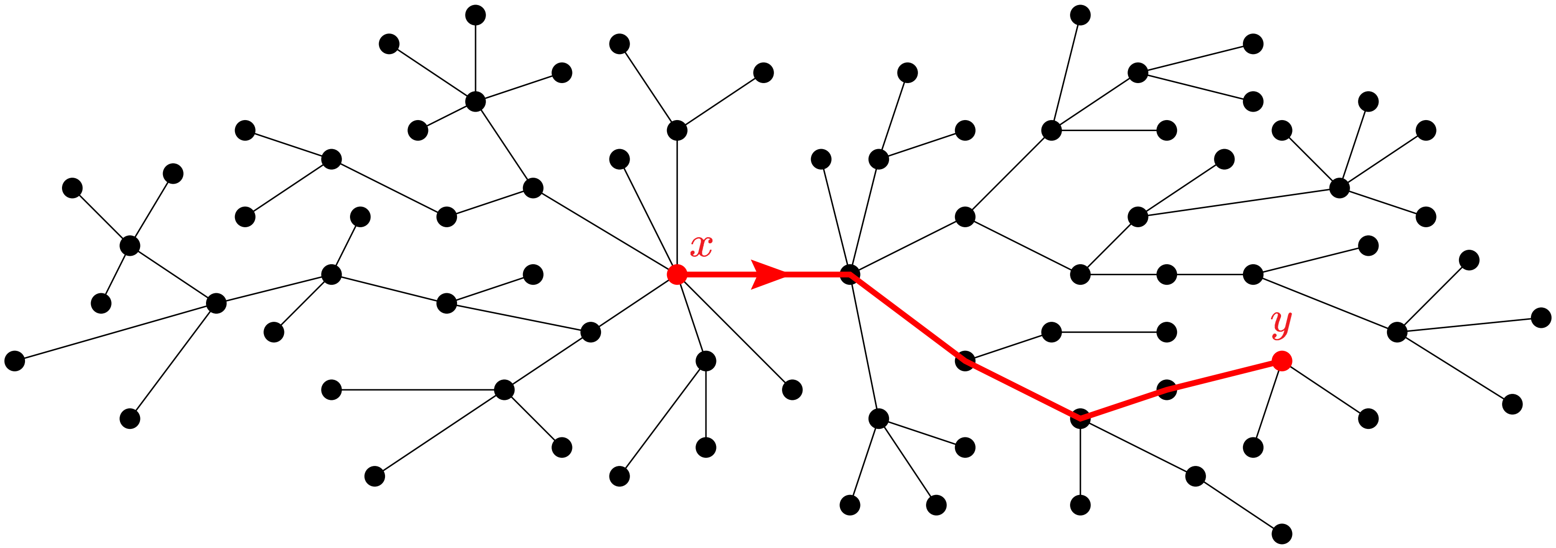
Routing in Trees

Pierre Fraigniaud - LRI, CNRS Orsay
Cyril Gavoille - Bordeaux University

ICALP - July 2001

Problem

To route a message from x to y in an n-node tree, $\forall\ x, y$



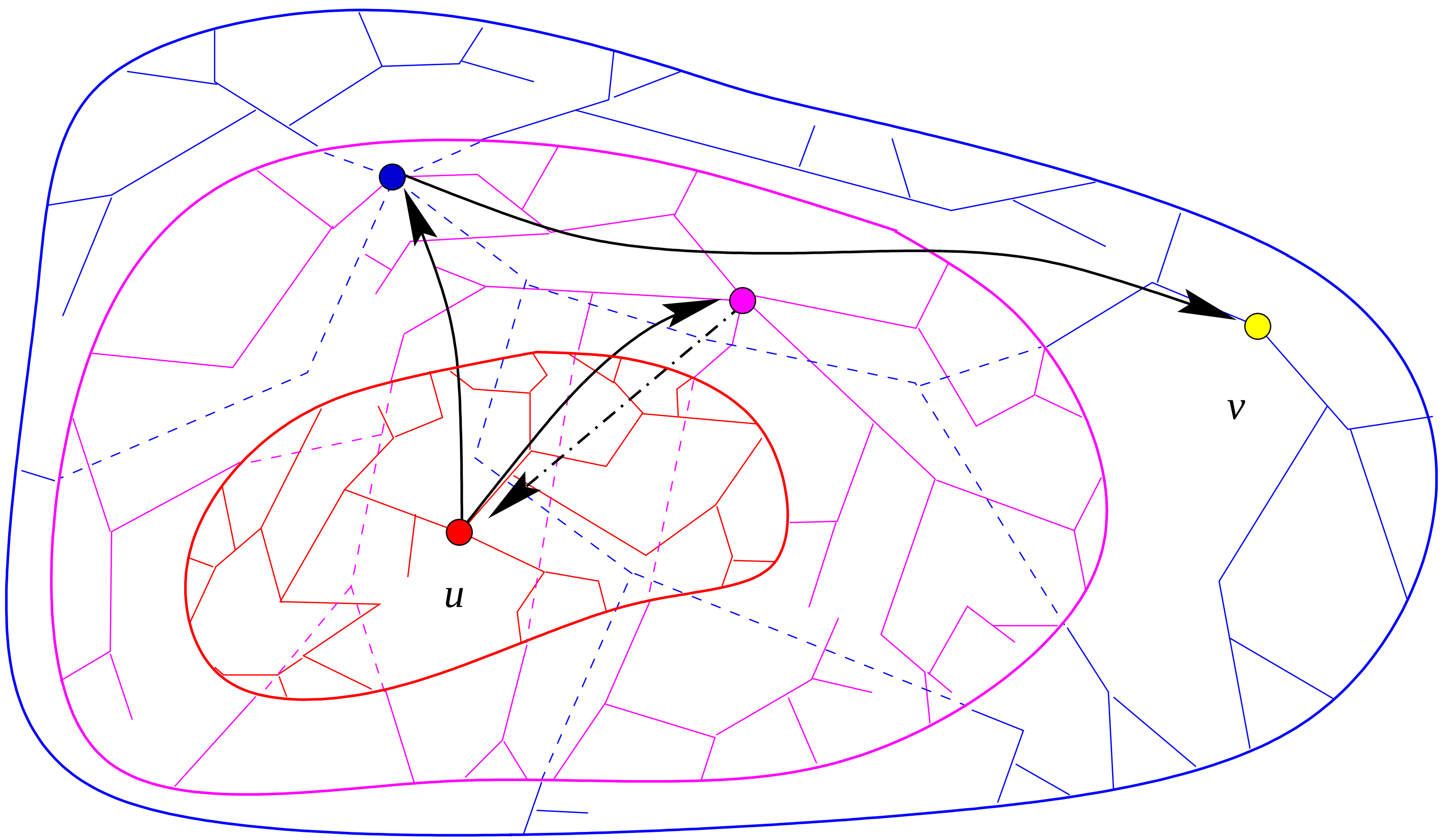
Constraints :

- shortest paths;
- <u>local</u> and <u>compact</u> routing tables;
- <u>fast</u> routing decision.

A Tool for Universal Routing Schemes

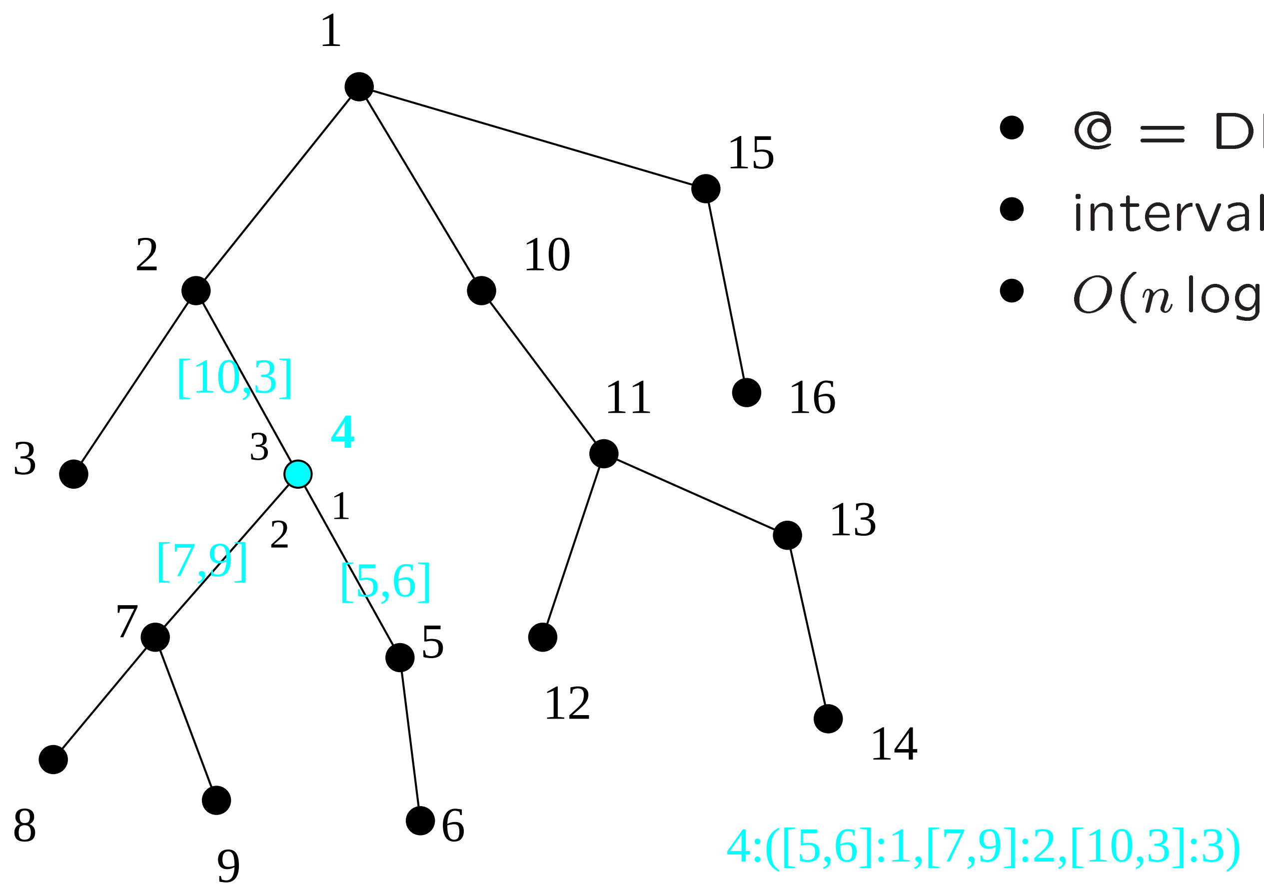
Hierarchical routing schemes applicable to <u>all</u> the networks

[Awerbuch,Cowen,Eilam,Frederickson,Peleg,Upfal,...]



Standard Techniques

[Santoro-Khatib/van Leeuwen-Tan]



- @ = DFS numbering
- interval routing based
- $O(n \log n)$ bits in total!

4:([5,6]:1,[7,9]:2,[10,3]:3)

Binary search $\Rightarrow O(\log n)$ routing time

Memory Space $\Rightarrow O(d \log n)$ bits, $d =$ degree

Problem: Space $\Omega(n)$ bits whenever $d \geqslant n/\log n$

Another solution: 4:

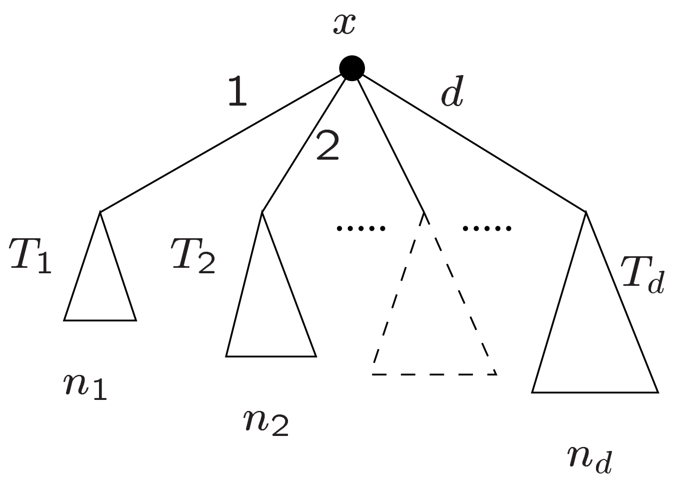
								y							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
				#1		#2			#3						

Routing algorithm (local): compute the number of 1's (mod d) before position y.

$\Rightarrow n$ bit space, but $\Omega(n)$ time! (In fact, this problem can be solved with $n + o(n)$ bits and constant time on a $O(\log n)$ bit word-RAM machine.)

Can we do better than n bits?

Lower bound: $c\sqrt{n}$ bits at least, $c \approx 3.7$



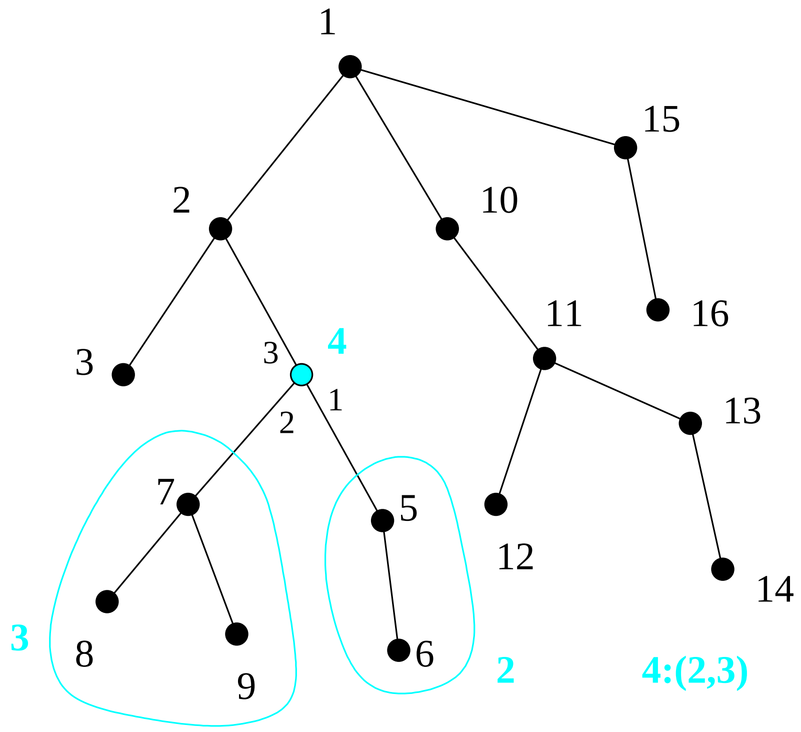
- test in x its local routing table towards all the possible destinations $y \in \{1, \ldots, n\} \setminus \{x\}$
- sort answers by frequency $n_i = |T_i|$, $n_1 \leqslant \cdots \leqslant n_d$

$\sum_{i=1}^{d} n_i = n - 1$ and $1 \leqslant n_1 \leqslant \cdots \leqslant n_d$.

x "computes" $(n_1, \ldots, n_d)$ knowing n and x.

The number of partitions of n is $\sim 2^{c\sqrt{n}}$ $\Rightarrow$ $c\sqrt{n}$ bits for x. $\square$

Can we do better than n bits? (cond't): YES!



- @ = DFS numbering w.r.t. the number of descendents
- port #1 = lightest child
  port #2= 2nd lightest child
  ...

$4:(2,3) \Rightarrow$ description with $O(\sqrt{n})$ bits (partition enumeration).

$y \in]4, 4+2]$? $y \in]6, 6+3]$? otherwise route on port 3.

Problem: Routing time? What happens if addresses $\notin \{1, \ldots, n\}$ and coded on more than $\log_2 n$ bits? #ports numbering?

Results (for n-node trees)

Port permut.?	Address size	Local Table size	Routing time	Preprocessing time
yes	$\log n$	$\geqslant 3.7\sqrt{n}$		
yes	$\log n$	$\leqslant 3.7\sqrt{n}$	$2^{O(\sqrt{n})}$	$2^{O(\sqrt{n})}$
no	$\log n$	$\geqslant n - o(n)$		
no	$\log n$	$\leqslant n + o(n)$	$O(1)$	$O(n^2)$
no	$O\left(\frac{\log^2 n}{\log\log n}\right)$	$O\left(\frac{\log^2 n}{\log\log n}\right)$	$O(1)$	$O(n\log n)$
yes	$\leqslant 5\log n$	$\leqslant 3\log n$	$O(1)$	$O(n\log n)$

Principle

(some notations)

r = root of the tree

weight(x) = number of descendents of x

id(x) = identifier for x using a DFS w.r.t. the heaviest children

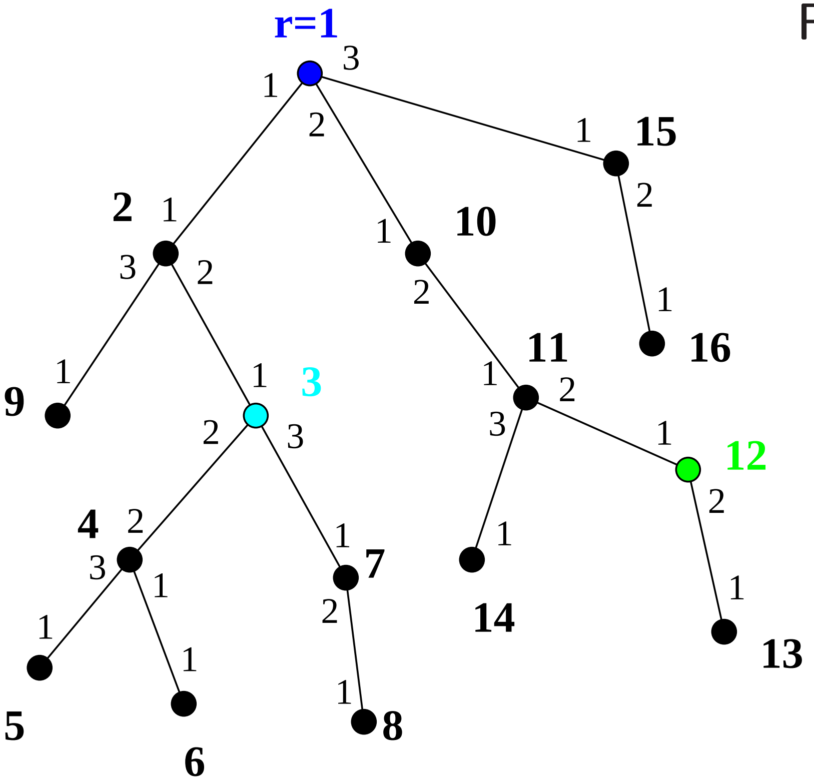
port$(x, \text{child}(x)) = p + 1$ for the p-th heaviest child of x (p=rank)

ℓpath(x) = list of large ports (rank$\geqslant$ 2) on the path from r to x

$$\begin{aligned} @(x) &= \langle \mathrm{id}(x), \ell\mathrm{path}(x) \rangle \\ \mathrm{table}(x) &= \langle \mathrm{id}(x), \mathrm{weight}(x), \mathrm{weight}(\mathrm{child}_1(x)), |\ell\mathrm{path}(x)| \rangle \end{aligned}$$

Routing Algorithm

$$\text{@}(x) \;=\; \langle \mathsf{id}(x), \ell\mathsf{path}(x)\rangle$$
$$\mathsf{table}(x) \;=\; \langle \mathsf{id}(x), \mathsf{weight}(x), \mathsf{weight}(\mathsf{child}_1(x)), |\ell\mathsf{path}(x)|\rangle$$



Routing algorithm from x to y:

- 0 (=stop) if $\mathsf{id}(y) = \mathsf{id}(x)$
- 1 if $\mathsf{id}(y) \notin [\mathsf{id}(x), \mathsf{id}(x) + \mathsf{weight}(x)[$
- 2 if $\mathsf{id}(y) \in [\mathsf{id}(x), \mathsf{id}(x) + \mathsf{weight}(\mathsf{child}_1(x))[$
- $p + 1$ otherwise, p-th item of $\ell\mathsf{path}(y)$

<12,2,1,1> $\ell\mathsf{path}(12) = (2, \cancel{2}, \cancel{2})$
<3,6,3,0> $\ell\mathsf{path}(3) = (\cancel{1}, \cancel{2})$

Bit Counting

$\text{rank}(x, y) = \text{rank of the weight of the child } y \text{ of } x$
$(\approx \text{port}(x, y) - 1 : 1, 2, \ldots)$

Observation:

- If y child x, $\text{rank}(x, y) \leqslant \text{weight}(x)/\text{weight}(y)$

$\Rightarrow$ If $\ell\text{path}(x) = (p_1, \ldots, p_k)$, $\prod p_i \leqslant n$ and $k \leqslant \log n$

Coding of ℓpath(x)

Ex: $\ell\text{path}(x) = (3, 9, 2, 5, 2, 4, 6) \longrightarrow (A, B)$

ℓpath(x)	=	3	9	2	5	2	4	6
binary writing: A	=	11	1001	10	101	10	100	110
field separator: B	=	10	1000	10	100	10	100	100

$$|B| = |A| \quad \text{and} \quad |A| \leqslant \sum_{i=1}^{k} \lceil \log p_i \rceil \leqslant \log \left(\prod p_i \right) + k \leqslant 2 \log n$$

$\Rightarrow$ $\ell\text{path}(x) = (A, B)$ on $4 \log n$ bits

$\Rightarrow$ @(x) on $5 \log n$ bits

$\Rightarrow$ table(x) on $3 \log n + O(\log \log n)$ bits

Extracting the p-th item

$\text{select}_B(t)$ = position of the t-th "1" in B.
If B is fixed, $\text{select}_B(\cdot)$ can be implemented with $|B| + o(|B|)$ bit space and constant time (word-RAM model)
[Munro-Raman FOCS'97]

Ex: $\ell\text{path}(x) = (3, 9, 2, \underline{5}, 2, 4, 6)$, $p = 4$.

$A = \overline{11100110}101\overline{10100110}$
$B = 10100010\underline{1}00\underline{1}0100100$

$a := \text{select}_B(p)$,
$b := \text{select}_B(p + 1)$,
$s := (A \ll (a - 1)) \gg (|A| + a - b) =$ "101" $\to 5$.

Improving the Constants ...

Lemma (Kalmár 1930). Let $Z(n)$ be the number of integer sequences $p_1, p_2, \ldots$ such that $p_i \geqslant 2$ and $\prod p_i \leqslant n$. Then,

$$Z(n) \sim n^{1.7286\ldots} \quad (\text{as } n \to +\infty)$$

(where 1.7286... is the real root of $\zeta(x) = 2$, $\zeta(x) = \sum_{i \geqslant 1} i^{-x}$).

Thus, @(x) on $2.7286 \log n$ bits and table(x) on $3 \log n$ bits.

Independently, [Thorup-Zwick, SPAA'01]:
@(x) on $(1 + o(1)) \log n$ bits and
table(x) on $(1 + o(1)) \log n$ bits.