header of every packet which passes out through it; the result is that the data immediately following becomes the new header as the packet enters the next node.

Header deletion can be used to minimize delays in the routing network. To do this, an initial header is used to route the packet to a destination transputer; this header is deleted as it leaves the final router and enters the transputer. A second header is then used to identify the virtual link within

3. Note that the optimal labelling of a ring requires that one of the connections be duplicated in order to avoid deadlock.

Optimal Interval Routing

Pierre Fraigniaud[1] and Cyril Gavoille[1]

LIP - CNRS
Ecole Normale Supérieure de Lyon
69364 Lyon Cedex 07, France

Abstract. Interval routing was introduced to reduce the size of the

A Characterization of Networks Supporting Linear Interval

Pierre Fraigniaud* and Cyril Gavoille+

LIP - CNRS
Ecole Normale Supérieure de Lyon
69364 Lyon Cedex 07, France
{pfraign,gavoille}@lip.ens-lyon.fr

Abstract

Compact routing tables are useful to implement routing algorithms on a distributed memory parallel computer. Interval routing is a popular way of building such compact tables. It was already known that any network can support an interval routing function with only one interval per output port as soon as one allows intervals to be "cyclic" [13]. However, it might be interesting for practical reasons to allow only the use of "linear" intervals (see [2]). This notion is particularly useful to derive results on networks built by cartesian products (as hypercubes and torus) [4]. In this paper, we characterize the networks that admit a *linear* interval routing function with at most one interval per output port. We also characterize the networks that admit a *strict* linear interval routing function with at most one interval per output port.

1 Introduction

If the structure of the interconnection network of a distributed memory parallel computer is fixed but complicated (a pancake graph, an undirected de Bruijn graph, etc) or if the interconnection network has no particular structure, it could be difficult to find a "simple algorithmic" way to find locally a shortest path between two nodes.

By a simple algorithm, we mean an algorithm such that both execution time and space for implementation in the router are small. A solution to this problem is obtained by the use of routing tables which are stored locally on each router. Of course, the memory requirement for these tables is to be as small as possible. For instance a size of $\Theta(n)$ for a network of n nodes is not realistic as soon as the number of nodes is larger that some tens).

Compact routing has already been intensively studied (see for instance [1, 7, 8]). In particular, there are many solutions to compress the size of routing tables. The general idea is to group in some way the destination addresses that correspond to the same output port, and to encode the group so as to be able to check if a destination address belongs to a group. A very popular solution of that kind is the use of intervals [12]. They are indeed very efficient in space (only store the bounds of each interval) and only two comparisons are enough to check if a destination address belongs to an interval. This kind of routing is used for instance on the C104 routing chip from Inmos.

The notion of interval routing has been introduced by Santoro and Khatib in [12]. They showed in particular that any directed acyclic network can support a 1-interval routing function with shortest paths and with one interval per output port. Moreover, they showed that if the network is not acyclic, they show that there exists a shortest path routing function such that the maximum number of intervals used to route between two vertices is at most the network diameter. Then van Leeuwen and Tan [13] studied the problem for undirected networks. They showed that any network supports an interval routing func-

* The first author received the support of the Centre de Recerca Matemàtica, Institut d'Estudis Catalans, Bellaterra, Spain.
+ Both authors are supported by the research programs ANM and C3.

Interval Routing Schemes allow Broadcasting with Linear Message-Complexity*

[Extended Abstract]

Pierre Fraigniaud† Cyril Gavoille‡ Bernard Mans§

ABSTRACT
The purpose of *compact*
the nodes of a network,
tables so that routing can
shortest paths) while keep
store the routing tables
per, we answer a long-sta
compact routing can also
putations. In particular, w
a shortest path *interval r*
with an $O(n)$ message-co
of nodes of the network.
$O(n)$ messages suffice to s
labeled by a shortest path
ing therefore the $O(m+n$

Keywords: Compact rou
ing, Distributed computin

1. INTRODUCTIO

This paper addresses a p
D. Peleg, and that can be
"Does networks supportin
schemes present specific a
putation? E.g., broadcasti

*Part of this work has b
was visiting the Compute
sity Paris-Sud at LRI, su
ARC/CNRS cooperation
†Laboratoire de Recherche
Paris-Sud, 91405 Orsay c
by the CNRS. http://ww
‡Laboratoire Bordelais de
Bordeaux I, 33405 Talen
support by the Aquitai
http://dept-info.labri
§Department of Com
Macquarie Univ., Syd
http://www.comp.mq.edu

Distrib. Comput. (2001) 14: 217–229

Interval routing schemes allow broadcasting with linear message-complexity

Pierre Fraigniaud[1,*], **Cyril Gavoille**[2,**], **Bernard Mans**[3,***]

[1] Laboratoire de Recherche en Informatique, Bât. 490, Université Paris-Sud, 91405 Orsay c
[2] Laboratoire Bordelais de Recherche en Informatique, Université Bordeaux I, 33405 Talen
[3] Department of Computing, Division of ICS, Macquarie University, Sydney, NSW 2109, A

Received: December 2000 / Accepted: July 2001

Summary. The purpose of *compact routing* is to provide a labeling of the nodes of a network and a way to encode the routing tables, so that routing can be performed efficiently (e.g., on shortest paths) whilst keeping the memory-space required to store the routing tables as small as possible. In this paper, we answer a long-standing conjecture by showing that compact routing may also assist in the performance of distributed computations. In particular, we show that a network supporting a shortest path *interval routing scheme* allows broadcasting with a message-complexity of $O(n)$, where n is the number of nodes of the network. As a consequence, we prove that $O(n)$ messages suffice to solve leader-election for any graph labeled by a shortest path interval routing scheme, improving the previous known bound of $O(m+n)$. A general consequence of our result is that a shortest path interval routing scheme contains ample *structural information* to avoid developing ad-hoc or network-specific solutions for basic problems that distributed systems must handle repeatedly.

Key words: Compact routing – Interval routing – Broadcasting – Distributed computing

1 Introduction

This paper addresses a problem originally formulated by D. Peleg that can be informally summarized as follows: "Do networks supporting shortest path compact routing schemes present specific ability in term of distributed computation? E.g., broadcasting, leader election, etc." This paper answers this question in t
works supportin
short), allow bro

Formally, a n
work, we will alw
out self loops an
of that network
way that the foll
degree d and lab
$\{1,\ldots,n\}$, one
that (1) $\{1,\ldots,n$
$i \neq j$ and (3) ℓ'
x to the node la
encode shortest p
set of destination
secutive integers
compact routing
supporting an IR
in comparison to
every destination
sponding to that
18,20,23,24], an
routing in gener
mainder of this p
will make no dist
other words, we
of node x in the

Broadcasting
formation dissem
given message to
of broadcasting i
the reception of t
quires at least $n-$
be performed by
tion of the messa
all its incident ed
received the mess
be derived as a fu
network and of th
see [1,2,5]). In th
is its label in son
dent edges in the

Part of this work was completed while the third author was visiting the Computer Science Department of University Paris-Sud at LRI, supported by the Australian-French ARC-IREX/CNRS cooperation #99N92/0523. A preliminary version of this paper was presented at the 19th ACM Symposium on Principles of Distributed Computing (PODC 2000).

* Additional support by the CNRS
** Additional support by the Aquitaine Region project #98024002
*** Additional support by the ARC

Algorithmica (1998) 21: 155–182

Algorithmica

Interval Routing Schemes[1]

P. Fraigniaud[2] and C. Gavoille[2]

Abstract. Interval routing was introduced to reduce the size of routing tables: a router finds the direction where to forward a message by determining which interval contains the destination address of the message, each interval being associated to one particular direction. This way of implementing a routing function is quite attractive but very little is known about the topological properties that must satisfy a network to support an interval routing function with particular constraints (shortest paths, limited number of intervals associated to each direction, etc.). In this paper we investigate the study of the interval routing functions. In particular, we characterize the set of networks which support a *linear* or a *linear strict* interval routing function with only one interval per direction. We also derive practical tools to measure the efficiency of an interval routing function (number of intervals, length of the paths, etc.), and we describe large classes of networks which support *optimal* (linear) interval routing functions. Finally, we derive the main properties satisfied by the popular networks used to interconnect processors in a distributed memory parallel computer.

Key Words. Routing in distributed networks, Compact routing, Routing function, Interval.

1. Introduction. Given a network of processors (such as the one of a distributed memory parallel computer), the way of routing messages among the processors is characterized, on one hand, by the routing *mode* (store-and-forward, circuit-switched, wormhole, . . .) and, on the other hand, by the routing *function* which determines the paths between the sources and the destinations. This paper focuses on the second parameter.

The routing function is generally implemented locally on the routers. The route of a message from its source to its destination is determined using a header attached to the message, and which contains information that will allow the intermediate routers to know where to forward the message. In this paper we are interested in routing functions which use only the destination address of the message to find the route.

As soon as a router receives a message, it looks at the header to read the destination, and then determines the output port which will be used to forward the message toward its destination. There are mainly two ways of determining the output port from the destination address:

1. Application of an algorithm.
2. Consultation of a routing table.

[1] All the results presented in this paper are entirely based on [8]. The first author received the support of the Centre de Recerca Matemàtica, Institut d'Estudis Catalans, Bellaterra, Spain. Both authors are supported by the research programs ANM and PRS of the CNRS.
[2] LIP - CNRS, École Normale Supérieure de Lyon, 46 allée d'Italie, 69364 Lyon Cedex 07, France. {pfraign,gavoille}@ens-lyon.fr.

Received February 7, 1996; revised November 25, 1996. Communicated by F. T. Leighton.

Pierre Fraigniaud and Cyril Gavoille. A characterization of networks supporting linear interval routing. In *13th Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 216-224. ACM Press, August 1994.

A Characterization of Networks Supporting Linear Interval Routing

Pierre Fraigniaud* and Cyril Gavoille+

LIP - CNRS
Ecole Normale Supérieure de Lyon
69364 Lyon Cedex 07, France
{pfraign,gavoille}@lip.ens-lyon.fr

Abstract

Compact routing tables are useful to implement routing algorithms on a distributed memory parallel computer. Interval routing is a popular way of building such compact tables. It was already known that any network can support an interval routing function with only one interval per output port as soon as one allows intervals to be "cyclic" [13]. However, it might be interesting for practical reasons to allow only the use of "linear" intervals (see [2]). This notion is particularly useful to derive results on networks built by cartesian products (as hypercubes and torus) [4]. In this paper, we characterize the networks that admit a *linear* interval routing function with at most one interval per output port. We also characterize the networks that admit a *strict* linear interval routing function with at most one interval per output port.

1 Introduction

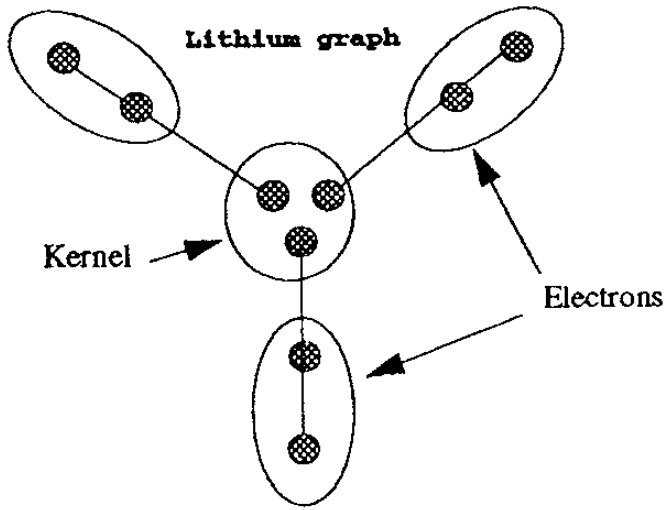
If the structure of the interconnection network of a distributed memory parallel computer is fixed but complicated (a pancake graph, an undirected de Bruijn graph, etc) or if the interconnection network has no particular structure, it could be difficult to derive a "simple algorithmic" way to find locally the path between two nodes.

By a simple algorithm, we mean an algorithm whose both execution time and space for implementing it on the router are small. A solution to that problem is obtained by the use of routing tables that are stored locally on each router. Of course, the main requirement for these tables is to be as small as possible (for instance a size of $\Theta(n)$ for a network of n processors is not realistic as soon as the number of processors is larger that some tens).

Compact routing has already been intensively studied (see for instance [1, 7, 8]). In particular, there exist many solutions to compress the size of the routing tables. The general idea is to group in some manner the destination addresses that correspond to the same output port, and to encode the group so that it is easy to check if a destination address belongs to a given group. A very popular solution of that kind is the use of intervals [12]. They are indeed very simple to code (only store the bounds of each interval) and at most two comparisons are enough to check if a destination address belongs to an interval. This kind of routing is used for instance on the C104 routing chip [11, 3] of Inmos.

The notion of interval routing has been introduced by Santoro and Khatib in [12]. They mainly show that any directed acyclic network can support an interval routing function with shortest paths and only one interval per output port. Moreover, if the digraph is not acyclic, they show that there exists an interval routing function such that the maximum length of the route between two vertices is at most two times the diameter. Then van Leeuwen and Tan [13] studied the problem for undirected networks. They showed that any network supports an interval routing func-

* *The first author received the support of the Centre de Recerca Matemàtica, Institut d'Estudis Catalans, Bellaterra, Spain.*
+ *Both authors are supported by the research programs ANM and C3.*



Theorem 2 $G \in 1\text{-}LIRS \Leftrightarrow G$ *is not a lithium-graph.*

Pierre Fraigniaud, Cyril Gavoille, and Bernard Mans. Interval routing schemes allow broadcasting with linear message-complexity. *Distributed Computing*, 14(4):217-229, July 2001. Ext. abstract in PODC'00.

Distrib. Comput. (2001) 14: 217–229

DISTRIBUTED COMPUTING

Interval routing schemes allow broadcasting with linear message-complexity

Pierre Fraigniaud[1,*], **Cyril Gavoille**[2,**], **Bernard Mans**[3,***]

[1] Laboratoire de Recherche en Informatique, Bât. 490, Université Paris-Sud, 91405 Orsay cedex, France
[2] Laboratoire Bordelais de Recherche en Informatique, Université Bordeaux I, 33405 Talence cedex, France
[3] Department of Computing, Division of ICS, Macquarie University, Sydney, NSW 2109, Australia

Received: December 2000 / Accepted: July 2001

Summary. The purpose of *compact routing* is to provide a labeling of the nodes of a network and a way to encode the routing tables, so that routing can be performed efficiently (e.g., on shortest paths) whilst keeping the memory-space required to store the routing tables as small as possible. In this paper, we answer a long-standing conjecture by showing that compact routing may also assist in the performance of distributed computations. In particular, we show that a network supporting a shortest path *interval routing scheme* allows broadcasting with a message-complexity of $O(n)$, where n is the number of nodes of the network. As a consequence, we prove that $O(n)$ messages suffice to solve leader-election for any graph labeled by a shortest path interval routing scheme, improving the previous known bound of $O(m+n)$. A general consequence of our result is that a shortest path interval routing scheme contains ample *structural information* to avoid developing ad-hoc or network-specific solutions for basic problems that distributed systems must handle repeatedly.

Key words: Compact routing – Interval routing – Broadcasting – Distributed computing

1 Introduction

This paper addresses a problem originally formulated by D. Peleg that can be informally summarized as follows: "Do networks supporting shortest path compact routing schemes present specific ability in term of distributed computation? E.g., broadcasting, leader election, etc." This paper answers this question in the affirmative, by showing that n-node networks supporting interval routing schemes [27,28] (IRS for short), allow broadcasting with $O(n)$ message-complexity.

Formally, a network $G = (V, E)$ (in this paper, by *network*, we will always mean a connected undirected graph without self loops and multi-edges) supports an IRS if the nodes of that network can be labeled from 1 to $n = |V|$ in such a way that the following is satisfied: given any node $x \in V$ of degree d and label ℓ, there is a set of d intervals $I_1, \ldots, I_d$ of $\{1, \ldots, n\}$, one for each edge $e_1, \ldots, e_d$ incident to x, such that (1) $\{1, \ldots, n\} \setminus \{\ell\} \subseteq \bigcup_{i=1}^{d} I_i$, (2) $I_i \cap I_j = \emptyset$ for every $i \neq j$ and (3) $\ell' \in I_i$ implies that there is a shortest path from x to the node labeled ℓ' passing through the edge e_i. (IRS encode shortest path routing tables with the property that the set of destination-addresses using a given link is a set of consecutive integers.) IRS are well-known in the framework of compact routing since a network of maximum degree Δ, and supporting an IRS, has routing tables of size $O(\Delta \log n)$ bits, in comparison to the $\Theta(n \log \Delta)$ bits of a table returning, for every destination label $i \in \{1, \ldots, n\}$, the output port corresponding to that label. For more about IRS, we refer to [7,11, 18,20,23,24], and to the survey [17]. For more about compact routing in general, we refer to [13–15,19,21,26]. In the remainder of this paper, given a network supporting an IRS, we will make no distinction between the nodes and their labels. In other words, we will assume $V = \{1, \ldots, n\}$ where the label of node x in the IRS is precisely $x \in \{1, \ldots, n\}$.

Broadcasting for an arbitrary node of a network is the information dissemination problem which consists of sending a given message to all the other nodes. The message-complexity of broadcasting is between $\Omega(n)$ and $O(m)$, $m = |E|$, since the reception of the message by every node but the source requires at least $n-1$ messages, and yet, broadcasting can always be performed by flooding the network (meaning, upon reception of the message, every node forwards that message through all its incident edges apart from the one through which it has received the message). Improved upper and lower bounds may be derived as a function of the knowledge of the nodes of the network and of the maximal size of the message-headers (e.g., see [1,2,5]). In this paper, the only knowledge of every node is its label in some IRS and the intervals attached to its incident edges in the same IRS. The size of each message-header

Part of this work was completed while the third author was visiting the Computer Science Department of University Paris-Sud at LRI, supported by the Australian-French ARC-IREX/CNRS cooperation #99IN92/0523. A preliminary version of this paper was presented at the 19th ACM Symposium on Principles of Distributed Computing (PODC 2000).

* Additional support by the CNRS
** Additional support by the Aquitaine Region project #98024002
*** Additional support by the ARC

Corollary 3 *In a network supporting a shortest path Interval Routing Scheme, the average distance between two nodes labeled by two consecutive integers is bounded by a constant.*

18

Theorem 4 *Networks supporting a shortest-path interval routing scheme allow leader-election with $O(n)$ message-complexity.*

Compact Routing Time

2012 – Salvador (BRE)

1995-2001 - Routing with little information at each node in every graph ...

Memory Requirement for Universal Routing Schemes

Pierre Fraigniaud and Cyril Gavoille
{pfraign, gavoille}@lip.ens-lyon.fr
Laboratoire de l'Informatique du Parallélisme - CNRS
École Normale Supérieure de Lyon
69364 Lyon cedex 07
France

Abstract

In this paper, we deal with the compact routing p
lem, that is implementing routing schemes that u
minimum memory size on each router. In [20], P
and Upfal showed that there is no hope to do that
less than a total $\Omega(n^{1+1/(2s+4)})$ memory bits for
stretch factor $s \geq 1$. We improve this bound for st
factors $s < 2$ by proving that any near-shortest
routing scheme uses a total of $\Omega(n^2)$ memory bits.

1 Introduction

The *general routing problem* in a network (as oppose
the *permutation routing problem* [18] or the *broad*
ing problem [8]) consists of finding a routing *protoc*
routing *function* such that, for any source-destina
pair, any message from the source can be routed t
destination. XY-routing [2] or e-cube routing [3
such protocols. The efficiency of a protocol is meas
in terms of latency (related to the length of the pa
and/or throughput (related to link or node congest
Finding shortest paths in a network is easy [14],
finding paths minimizing congestion can be done e
for a large class of networks (namely Cayley graphs)
However, the routings protocols that are obtained
be of no use from a VLSI point of view. Indeed,
implementation of these protocols may require su
large amount of hardware that they cannot be use
practice.

For instance, implementing XY or e-cube rou
is simple because routing is locally performed by
spectively comparison and XOR-ing of the local ad
with the destination address. However, networks d
necessarily have the simple structure of meshes o
percubes. For instance routing in *pancake* network
quires to decompose any given permutation in a p
uct of particular ones, and this is definitely not easy
Problem 39 in [9]). Also, networks may even hav

Universal Routing Schemes

Pierre Fraigniaud and Cyril Gavoille *

Laboratoire de l'Informatique du Parallélisme - CNRS
École Normale Supérieure de Lyon
69364 Lyon cedex 07
France

Abstract

In this paper, we deal with the compact routing problem, that is implementing
schemes that use a minimum memory size on each router. A *universal* routing sch
scheme that applies to all n-node networks. In [31], Peleg and Upfal showed that one
implement a universal routing scheme with less than a total of $\Omega(n^{1+1/(2s+4)})$ memory
any given stretch factor $s \geq 1$. We improve this bound for stretch factors s, $1 \leq s$
proving that any near-shortest path universal routing scheme uses a total of $\Omega(n^2)$ mem
in the worst-case. This result is obtained by counting the minimum number of routing f
necessary to route on all n-node networks.

Moreover, and more fundamentally, we give a tight bound of $\Theta(n \log n)$ bits for
minimum memory requirement of universal routing scheme of stretch factors s, $1 \leq s <$
precisely, for any fixed constant ε, $0 < \varepsilon < 1$, there exists an n-node network G on
least $\Omega(n^\varepsilon)$ routers require $\Theta(n \log n)$ bits each to code any routing function on G o
factor < 2. This means that, whatever you choose the routing scheme, there exists a
on which one can not compress locally the routing information better than routing tab

1 Introduction

The *general routing problem* in a network (as opposed to the *permutation routing pr*
the *broadcasting problem* [13]) consists in finding a routing *protocol* or routing *functi*
for any source-destination pair, any message from the source can be routed to th
XY-routing [4] or e-cube routing [5] are such protocols. The efficiency of a protoco
in terms of latency (related to the length of the paths) and/or throughput (relat
node congestion). Finding shortest paths in a network is easy [22], and finding path
congestion can be done easily for a large class of networks (namely Cayley graphs) [
the routings protocols that are obtained may be of no use from a VLSI point of view
implementation of these protocols may require such a large amount of hardware that they can not
be used in practice.

*Both authors are supported by the research programs ANM and PRS of the CNRS, and by the DRET of the DGA.

A Theoretical Model for Routing Complexity

P. Fraigniaud
Université Paris Sud, France

C. Gavoille
Université Bordeaux I, France

Abstract

This paper introduces a formal model for studying the complexity of routing in networks. The aim of this model is to capture both time complexity and space complexity. In particular, the model takes into account the input and output facilities of routers. A *routing program* is a RAM-program with five additional instructions that allow to handle incoming and outgoing headers, and input and output ports. One of these five additional instructions, called `release`, captures the possible use of hardware facilities to speed up routing.

Using our model, we show that there are routing functions which, if compacted, would require an arbitrarily large computation time to be decoded. The *latency* is the sum of the time (in bit-operation) required at every intermediate node to establish the route. We also show that, in any n-node network of diameter D, the latency is bounded by $O(D+n^{1/k}\log n)$, for every constant $k \geq 2$. This latter result has to be compared with the latency of the routing tables which is $\Theta(D \log n)$.

1 Introduction

The search for compact routing tables deserves more and more attention as the interest in large telecommunication systems grows [5]. Most of the literature devoted to compact routing focuses on strategies to encode point-to-point connections between pairs of sites in a networks. E.g., interval routing, introduced in [16, 17], is a famous compact routing method which encodes the set of destination addresses as a union of intervals. This method allows to decrease the memory requirement for routing in any network to $O(\log n)$ bits per communication link. However, if one insists on shortest paths, this method may fail, and may then require as many bits as the routing tables [8]. Therefore, other methods have been introduced whose goal is to satisfy some tradeoff between the space complexity

Local Memory Requirement of Universal Routing Schemes

Pierre Fraigniaud and Cyril Gavoille *

Laboratoire de l'Informatique du Parallélisme - CNRS
École Normale Supérieure de Lyon
69364 Lyon cedex 07
France

Abstract

In this paper, we deal with the compact routing problem, that is the problem of implementing routing schemes that use a minimum memory size on each router. A *universal* routing scheme is a scheme that applies to all networks. In [13], Peleg and Upfal showed that one can not implement a universal routing scheme with less than a total of $\Omega(n^{1+1/(2s+4)})$ memory bits for any routing scheme satisfying that the maximum ratio between the lengths of the routing paths and the lengths of the shortest paths, that is the *stretch factor*, is bounded by s. In [6], Fraigniaud and Gavoille improve this bound by proving that universal routing schemes of stretch factors at most 2 use a total of $\Omega(n^2)$ memory bits in the worst-case. Recently, Gavoille and Pérennès [9] showed that, in fact, in the worst-case, $\Theta(n)$ routers of an n-node network may require up to $\Omega(n \log n)$ memory bits for shortest path routing. In this paper, we extend this result by showing that, for any constant ε, $0 < \varepsilon < 1$, $\Omega(n^\varepsilon)$ routers of an n-node network may require up to $\Omega(n \log n)$ memory bits even if each routing path is of length up to twice the distance between its source and its destination.

1 Introduction and statement of the problem

One of the most important measures of complexity of routing schemes is the size of the routing information that must be stored locally and globally in a network. In this paper, we use a similar model as the one introduced by Peleg and Upfal in [13]. A *routing function* R is a triple (I, H, P) consisting of *initialization*, *header*, and *port* functions, respectively. For any two distinct nodes u and v, R produces a path $u = u_0, u_1, \ldots, u_k = v$ of nodes, and a sequence $h_0, h_1, \ldots, h_k$ of headers such that $h_0 = I(u, v)$, $P(u_k, h_k) = \emptyset$, and for all i, $1 \leq i < k$, $H(u_i, h_i) = h_{i+1}$, $P(u_i, h_i) = (u_i, u_{i+1})$. We are interested in the distributed way of implementing routing functions. In this paper, we assume that nodes are labeled by integers in $\{1, \ldots, n\}$, and that the output ports of each node x are labeled by integers in $\{1, \ldots, \deg(x)\}$, where n and $\deg(x)$ are the total number of nodes in the network and the degree of the node x, respectively.

A *routing scheme* is a function that returns a routing function R for any network G. For instance, the *shortest path interval routing scheme* [14, 15] is the selection, for any network G, of a shortest path routing function R whose implementation is based on grouping in intervals the destination addresses associated to each arc, and whose aim is to minimize the number of intervals per link. For any network, the obtained routing function R may require a large number of intervals but it exists. The shortest path interval routing scheme is therefore said to be *universal* because it applies to all networks. On the contrary, the shortest path 1-interval routing scheme, that is the routing scheme that returns a shortest path routing function that can be implemented using a unique interval per link, is only *partial* in the sense that there exist networks for which no such routing scheme exists [5].

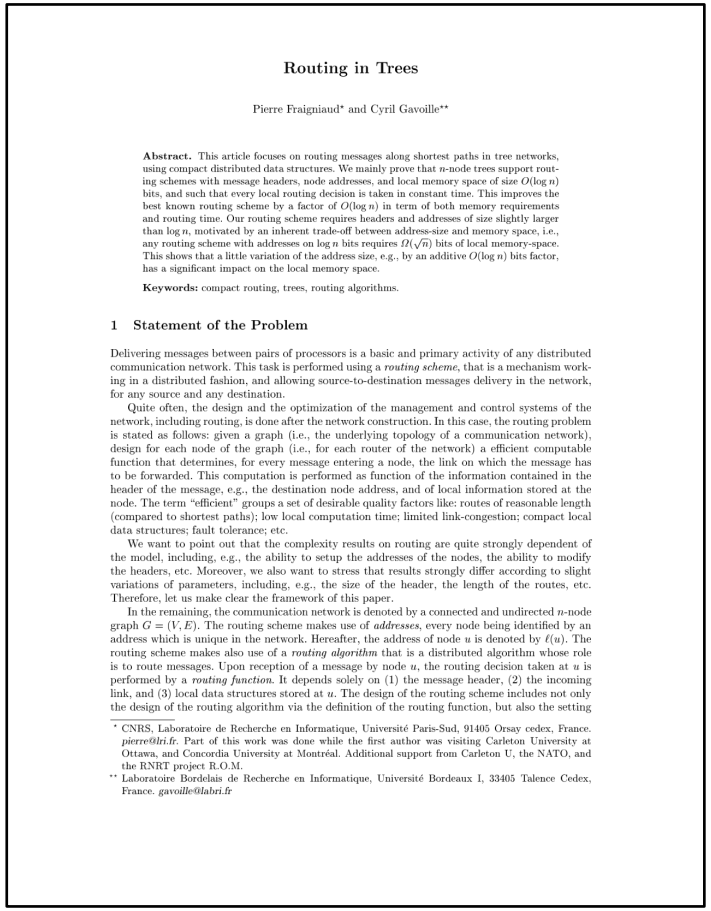
We consider the standard model of point-to-point communication networks described as finite connected symmetric digraphs $G = (V, E)$. In the following, we denote by n the number of vertices of such a graph. The vertices represent the processors or the nodes of the network, and the edges represent bidirectional communication links between the nodes (to each edge corresponds two symmetric arcs). A vertex can directly communicate with its neighbors only, and messages between nonadjacent vertices are sent along a path connecting them in the network. In the following, we always refer to *finite connected symmetric digraph* by the simple term *graph*.

Let G be any graph, and let x be any vertex of G. For every routing function R on G, we denote by $\mathrm{MEM}(G, R, x)$ the minimum *memory requirement* of R in x. The parameter $\mathrm{MEM}(G, R, x)$ is simply the Kolmogorov complexity [10] of the local computation of R in x. Of course, the memory requirement is defined for a fixed coding strategy. From the definition above, the memory requirement does not take into account the size of the header. Indeed, to be as general as possible, we allow headers to be of unbounded size. We then define, for any routing function R on a graph G:

*Both authors are supported by the research programs ANM and PRS of the CNRS, and by the DRET of the DGA.

2001 - Routing in trees … with $\Theta(\log^2 n/\log\log n)$ bits/node

2012 – Salvador (BRE)



Routing in Trees

Pierre Fraigniaud* and Cyril Gavoille**

Abstract. This article focuses on routing messages along shortest paths in tree networks, using compact distributed data structures. We mainly prove that n-node trees support routing schemes with message headers, node addresses, and local memory space of size $O(\log n)$ bits, and such that every local routing decision is taken in constant time. This improves the best known routing scheme by a factor of $O(\log n)$ in term of both memory requirements and routing time. Our routing scheme requires headers and addresses of size slightly larger than $\log n$, motivated by an inherent trade-off between address-size and memory space, i.e., any routing scheme with addresses on $\log n$ bits requires $\Omega(\sqrt{n})$ bits of local memory-space. This shows that a little variation of the address size, e.g., by an additive $O(\log n)$ bits factor, has a significant impact on the local memory space.

Keywords: compact routing, trees, routing algorithms.

1 Statement of the Problem

Delivering messages between pairs of processors is a basic and primary activity of any distributed communication network. This task is performed using a *routing scheme*, that is a mechanism working in a distributed fashion, and allowing source-to-destination messages delivery in the network, for any source and any destination.

Quite often, the design and the optimization of the management and control systems of the network, including routing, is done after the network construction. In this case, the routing problem is stated as follows: given a graph (i.e., the underlying topology of a communication network), design for each node of the graph (i.e., for each router of the network) a efficient computable function that determines, for every message entering a node, the link on which the message has to be forwarded. This computation is performed as function of the information contained in the header of the message, e.g., the destination node address, and of local information stored at the node. The term "efficient" groups a set of desirable quality factors like: routes of reasonable length (compared to shortest paths); low local computation time; limited link-congestion; compact local data structures; fault tolerance; etc.

We want to point out that the complexity results on routing are quite strongly dependent of the model, including, e.g., the ability to setup the addresses of the nodes, the ability to modify the headers, etc. Moreover, we also want to stress that results strongly differ according to slight variations of parameters, including, e.g., the size of the header, the length of the routes, etc. Therefore, let us make clear the framework of this paper.

In the remaining, the communication network is denoted by a connected and undirected n-node graph $G = (V, E)$. The routing scheme makes use of *addresses*, every node being identified by an address which is unique in the network. Hereafter, the address of node u is denoted by $\ell(u)$. The routing scheme makes also use of a *routing algorithm* that is a distributed algorithm whose role is to route messages. Upon reception of a message by node u, the routing decision taken at u is performed by a *routing function*. It depends solely on (1) the message header, (2) the incoming link, and (3) local data structures stored at u. The design of the routing scheme includes not only the design of the routing algorithm via the definition of the routing function, but also the setting

* CNRS, Laboratoire de Recherche en Informatique, Université Paris-Sud, 91405 Orsay cedex, France. *pierre@lri.fr*. Part of this work was done while the first author was visiting Carleton University at Ottawa, and Concordia University at Montréal. Additional support from Carleton U, the NATO, and the RNRT project R.O.M.

** Laboratoire Bordelais de Recherche en Informatique, Université Bordeaux I, 33405 Talence Cedex, France. *gavoille@labri.fr*

ICALP 2001

A Space Lower Bound for Routing in Trees

Pierre Fraigniaud[1] and Cyril Gavoille[2]

[1] CNRS, Laboratoire de Recherche en Informatique, Université Paris-Sud, 91405 Orsay cedex, France. *pierre@lri.fr*

[2] Laboratoire Bordelais de Recherche en Informatique, Université Bordeaux I, 33405 Talence cedex, France. *gavoille@labri.fr*

Abstract. The design of compact routing schemes in trees form the kernel of sophisticated strategies for compact routing in arbitrary graphs. This paper focuses on the space complexity for routing messages along shortest paths in trees. It was recently shown that the family of n-node trees supports routing schemes using addresses and routing tables of size $O(\log^2 n/\log\log n)$ bits per node, if the output port numbers of each node are chosen by an adversary. This paper shows that this result is tight, that is the sum of the sizes of the address and of the local routing table is at least $\Omega(\log^2 n/\log\log n)$ bits for some node of some tree.

Keywords: algorithms and data structures, computational and structural complexity, compact routing.

1 Introduction

It is well known, a *routing scheme* consists of two parts:

1. a pre-processing algorithm in charge of, giving a network G, setting data structures (e.g., routing tables, node addresses, port labeling, etc.) on which the routing decisions in G will be based;
2. a routing protocol, i.e., a distributed algorithm, running at each node of G, which mainly determines, for every message entering the node, the output port on which the message has to be forwarded. Note that the routing protocol may also be in charge of other tasks such as updating the headers, etc.

For instance, a classical routing scheme is the so-called *routing table* which stores a look-up table at each node. This table returns, for each possible destination address of a message, the output port on which the message has to be forwarded. To be more precise, upon reception of a message M, a node x considers the *header* attached to M. The header contains the *address* of the destination, say node y. This address is denoted by address(y). Node x then extracts from its look-up table the *output port number* p corresponding to the entry address(y). Finally M is forwarded on the incident link of x labeled by p. We note that the header is not modified here, and the entire routing of M is determined by the information address(y) originally stored in the header of M by the source node. The pre-processing algorithm is in charge of constructing the look-up tables. Giving different addresses between 1 and n to the nodes of an n-node network, and labeling the incident links of each node x by different integers between 1 and $\deg(x)$, yields look-up tables of size $O(n \log n)$. The question is: can we do better? I.e., can we design routing schemes requiring less space at each node of the network? The answer is "yes", and the main tool to achieve

SATCS 2002

Beyong Routing: Navigating

2009 – Carry-le-Rouet (FRA)

2003-2009 – Compact headers, Compact advices, Small-World and … Eclecticism!

Lower Bounds for Oblivious Single-Packet End-to-End Communication

Eclecticism Shrinks Even Small Worlds*

[Extended Abstract]

Pierre Fraigniaud† CNRS-LRI Univ. Paris-Sud & INRIA France

Cyril Gavoille LaBRI Univ. of Bordeaux & CNRS France

Christoph CNRS-L Univ. of Mo Franc

Distrib. Comput. (2006) 18(4): 279–291
DOI 10.1007/s00446-005-0137-4

SPECIAL ISSUE PODC 04

Pierre Fraigniaud · Cyril Gavoille · Christophe Paul

Eclecticism shrinks even small worlds

Submitted: 28 October 2004 / Accepted: 4 July 2005 / Published online: 3 February 2006
© Springer-Verlag 2005

Abstract We consider small world graphs as defined by Kleinberg (2000), i.e., graphs obtained from a d-dimensional mesh by adding links chosen at random according to the d-harmonic distribution. In these graphs, greedy routing performs in $O(\log^2 n)$ expected number of steps. We introduce *indirect-greedy* routing. We show that giving $O(\log^2 n)$ bits of topological awareness per node enables indirect-greedy routing to perform in $O(\log^{1+1/d} n)$ expected number of steps in d-dimensional augmented meshes. We also show that, independently of the amount of topological awareness given to the nodes, indirect-greedy routing performs in $\Omega(\log^{1+1/d} n)$ expected number of steps. In particular, augmenting the topological awareness above this optimum of $O(\log^2 n)$ bits would drastically decrease the performance of indirect-greedy routing.

Our model demonstrates that the efficiency of indirect-greedy routing is sensitive to the "world's dimension," in the sense that high dimensional worlds enjoy faster greedy routing than low dimensional ones. This could not be observed in Kleinberg's routing. In addition to bringing new light to Milgram's experiment, our protocol presents several desirable properties. In particular, it is totally *oblivious*, i.e., there is no header modification along the path from the source to the target, and the routing decision depends

Keywords Sma
experiment

1 Introduction

We consider sma
i.e., graphs obta
fixed $d \geq 1$, by
according to the
ity that x choos
$1/(Z_x \cdot \text{dist}(x, y)$
the mesh (i.e., th
malizing coeffici
This model aims
of separation"
gram [14], and
and Watts [5] (se
as well as leisure
ity, and religious
man multi-dime
possibly differen
sions should be
in the social gra
that the more c
should be Surp

Available online at www.sciencedirect.com
SCIENCE DIRECT®

Computer Networks 50 (2006) 1630–1638

ELSEVIER

Computer Networks

www.elsevier.com/locate/comnet

Header-size lower bounds for end-to-end communication in memoryless networks ☆

Pierre Fraigniaud [a,*], Cyril Gavoille [b]

[a] CNRS, Lab. de Recherche en Informatique, Université Paris-Sud, Bâtiment 490, 91405 Orsay Cedex, France

Distri
Informat

Pierre Fraigniau

[3] Départe

Abstract. W
vice) about a
fast distribut
to measure th
is informatio
rapidly (i.e.,
is informatio
nodes in orde
we study the

1 Introductio

This work is a pa
impact of *knowled*
(nodes of a distrib
as observed by Lini
computers, the lim
are specific to the d
analyzing the limit
computation. One
locally, i.e., when e
are at constant dis
locally [20]. This is
to some resource
distributed compu
computing an appr
dominating set.

The other frame
advice. In this fram

* A part of this wor

Distrib. Comput. (2009) 21:395–403
DOI 10.1007/s00446-008-0076-y

Distributed computing with advice: information sensitivity of graph coloring

Pierre Fraigniaud · Cyril Gavoille · David Ilcinkas · Andrzej Pelc

Received: 5 October 2007 / Accepted: 2 December 2008 / Published online: 9 January 2009
© Springer-Verlag 2009

Abstract We study the problem of the amount of information (advice) about a graph that must be given to its nodes in order to achieve fast distributed computations. The required size of the advice enables to measure the information sensitivity of a network problem. A problem is information sensitive if little advice is enough to solve the problem rapidly (i.e., much faster than in the absence of any advice), whereas it is information insensitive if it requires giving a lot of information to the nodes in order to ensure fast computation of the solution. In this paper, we study the inform
of distributed graph coloring.

Keywords Network algorithm · Graph co
Distributed computing

1 Introduction

This work is a part of a recent project aiming
quantitative impact of *knowledge* on the *effici*
puting with distributed entities (nodes of a dis
mobile users in ad hoc networks, etc.). Indeed, as observed by Linial [18], "within the various computational models for parallel computers, the limitations that follow from the local nature of the computation are specific to the distributed context". Two frameworks have been considered for analyzing the limitations incurred by the local nature of distributed computations. One aims at identifying which tasks can or cannot be computed locally, i.e., when every node

A preliminary version of this paper appeared in the proceedings of the 34th International Colloquium on Automata, Languages and Programming (ICALP), July 2007. A part of this work was done during the stay of David Ilcinkas at the Research Chair in Distributed Computing of the Université du Québec en Outaouais, as a postdoctoral fellow.
P. Fraigniaud received additional support from the ANR project ALADDIN.
A. Pelc was supported in part by NSERC discovery grant and by the Research Chair in Distributed Computing of the Université du Québec en Outaouais.

Polylogarithmic Network Navigability Using Compact Metrics with Small Stretch

Pierre Fraigniaud* CNRS and University Paris Diderot

Cyril Gavoille† University of Bordeaux

ABSTR
Graph aug
lyzing nav
large class
graphs suc
path metr
of steps. I
graphs for
ing accord
than $\Omega(n^1$
best known
trary grap
is an augm
the shortes
ber of step
est path n
general spa
distances
ing accord
in some gr
We prov
in a huge
ficiency of
for any co
there exist
with stret
performs i
a corollary
there exist
G with str
to μ perfo
This latter
using only

*Additio
the INRIA
DYNAMO
†Additio
from the I

Permission to
personal or c
not made or
bear this noti
republish, to
permission a
SPAA'08, Jun
Copyright 20

Universal Augmentation Schemes for Network Navigability: Overcoming the $\sqrt{n}$-Barrier*

Pierre Fraigniaud CNRS and University Paris 7 France Pierre.Fraigniaud@liafa.jussieu.fr

Cyril Gavoille University of Bordeaux France gavoille@labri.fr

Adrian Kosowski Gdańsk Univ. of Technology Poland kosowski@sphere.pl

Emmanuelle Lebhar

Zvi Lotker

ABSTRACT
Augmented graph
lyzing [illegible] six de
observed experim
gram in the 60's.
(G, φ) where G is
bility distribution
is given an extra
some node v, call
of this link is cho
augmented graph
process in which
its neighbors (inc
is closest to the t
in the underlying
augmented graph
networks, while g
ing procedure app

Our objective is
schemes, i.e., aug
G a collection of p
routing in (G, φ) i
φ_{unif} is a univers

*This work was pa
at University Paris
"DYNAMO", and
at University of B
tion 295 "DYNAM
Emmanuelle Lebha
Pair". Pierre Frai
ACI project "FRA
Large".

Permission to make d
personal or classroom
not made or distribute
bear this notice and th
republish, to post on s
permission and/or a fe
SPAA'07, June 9–11,
Copyright 2007 ACM

2nd SIROCCO-SPAA Best Paper Award

Theoretical Computer Science 410 (2009) 1970–1981

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

ELSEVIER

Universal augmentation schemes for network navigability☆

Pierre Fraigniaud [a,*], Cyril Gavoille [b], Adrian Kosowski [c], Emmanuelle Lebhar [a], Zvi Lotker [d]

[a] CNRS and University Paris Diderot, 75205 Paris, France
[b] University of Bordeaux, 33405 Talence, France
[c] Gdansk University of Technology, 80-952 Gdańsk Wrzeszcz, Poland
[d] Ben Gurion University, Beer-Sheva, Israel

ARTICLE INFO

Article history:
Received 5 October 2007
Received in revised form 8 October 2008
Accepted 8 December 2008
Communicated by D. Peleg

Keywords:
Small world phenomenon
Informative labeling schemes
Routing

ABSTRACT

Augmented graphs were introduced for the purpose of analyzing the "six degrees of separation between individuals" observed experimentally by the sociologist Standley Milgram in the 60's. We define an augmented graph as a pair (G, M) where G is an n-node graph with nodes labeled in $\{1, \ldots, n\}$, and M is an $n \times n$ stochastic matrix. Every node $u \in V(G)$ is given an extra link, called a long range link, pointing to some node v, called the long range contact of u. The head v of this link is chosen at random by $\Pr[u \to v] = M_{u,v}$. In augmented graphs, greedy routing is the oblivious routing process in which every intermediate node chooses from among all its neighbors (including its long range contact) the one that is closest to the target according to the distance measured in the underlying graph G, and forwards to it. The best augmentation scheme known so far ensures that, for any n-node graph G, greedy routing performs in $O(\sqrt{n})$ expected number of steps.

Our main result is the design of an augmentation scheme that overcomes the $O(\sqrt{n})$ barrier. Precisely, we prove that for any n-node graph G whose nodes are arbitrarily labeled in $\{1, \ldots, n\}$, there exists a stochastic matrix M such that greedy routing in (G, M) performs in $\tilde{O}(n^{1/3})$, where the $\tilde{O}$ notation ignores the polylogarithmic factors.

We prove additional results when the stochastic matrix M is universal to all graphs. In

Pierre Fraigniaud, Cyril Gavoille, and Christophe Paul. Eclecticism shrinks even small worlds. *Distributed Computing*, 18(4):279-291, March 2006. Appears also in PODC '04.

Distrib. Comput. (2006) 18(4): 279–291
DOI 10.1007/s00446-005-0137-4

SPECIAL ISSUE PODC 04

Pierre Fraigniaud · Cyril Gavoille · Christophe Paul

Eclecticism shrinks even small worlds

Submitted: 28 October 2004 / Accepted: 4 July 2005 / Published online: 3 February 2006

Abstract We consider small world graphs as defined by Kleinberg (2000), i.e., graphs obtained from a d-dimensional mesh by adding links chosen at random according to the d-harmonic distribution. In these graphs, greedy routing performs in $O(\log^2 n)$ expected number of steps. We introduce *indirect-greedy* routing. We show that giving $O(\log^2 n)$ bits of topological awareness per node enables indirect-greedy routing to perform in $O(\log^{1+1/d} n)$ expected number of steps in d-dimensional augmented meshes. We also show that, independently of the amount of topological awareness given to the nodes, indirect-greedy routing performs in $\Omega(\log^{1+1/d} n)$ expected number of steps. In particular, augmenting the topological awareness above this optimum of $O(\log^2 n)$ bits would drastically decrease the performance of indirect-greedy routing.

Our model demonstrates that the efficiency of indirect-greedy routing is sensitive to the "world's dimension," in the sense that high dimensional worlds enjoy faster greedy routing than low dimensional ones. This could not be observed in Kleinberg's routing. In addition to bringing new light to Milgram's experiment, our protocol presents several desirable properties. In particular, it is totally *oblivious*, i.e., there is no header modification along the path from the source to the target, and the routing decision depends only on the target, and on information stored locally at each node.

A preliminary version of this paper appeared in the proceedings of the 23rd ACM Symposium on Principles of Distributed Computing (PODC), St. Johns, Newfoundland, Canada, July 25–28, 2004.

P. Fraigniaud (✉)
CNRS, LRI, Paris-Sud
E-mail: pierre@lri.fr

C. Gavoille
University of Bordeaux, LaBRI, Bordeaux
E-mail: gavoille@labri.fr

C. Paul
CNRS, LIRMM, Montpellier
E-mail: paul@lirmm.fr

Keywords Small world graphs · Routing · Milgram's experiment

1 Introduction

We consider small world graphs as defined by Kleinberg [7], i.e., graphs obtained from a d-dimensional mesh, for some fixed $d \geq 1$, by adding *long-range* links chosen at random according to the d-harmonic distribution, i.e., the probability that x chooses y as long-range contact is $h(x, y) = 1/(Z_x \cdot \text{dist}(x, y)^d)$ where dist() is the Manhattan distance in the mesh (i.e., the distance in the L_1 metric), and Z_x is a normalizing coefficient (cf. Sects. 5.1 and 5.2 for more details). This model aims at giving formal support to the "six degrees of separation" between individuals experienced by Milgram [14], and recently reproduced by Dodds, Muhamad, and Watts [5] (see also [1]). In a social context, professional as well as leisure occupation, citizenship, geography, ethnicity, and religiousness are all intrinsic dimensions of the human multi-dimensional world, playing different roles with possibly different impact degrees [6]. Each of these dimensions should be used as an independent criterion for routing in the social graph. In this context, one would thus expect that the more criteria used the more efficient the routing should be. Surprisingly however, Kleinberg's model does not reflect this fact, in the sense that greedy routing has the same performance whether the number of mesh dimensions considered is one, two, or more. Indeed, Kleinberg has shown that greedy routing in the n-node d-dimensional mesh augmented with long-range links chosen according to the d-harmonic distribution performs in $O(\log^2 n)$ expected number of steps, i.e., independently of d. (This bound is tight as it was shown in [3] that greedy routing performs in at least $\Omega(\log^2 n)$ expected number of steps, independently of d). Kleinberg has also shown that augmenting the d dimensional mesh with the r-harmonic distribution, $r \neq d$, results in poor performance, i.e., $\Omega(n^{\alpha_r})$ expected number of steps for some positive constant α_r. Furthermore, it is shown in [2] that, in the 1-dimensional mesh augmented

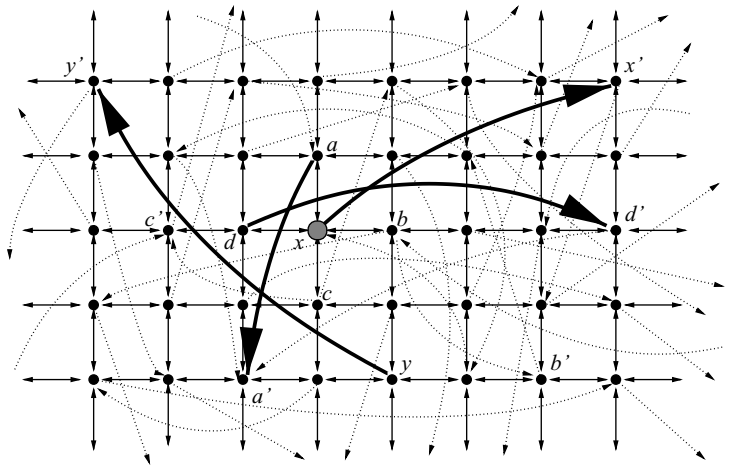


Fig. 1 Long-range links in the 2-dimensional mesh. The topological awareness of node x is composed of the four plain long-range links

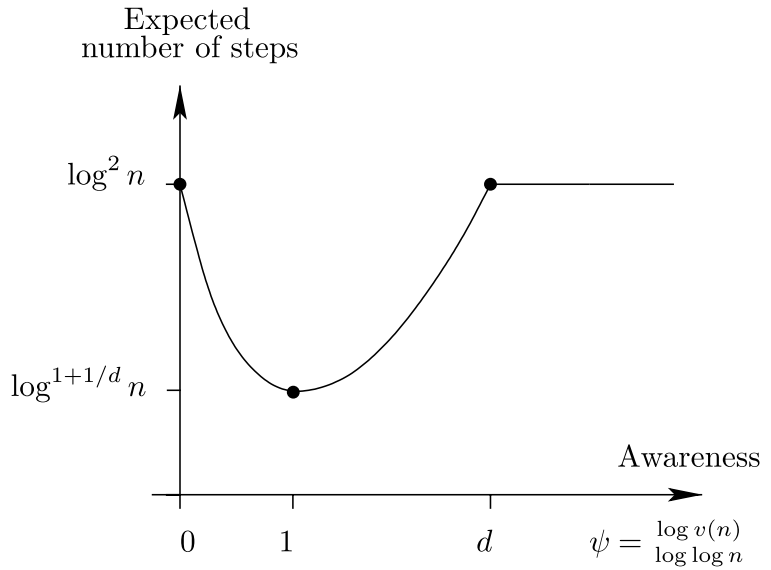


Fig. 4 The expected number of steps vs. the awareness $v(n) = (\log n)^\alpha$. The expected number of steps is $\Omega((\log n)^{2+\alpha/d-\alpha})$ if $\alpha < 1$ (by Lemma 5), and $\Omega((\log n)^{1+\alpha/d-o(1)})$ if $1 \leq \alpha < d$ (by Lemma 6). For $\alpha > d$, the expected number of steps is $\Theta(\log^2 n)$ (by Lemma 6)

The End!

2008 – Parc Algonquin (CAN)

Bonus