

Localized Data Structures

Cyril Gavaille


(LaBRI, University of Bordeaux)

LOCALITY 2007

Portland, Oregon



Contents

1. Efficient data structures
 2. Localized data structures
 3. Informative labeling schemes
 4. Conclusion
- 

1. Efficient data structures (Tarjan's like)

Example 1:

A tree (static) T with n vertices

Question: nearest common ancestor
 $nca(x,y)$ for some vertices x,y ?

Note: queries (x,y) are not known in
advance

(on-line queries on a static tree)

[Harel-Tarjan '84]

Each tree with n vertices has a data structure of $O(n)$ space (computable in linear time) such that nca queries can be answered in constant time.

Example 2:

A weighted graph G with n vertices,
and a parameter $k \geq 1$

Question: a k -approximation $\delta(x,y)$ on
 $\text{dist}(x,y)$ in G for some vertices x,y ?

with $\text{dist}(x,y) \leq \delta(x,y) \leq k \cdot \text{dist}(x,y)$

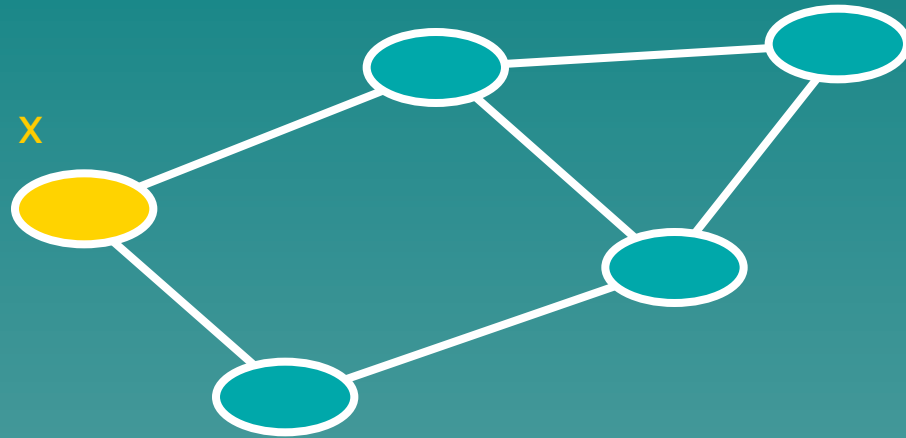
[Thorup-Zwick - J.ACM '05]

Each undirected weighted graph G with n vertices, and each integer $k \geq 1$, has a data structure of $O(k \cdot n^{1+1/k})$ space (computable in $O(km \cdot n^{1/k})$ expected time) such that $(2k-1)$ -approximated distance queries can be answered in $O(k)$ time.

Essentially optimal, related to an Erdős Conjecture.

2. Localized data structures

A network

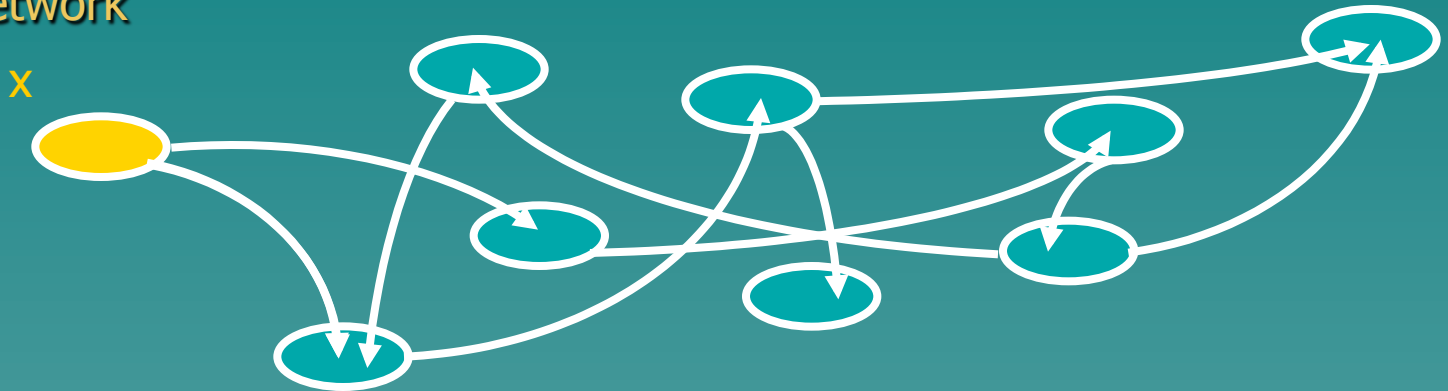


Typical questions are:

Answer to query Q with the local knowledge of x (or its vicinity), so without any access to a global data structure.

Example 1: Distributed Hash Tables (DHT)

set of peers
logical network

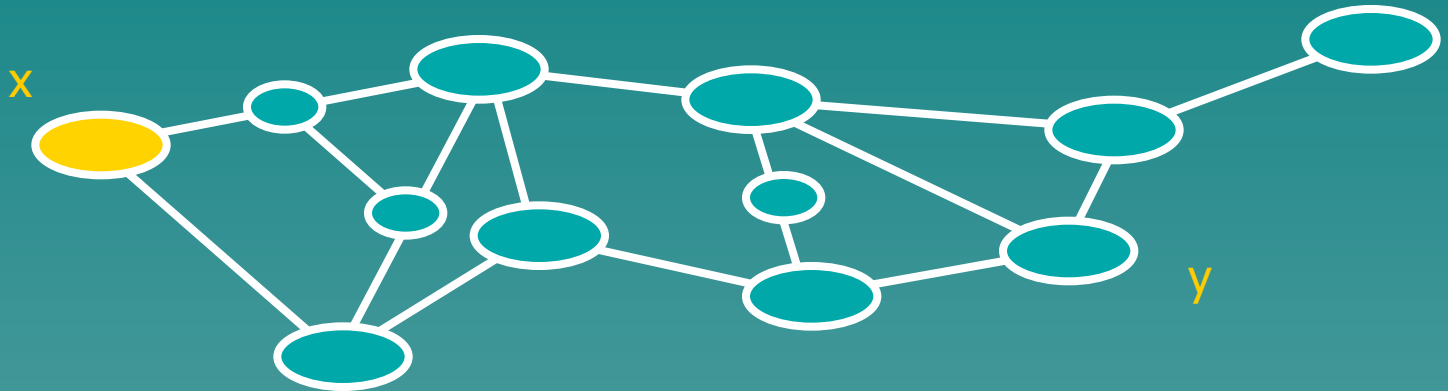


Query at x: who has any mpeg file named
“Sta*Wa*”?

Answer: go to **w** and ask it.

x does not know, but **w** certainly knows ... at least a pointer

Example 2: Routing in a physical network



Query at x : next hop to go to y ?

Example 3: in a dynamic setting

A growing rooted tree

Query at x : the number of descents of x
(or a constant approximation of it)

[Afek, Awerbuch, Plokin, Saks – J.ACM '96]

It is possible to maintain a 2-approximation on the number of descendants with $O(\log^2 n)$ amortized messages of $O(\log \log n)$ bits each, n number of inserted vertices.

Goals are:

- ◆ The same as for global data structures:
 - Low preprocessing time
 - Small size data structure
 - Fast query time
 - Efficient updates
- + Smaller and balanced local data structures
- + Low communication cost (trade-offs), for multiple hops answers

3. Informative Labeling Schemes

For the talk

- A static network/graph
- Queries: involve only vertices
- Answers: do not require any communication (direct data structures)

Question: $\text{dist}(x,y)$ in a graph G ? (with localized data structure)

data structure
for graph G



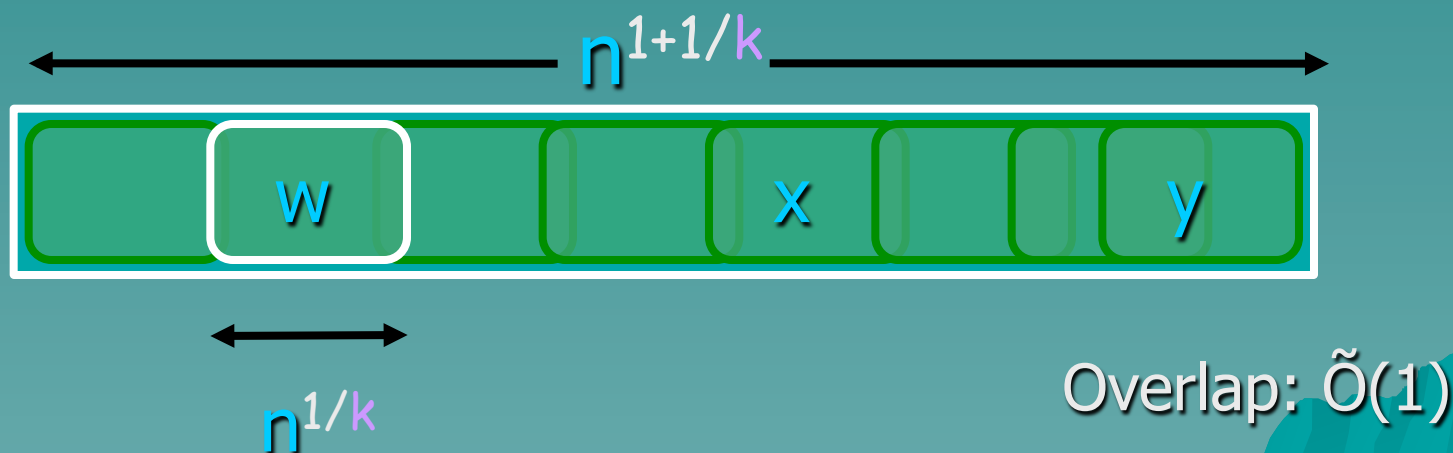
Answering to $\text{dist}(x,y)$ consists only in inspecting the local data structure of x and of y .

Main goal: minimize the maximal size of a local data structure. Wish: $|\text{DS}(x,G)| \ll |\text{DS}(G)|$, ideally

$$|\text{DS}(x,G)| \approx (1/n) \cdot |\text{DS}(G)|$$

[Thorup-Zwick - J.ACM '05]

... Moreover, each vertex $w \rightarrow L(w)$ of $\tilde{O}(n^{1/k})$ bits such that a $(2k-1)$ -approximation on $\text{dist}(x,y)$ can be answered from $L(x)$ and $L(y)$ only.



Informative Labeling Schemes (more formally) [Peleg '00]

Let P be a graph property defined on pairs of vertices (can be extended to any tuple), and let F be a graph family.

A P -labeling scheme for F is a pair $\langle L, f \rangle$ such that: $\forall G \in F, \forall u, v \in G$:

- (labeling) $L(u, G)$ is a binary string
- (decoder) $f(L(u, G), L(v, G)) = P(u, v, G)$

Some P-labeling schemes

- ◆ Adjacency
- ◆ Distance (exact or approximate)
- ◆ First edge on a (near) shortest path (compact routing, labeled-based routing)
- ◆ Ancestry, parent, nca, sibling relation in trees
- ◆ Edge/vertex connectivity, flow
- ◆ Proof labeling systems [Korman, Kutten, Peleg]
- ◆ Small-world & navigability
- ◆ Forbidden set [Courcelle, Twigg]

Ancestry in rooted trees

Motivation: [Abiteboul, Kaplan, Milo ' 01]

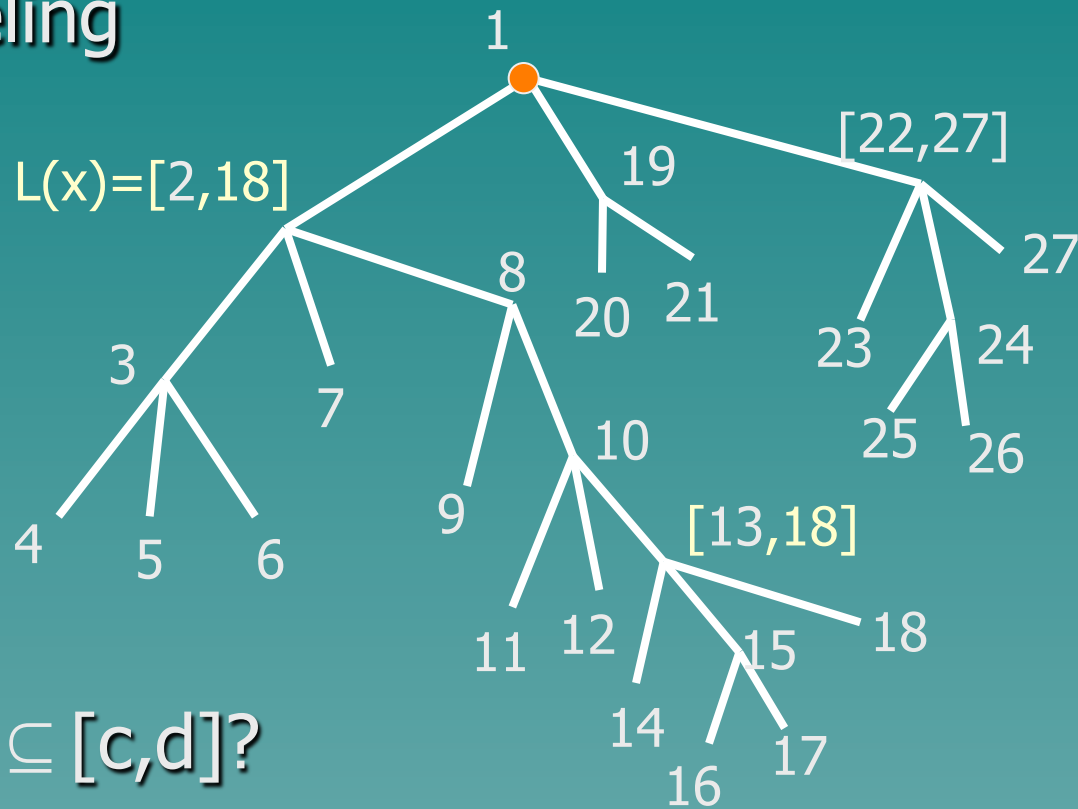
The `<TAG> ... </TAG>` structure of a huge XML data-base is a rooted tree. Some queries are ancestry relations in this tree.

Use compact index for fast query XML search engine. Here the constants do matter. Saving **1** byte on each entry of the index table is important. Here **n** is very large, $\sim 10^9$.

Ex: Is `<"distributed computing">` descendant of `<book_title>`?

Folklore? [Santoro, Khatib '85]

DFS labeling

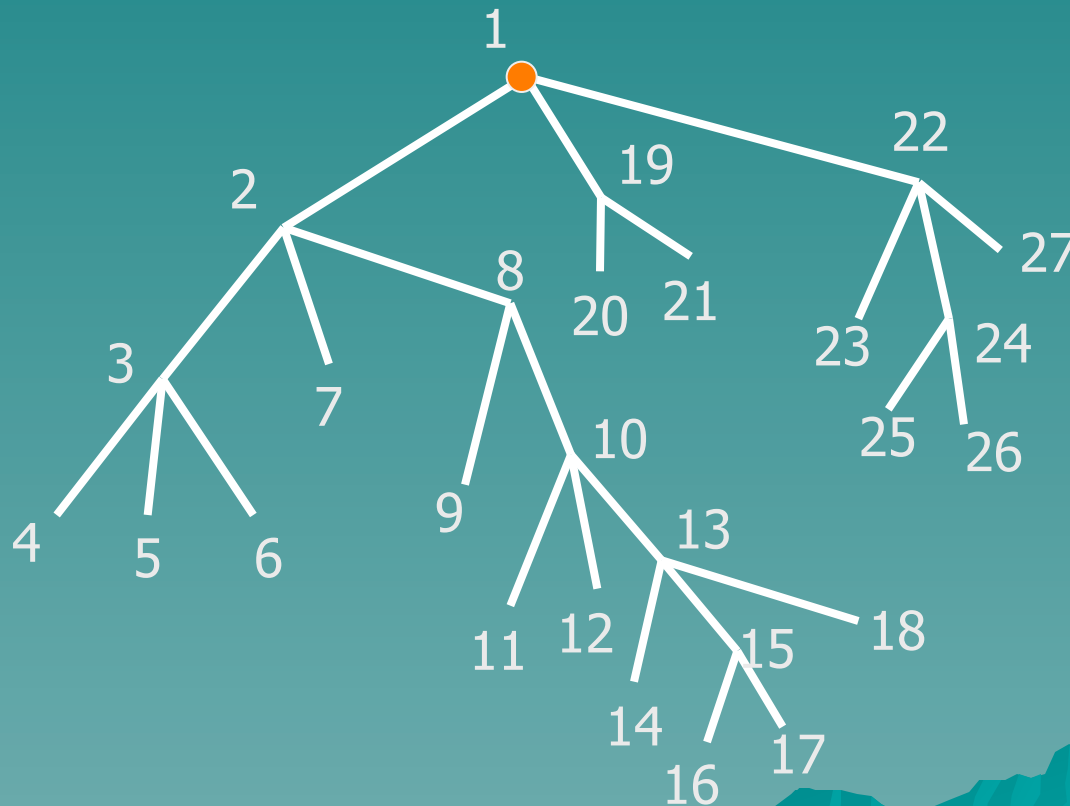


$\Rightarrow 2 \log n$ bit labels

[Alstrup, Rauhe – Siam J. Comp. '06]

Upper bound: $\log n + O(\sqrt{\log n})$ bits

Lower bound: $\log n + \Omega(\log \log n)$ bits



Adjacency Labeling / Implicit Representation

$$P(x,y,G)=1 \text{ iff } xy \text{ in } E(G)$$

[Kanan,Naor,Rudich – STOC ' 92]

$O(\log n)$ bit labels for:

- trees (and forests)
- bounded arboricity graphs (planar, ...)
- bounded treewidth graphs

In particular:

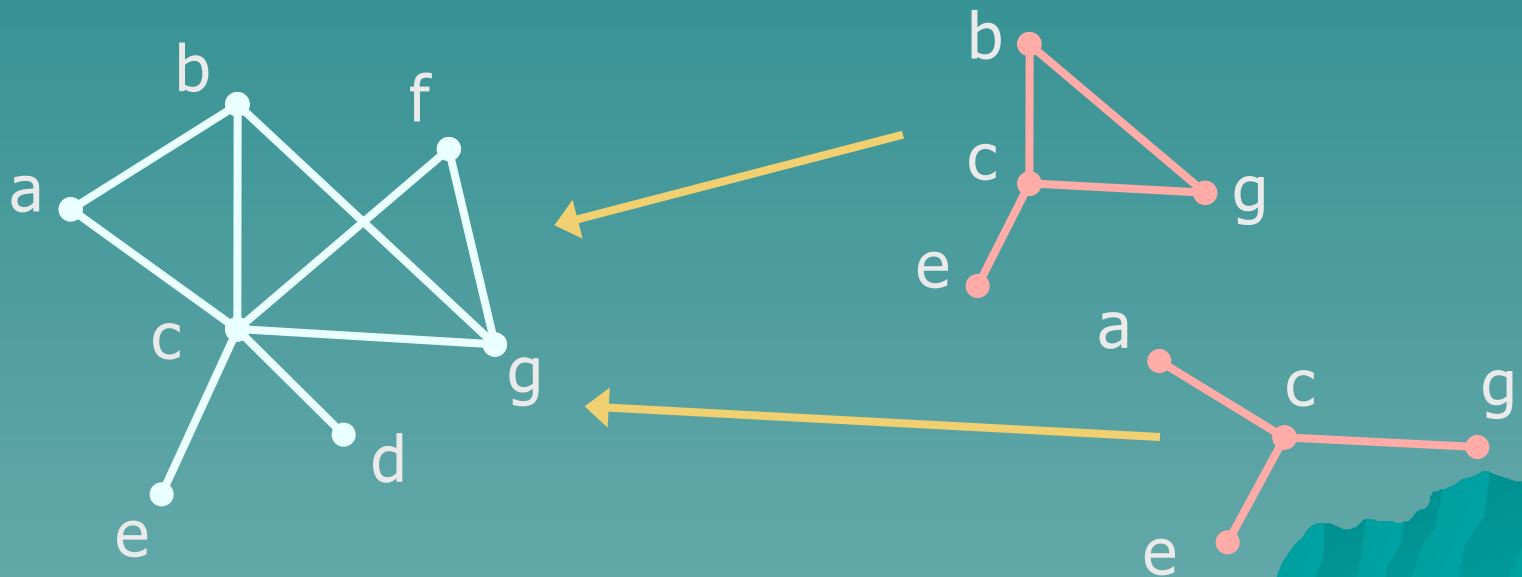
- $2\log n$ bits for trees
- $4\log n$ bits for planar

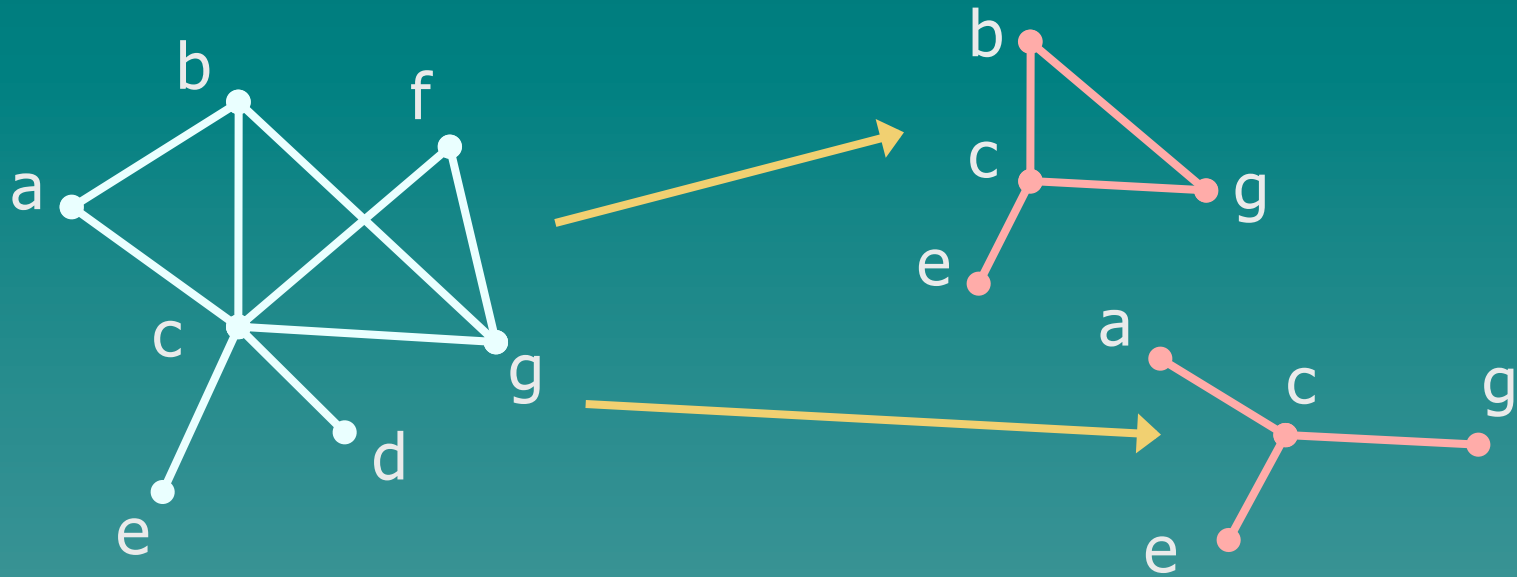
Actually, the problem is equivalent to an old combinatorial problem:

[Babai, Chung, Erdős, Graham, Spencer '82]

Small Universal Induced Graph

U is an universal graph for the family F if every graph of F is isomorphic to an induced subgraph of U





Universal graph U
(fixed for F)

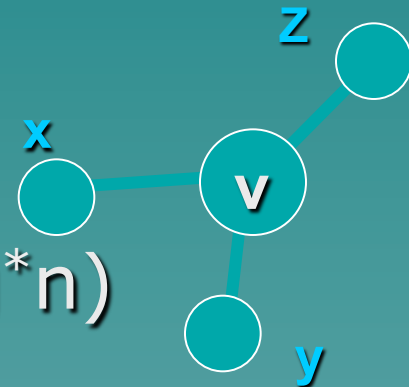
Graph G of F

$$|L(x, G)| = \lceil \log_2 |V(U)| \rceil$$

Best known results/Open questions

- ◆ Bounded degree graphs: $1.867 \log n$
[Alon,Asodi – J. Comp. App. Math '02]

- ◆ Trees: $\log n + O(\log^* n)$
[Alstrup,Rauhe - FOCS '02]
 \Rightarrow Planar: $3 \log n + O(\log^* n)$

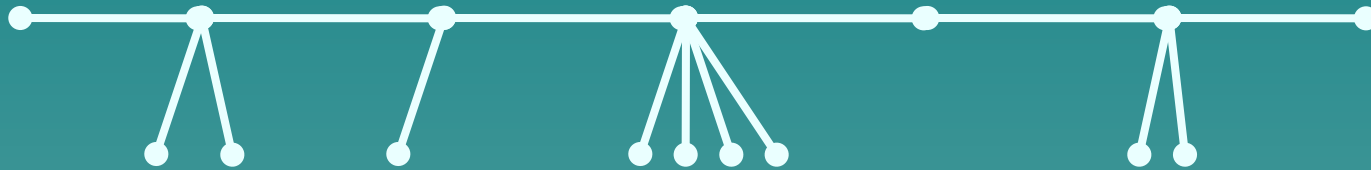


- ◆ Planar: $2 \log n + O(\log \log n)$
[Gavoille,Labourel - ESA '07]

$$\log^* n = \min\{ i \geq 0 \mid \log^{(i)} n \leq 1 \}$$

- Lower bounds?: $\log n + \Omega(1)$ for planar

$\log n + O(1)$ bits for this family?



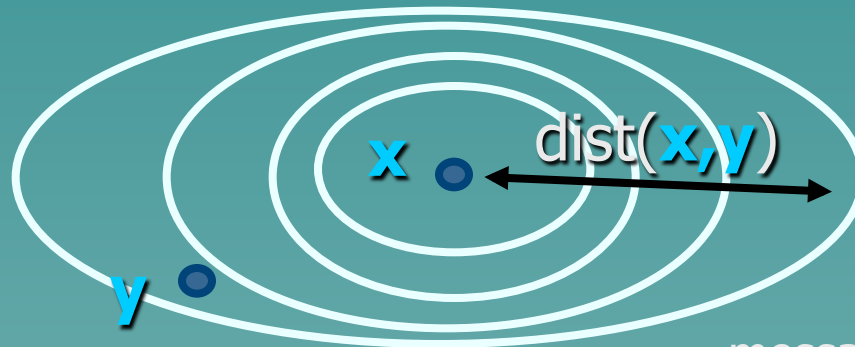
- No hereditary family with $n!2^{O(n)}$ labeled graphs (trees, planar, bounded genus, bounded treewidth,...) is known to require labels of $\log n + \omega(1)$ bits.

Distance

$$P(x,y,G)=\text{dist}(x,y) \text{ in } G$$

Motivation: [Peleg '99]

If a short label (say of polylogarithmic size) can be added to the address of the destination, then routing to any destination can be done without routing tables and with a “limited” number of messages.



message header=hop-count

A selection results

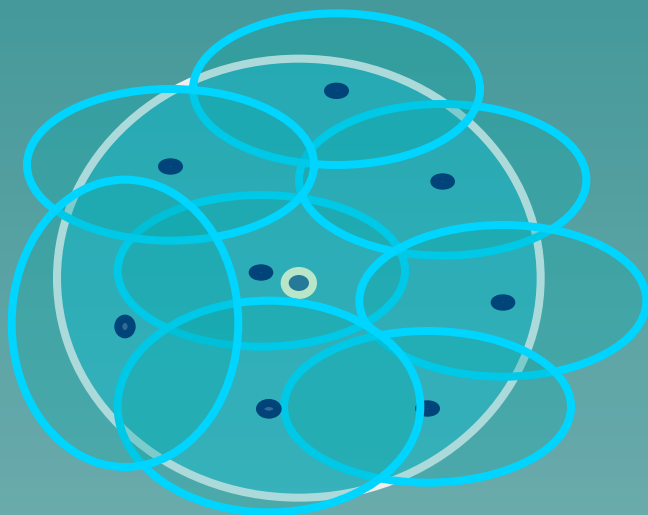
- ◆ $\Theta(n)$ bits for general graphs
 - 1.56n bits, but with $O(n)$ time decoder!
[Winkler '83 (Squashed Cube Conjecture)]
 - 11n bits and $O(\log \log n)$ time decoder
[Gavoille, Peleg, Pérennès, Raz '01]
- ◆ $\Theta(\log^2 n)$ bits for trees and bounded treewidth graphs, ... [Peleg '99, GPPR '01]
- ◆ $\Theta(\log n)$ bits and $O(1)$ time decoder for interval, permutation graphs, ... [ESA '03]:
 $\Rightarrow O(n)$ space $O(1)$ time data structure, even for $m = \Omega(n^2)$

Results (cont' d)

- ◆ $\Theta(\log n \cdot \log \log n)$ bits and $(1+o(1))$ -approximation for trees and bounded treewidth graphs

[GKKPP – ESA '01]

- ◆ More recently: doubling dimension- α graphs
Every radius- $2r$ ball can be covered by $\leq 2^\alpha$ radius- r balls



- Euclidean graphs have $\alpha=O(1)$
- Include bounded growing graphs
- Robust notion

Distance labeling for doubling dimension- α graphs

$O(\varepsilon^{-O(\alpha)} \log n \cdot \log \log n)$ bits

$(1+\varepsilon)$ -approximation for doubling
dimension- α graphs

[Gupta, Krauthgamer, Lee – FOCS
'03]

[Talwar – STOC '04]

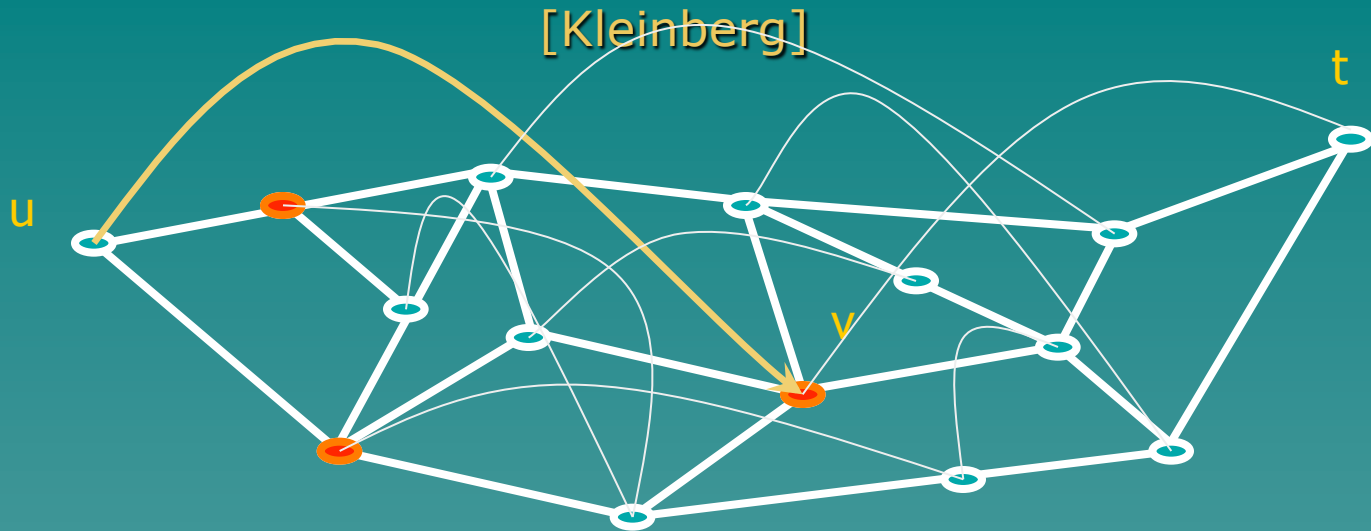
[Mendel, Har-Peled – SoCG '05]

[Slivkins – PODC '05]

Distance labeling for planar

- ◆ $O(\log^2 n)$ bits for 3-approximation
[Gupta, Kumar, Rastogi – *Siam J. Comp* '05]
- ◆ $O(\varepsilon^{-1} \log^2 n)$ bits for $(1+\varepsilon)$ -approximation
[Thorup – *J.ACM* '04]
- ◆ $\Omega(n^{1/3}) \leq ? \leq \tilde{O}(\sqrt{n})$ for exact distance
- ◆ $O(\varepsilon^{-1} \log^2 n)$ bits for $(1+\varepsilon)$ -approximation for graphs excluding a fixed minor (K_5, K_6, \dots)
[Abraham, Gavoille – *PODC* '06]

Small-World & Navigability



- ◆ Augmented graph: (G, P) [base graph, distributions]

$$P(u, v) = \Pr(u \text{ has } v \text{ as long range contact})$$

- ◆ Greedy Routing: closest neighbor (in G)
- ◆ Expected number of hops (according to P)

Small-World & Navigability



- ◆ Uniform distribution: $P(u,v) = 1/n$
- ◆ r -harmonic distribution: $P(u,v) \propto 1/d(u,v)^r$

Note: for harmonic need to know distances in G in order to compute P

Small-World & Navigability

- ◆ How much complicated is P?
- ◆ Design a “*simple*” distribution with a low expected number of hops for G

k -navigability labeling scheme

is a labeling scheme $\langle L, f \rangle$ st: $\forall u, v$ of G

- $P(u, v) = f(L(u, G), L(v, G))$
- Expected #hops $\leq k$

Navigability: Results

- ◆ All graphs [uniform]:
 $O(\sqrt{n})$ -navigability with $\lceil \log n \rceil$ -bit labels
- ◆ Grids [in DISC '05]:
 $O(\log n)$ -navigability with $\lceil \log n \rceil$ -bit labels
- ◆ Trees [Fraigniaud et al.]:
 $O(\log^2 n)$ -navigability with $O(\log^2 n)$ bits
 $O(\log^3 n)$ -navigability with $\lceil \log n \rceil$ bits

Navigability: Results (cont' d)

- ◆ Bounded *path-shape* graphs [SPAA '07]:
 $O(\text{ps}(G) \cdot \log^2 n)$ -navigability with $\lceil \log n \rceil$ bits
[bounded tree-width, bounded path-length, AT-free, permutation, interval graphs ... have $\text{ps} = O(\log n)$]
- ◆ All graphs:
 $\tilde{O}(n^{1/3})$ -navigability, but with $O(n)$ bits!
- ◆ Questions:
 - smallest $k = k(n)$ for k -navigability \forall graphs
currently: $c^{\sqrt{\log n}} \leq k \leq \tilde{O}(n^{1/3})$
 - $o(\sqrt{n})$ -navigability with $\text{polylog}(n)$ labels

Forbidden-Set Labeling

[Courcelle, Twigg '07]

◆ Problem:

Design a routing scheme for G st for every subset X of “forbidden” nodes (crashes, malicious, ...) routing tables can be updated efficiently provided X .

⇒ This capture routing policies

◆ *Extension:* Forbidden-set P-labeling

$\langle L, f \rangle$ st $\forall X, P(u, v, G \setminus X) = f(L(u, G), L(u, G), L(X, G))$

Forbidden-Set Labeling: Results

- ◆ Example:

Connectivity in trees: $P(u, v, T \setminus X) = \text{TRUE}$ iff u and v are in the same connected component of $T \setminus X$.
 \Rightarrow can be done with $O(\log n)$ -bit labels.

- ◆ [Courcelle, Twigg – STACS '07]

If G has bounded “*clique-width*” (generalization of tree-width) and every monadic second order predicate P (distances, connectivity, ...) then labels of $O(\log^2 n)$ -bit suffice.

- ◆ *Note:* same (optimal) bounds for distances in trees as the static case, but do not include planar ...

Conclusion

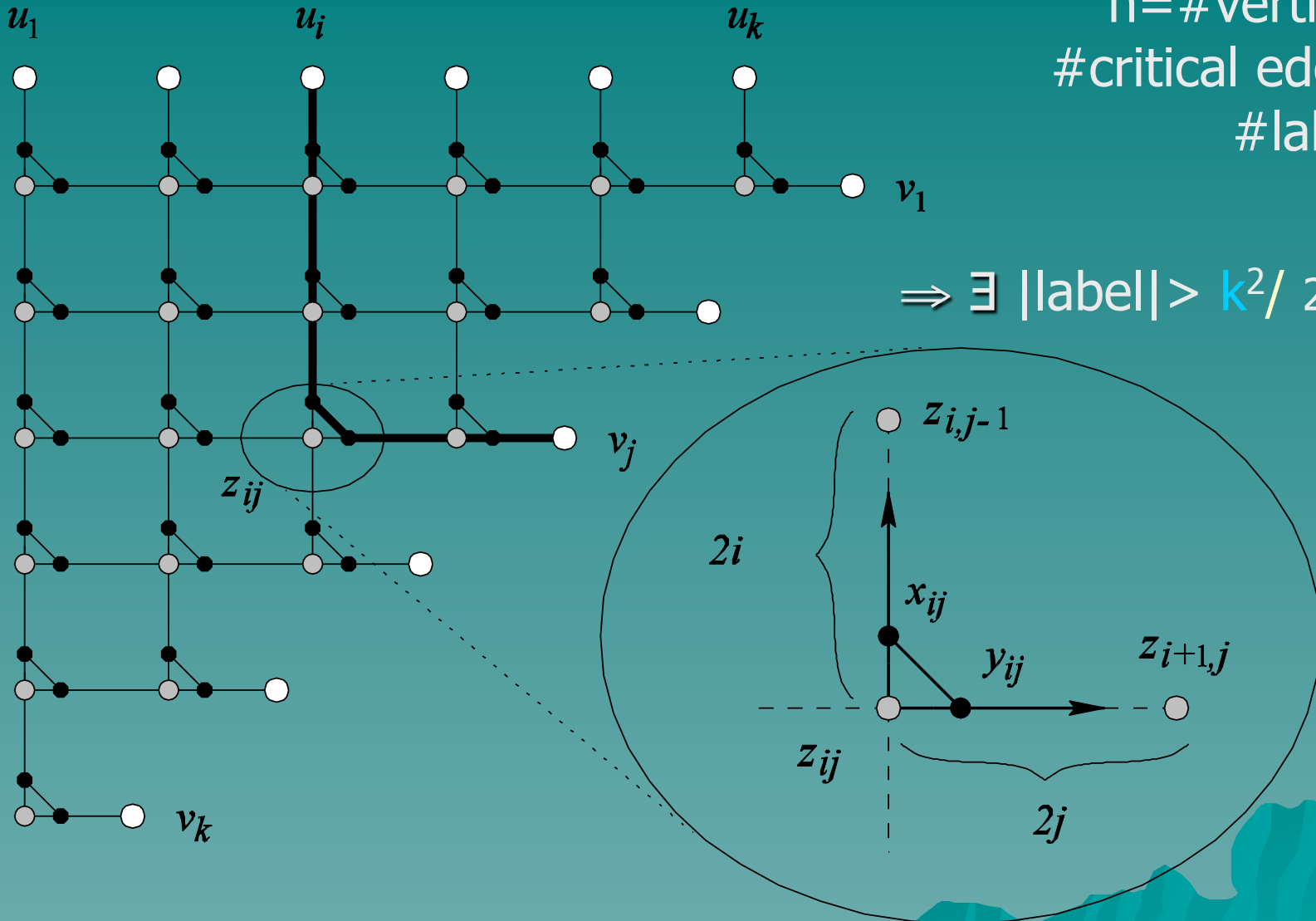
- ◆ Labeling scheme for *distributed computing* is a rich concept.
- ◆ Many things remain to do, specially lower bounds

Lower bounds for planar

[Gavoille, Peleg, Pérennès, Raz – SODA '01]

$n = \# \text{vertices} \sim k^3$
 $\# \text{critical edges} \sim k^2$
 $\# \text{labels} = 2k$

$\Rightarrow \exists \text{label} > k^2 / 2k \sim n^{1/3}$

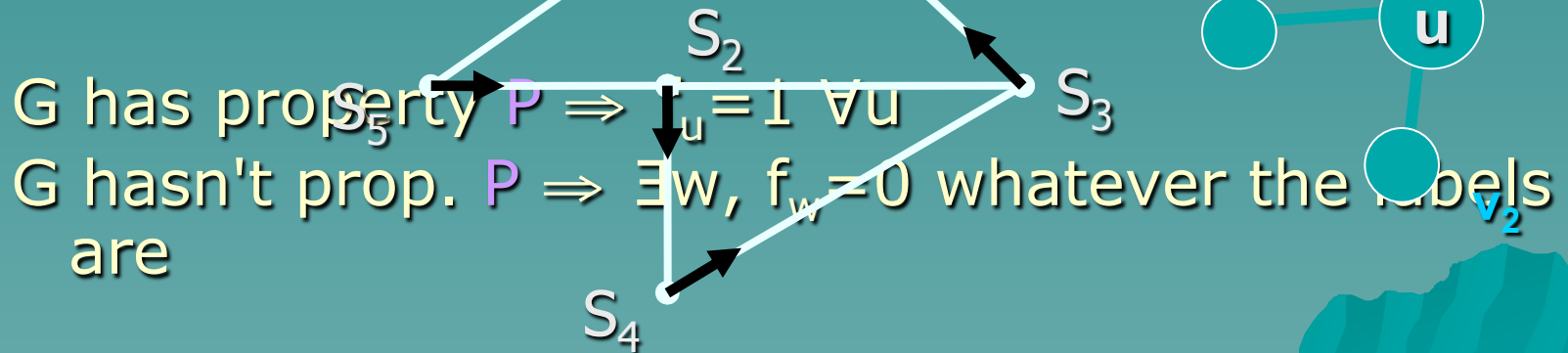


Proof Labeling Systems

[Korman, Kutten, Peleg – PODC '05]

- ◆ A graph G with a state S_u at each vertex u : (G, S)
- ◆ A global property P (MST, 3-coloring, ...)
- ◆ A marker algorithm applied on (G, S) that returns a label $L(u)$ for u

- ◆ A binary decoder (checker) for u applied on $N(u)$:
 $f_u = f(S_u, L(u), L(v_1), \dots, L(v_k)) \in \{0, 1\}$



What is the knowledge needed for local verifications of global properties?

