An Overview on Compact Routing

Cyril Gavoille[1]

[1]University of Bordeaux, France

2-6 October 2006
2nd Research Workshop on Flexible Network Design
University of Bologna Residential Center Bertinoro (Forlì), Italy

The Compact Routing Problem

Input: a network G (a weighted connected graph)

Ouput: a **routing scheme** for G

A *routing scheme* is a distributed algorithm that allows **any** source node to route messages to **any** destination node, given the destination's network identifier

The Compact Routing Problem

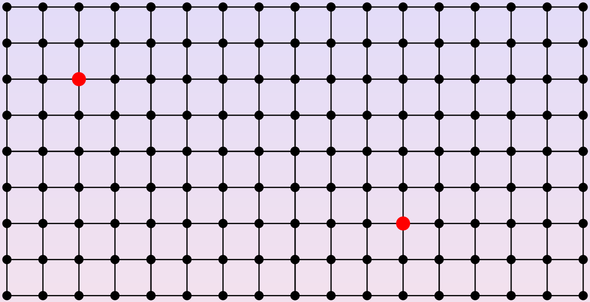
Input: a network G (a weighted connected graph)

Ouput: a **routing scheme** for G

A *routing scheme* is a distributed algorithm that allows **any** source node to route messages to **any** destination node, given the destination's network identifier

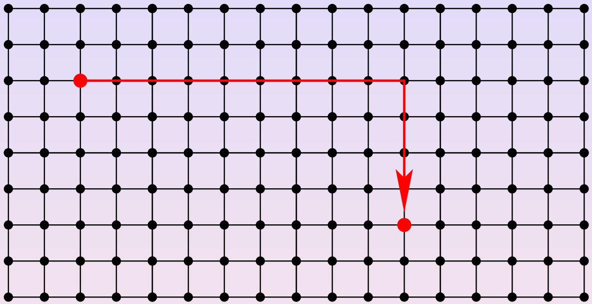
Goal: to minimize the size of the routing tables

Example: Grid with X,Y-coordinates



Routing algorithm: X,Y-routing

Example: Grid with X,Y-coordinates



Routing algorithm: X,Y-routing

Complexity Measures: Space & Stretch

Space = size of the largest local routing tables

Complexity Measures: Space & Stretch

Space = size of the largest local routing tables
(more precisely, size of the smallest local routing algorithm including all constants and data-structures)

In the grid example: $\mathrm{space} = O(\log n)$ bits

Complexity Measures: Space & Stretch

Space = size of the largest local routing tables
(more precisely, size of the smallest local routing algorithm including all constants and data-structures)

In the grid example: $\mathrm{space} = O(\log n)$ bits

Stretch = ratio between length of the route and distance

$$|\mathrm{route}(x, y)| \leqslant \mathrm{stretch} \cdot \mathrm{dist}(x, y)$$

In the grid example: $\mathrm{stretch} = 1$ (shortest path)

Complexity Measures: Space & Stretch

Space = size of the largest local routing tables
(more precisely, size of the smallest local routing algorithm including all constants and data-structures)

In the grid example: $\mathrm{space} = O(\log n)$ bits

Stretch = ratio between length of the route and distance

$$|\mathrm{route}(x, y)| \leqslant \mathrm{stretch} \cdot \mathrm{dist}(x, y)$$

In the grid example: $\mathrm{stretch} = 1$ (shortest path)

Question: for a given family of graphs, find the best space-stretch trade-off

Two variants: Name-independent vs. Labeled

The destination enters the network with its **name**, which is determined by either the designer of the routing scheme (labeled), or an advesary (name-independent).

Labeled: the designer is free to name the nodes according to the topology and the edge weights of the graph

Name-independent: the input is a graph with fixed node manes

An overview: Labeled Model

Labels are of polylogarithmic size
$\tilde{O}(f(n)) = f(n) \cdot \mathrm{polylog}(n)$

network	stretch	space/node (bits)	
arbitrary	1	$n \log n$	[folk]

An overview: Labeled Model

Labels are of polylogarithmic size
$\tilde{O}(f(n)) = f(n) \cdot \mathrm{polylog}(n)$

network	stretch	space/node (bits)	
arbitrary	1	$n \log n$	[folk]
$(2 \leqslant k \in \mathbb{N})$	$4k-5$	$\tilde{O}(n^{1/k})$	[Thorup,Zwick]

An overview: Labeled Model

Labels are of polylogarithmic size
$\tilde{O}(f(n)) = f(n) \cdot \mathrm{polylog}(n)$

network	stretch	space/node (bits)	
arbitrary	1	$n \log n$	[folk]
$(2 \leqslant k \in \mathbb{N})$	$4k-5$	$\tilde{O}(n^{1/k})$	[Thorup,Zwick]
tree	1	$\tilde{O}(1)$	[TZ/Fraigniaud,G.]

An overview: Labeled Model

Labels are of polylogarithmic size
$\tilde{O}(f(n)) = f(n) \cdot \mathrm{polylog}(n)$

network	stretch	space/node (bits)	
arbitrary	1	$n \log n$	[folk]
$(2 \leqslant k \in \mathbb{N})$	$4k-5$	$\tilde{O}(n^{1/k})$	[Thorup,Zwick]
tree	1	$\tilde{O}(1)$	[TZ/Fraigniaud,G.]
doubling-α dim.	$1+\varepsilon$	$\log \Delta$	[Talwar/Slivkins]
		$\tilde{O}(1)$	[Chan et al./Abraham et al.]

An overview: Labeled Model

Labels are of polylogarithmic size
$\tilde{O}(f(n)) = f(n) \cdot \mathrm{polylog}(n)$

network	stretch	space/node (bits)	
arbitrary	1	$n \log n$	[folk]
$(2 \leqslant k \in \mathbb{N})$	$4k-5$	$\tilde{O}(n^{1/k})$	[Thorup,Zwick]
tree	1	$\tilde{O}(1)$	[TZ/Fraigniaud,G.]
doubling-α dim.	$1+\varepsilon$	$\log \Delta$	[Talwar/Slivkins]
		$\tilde{O}(1)$	[Chan et al./Abraham et al.]
planar	$1+\varepsilon$	$\tilde{O}(1)$	[Thorup]

An overview: Labeled Model

Labels are of polylogarithmic size

$\tilde{O}(f(n)) = f(n) \cdot \mathrm{polylog}(n)$

network	stretch	space/node (bits)	
arbitrary	1	$n \log n$	[folk]
$(2 \leqslant k \in \mathbb{N})$	$4k-5$	$\tilde{O}(n^{1/k})$	[Thorup,Zwick]
tree	1	$\tilde{O}(1)$	[TZ/Fraigniaud,G.]
doubling-α dim.	$1+\varepsilon$	$\log \Delta$	[Talwar/Slivkins]
		$\tilde{O}(1)$	[Chan et al./Abraham et al.]
planar	$1+\varepsilon$	$\tilde{O}(1)$	[Thorup]
H-minor-free	$1+\varepsilon$	$\tilde{O}(1)$	[Abraham,G.]

An overview: Name-independent Model

network	stretch	space/node (bits)	
bounded growth	$1+\varepsilon$	$\tilde{O}(1)$	[Abraham et al.]

An overview: Name-independent Model

network	stretch	space/node (bits)	
bounded growth	$1+\varepsilon$	$\tilde{O}(1)$	[Abraham et al.]
doubling-α dim.	$9+\varepsilon$	$\tilde{O}(1)$	[Konjevod et al./Abraham et al.]

An overview: Name-independent Model

network	stretch	space/node (bits)	
bounded growth	$1+\varepsilon$	$\tilde{O}(1)$	[Abraham et al.]
doubling-α dim.	$9+\varepsilon$	$\tilde{O}(1)$	[Konjevod et al./Abraham et al.]
H-minor-free (unweighted)	$O(1)$	$\tilde{O}(1)$	[Abraham et al.]

An overview: Name-independent Model

network	stretch	space/node (bits)	
bounded growth	$1+\varepsilon$	$\tilde{O}(1)$	[Abraham et al.]
doubling-α dim.	$9+\varepsilon$	$\tilde{O}(1)$	[Konjevod et al./Abraham et al.]
H-minor-free (unweighted)	$O(1)$	$\tilde{O}(1)$	[Abraham et al.]
trees	2^k-1	$\tilde{O}(n^{1/k})$	[Laing]

An overview: Name-independent Model

network	stretch	space/node (bits)	
bounded growth	$1+\varepsilon$	$\tilde{O}(1)$	[Abraham et al.]
doubling-α dim.	$9+\varepsilon$	$\tilde{O}(1)$	[Konjevod et al./Abraham et al.]
H-minor-free (unweighted)	$O(1)$	$\tilde{O}(1)$	[Abraham et al.]
trees	2^k-1	$\tilde{O}(n^{1/k})$	[Laing]
arbitrary	3	$\tilde{O}(\sqrt{n})$	[A.,G.,Malkhi,Nisan,Thorup]

An overview: Name-independent Model

network	stretch	space/node (bits)	
bounded growth	$1+\varepsilon$	$\tilde{O}(1)$	[Abraham et al.]
doubling-α dim.	$9+\varepsilon$	$\tilde{O}(1)$	[Konjevod et al./Abraham et al.]
H-minor-free (unweighted)	$O(1)$	$\tilde{O}(1)$	[Abraham et al.]
trees	2^k-1	$\tilde{O}(n^{1/k})$	[Laing]
arbitrary	3	$\tilde{O}(\sqrt{n})$	[A.,G.,Malkhi,Nisan,Thorup]
	$O(k^2 2^k)$	$\tilde{O}(n^{1/k})$	[Arias et al./Awerbuch,Peleg]
	$O(k)$	$\tilde{O}(n^{1/k})$	[Abraham et al.]

Lower Bounds for Name-Independent

Rem: lower bound for labeled $\Rightarrow$ lower bound for name-indep

Lower Bounds for Name-Independent

Rem: lower bound for labeled $\Rightarrow$ lower bound for name-indep

network	stretch	space/node (bits)	
arbitrary	< 1.4	$\Omega(n \log n)$	[G.,Pérennès.]
	< 3	$\Omega(n)$	[G.,Gengler]
(only $k = 1, 2, 3, 5$)	$< 2k + 1$	$\Omega(n^{1/k})$	[Thorup,Zwick]

Lower Bounds for Name-Independent

Rem: lower bound for labeled $\Rightarrow$ lower bound for name-indep

network	stretch	space/node (bits)	
arbitrary	< 1.4	$\Omega(n \log n)$	[G.,Pérennès.]
	< 3	$\Omega(n)$	[G.,Gengler]
(only $k = 1, 2, 3, 5$)	$< 2k + 1$	$\Omega(n^{1/k})$	[Thorup,Zwick]
trees	$\leqslant 3$	$\Omega(\sqrt{n}\,)$	[Laing,Rajaraman]
	$\leqslant 9 - \varepsilon$	$\Omega(n^{(\varepsilon/60)^2})$	[Konjevod et al.]

Lower Bounds for Name-Independent

Rem: lower bound for labeled $\Rightarrow$ lower bound for name-indep

network	stretch	space/node (bits)	
arbitrary	< 1.4	$\Omega(n \log n)$	[G.,Pérennès.]
	< 3	$\Omega(n)$	[G.,Gengler]
(only $k = 1, 2, 3, 5$)	$< 2k + 1$	$\Omega(n^{1/k})$	[Thorup,Zwick]
trees	$\leqslant 3$	$\Omega(\sqrt{n}\,)$	[Laing,Rajaraman]
	$\leqslant 9 - \varepsilon$	$\Omega(n^{(\varepsilon/60)^2})$	[Konjevod et al.]
for all $k \geqslant 1$	$< 2k + 1$	$\Omega((n \log n)^{1/k})$	[Abraham et al.]

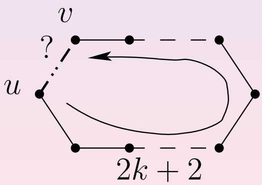
Theorem [Abraham,G.,Malkhi]

1. Any name-indep. routing scheme using $< (n \log n)^{1/k}$ bits/node has a **max stretch** $\geqslant 2k+1$ for some graph.
2. Any name-indep. routing scheme using $< (n/k)^{1/k}$ bits/node has an **average stretch** $\geqslant k/4$ for some graph.

Theorem [Abraham,G.,Malkhi]

1. Any name-indep. routing scheme using $< (n \log n)^{1/k}$ bits/node has a **max stretch** $\geqslant 2k+1$ for some graph.
2. Any name-indep. routing scheme using $< (n/k)^{1/k}$ bits/node has an **average stretch** $\geqslant k/4$ for some graph.

Rem 1: All previous lower bounds for labeled case (Peleg,Upfal / G.,Pérennès / G.,Gengler / Kranakis,Krizanc / Thorup,Zwick) are based on the construction of **dense large girth** graphs



if stretch$< 2k+1$, then u is forced to "know" the edge (u,v)

Theorem [Abraham,G.,Malkhi]

1. Any name-indep. routing scheme using $< (n \log n)^{1/k}$ bits/node has a **max stretch** $\geqslant 2k+1$ for some graph.
2. Any name-indep. routing scheme using $< (n/k)^{1/k}$ bits/node has an **average stretch** $\geqslant k/4$ for some graph.

Erdös Conjecture: $\exists$ graph of girth $2k+2$ with $\Omega(n^{1+1/k})$ edges (proved only for $k = 1, 2, 3, 5$). So, the extra $(\log n)^{1/k}$ term **cannot** be obtained with a girth approach.

Theorem [Abraham,G.,Malkhi]

1. Any name-indep. routing scheme using $< (n \log n)^{1/k}$ bits/node has a **max stretch** $\geqslant 2k+1$ for some graph.
2. Any name-indep. routing scheme using $< (n/k)^{1/k}$ bits/node has an **average stretch** $\geqslant k/4$ for some graph.

Rem 2: It makes a clear separation between labeled and name-independent routing, at least for the average stretch.

In the **labelel** model, $O(\mathrm{polylog}(n))$ space and $O(1)$ average stretch exsits for every graph! [Abraham, Bartal, Chan, Gupta, Kleinberg et al. (FOCS05)]

In the **name-indep** model, if space is $O(\mathrm{polylog}(n))$, then the average stretch must be $\Omega(\log n / \log\log n)$ for some graphs.

The Metric Model

A weaker model, but conceptually easier

Input: a metric space (V, d)

Ouput: an overlay network $G = (V, E)$, and a routing scheme for G

The Metric Model

A weaker model, but conceptually easier

Input: a metric space (V, d)

Ouput: an overlay network $G = (V, E)$, and a routing scheme for G

An extra complexity measure: the size $|E|$ of the overlay

The Metric Model

A weaker model, but conceptually easier

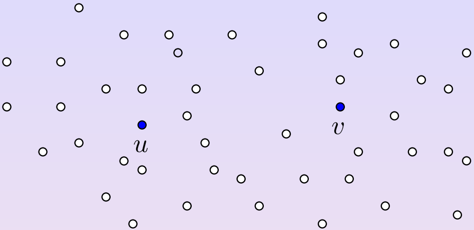
Input: a metric space (V, d)

Ouput: an overlay network $G = (V, E)$, and a routing scheme for G

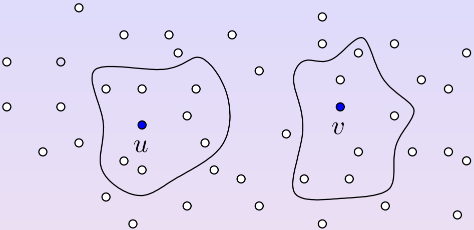
An extra complexity measure: the size $|E|$ of the overlay

Goal: to minimize the size of G, and the space for each node must be $\approx$ the average degree of G

Example: Stretch-3 for Arbitrary Metric

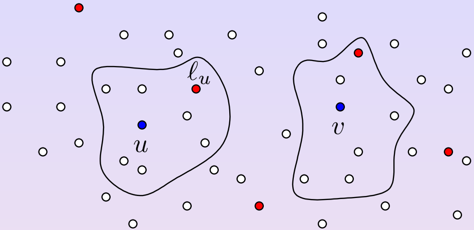


Example: Stretch-3 for Arbitrary Metric



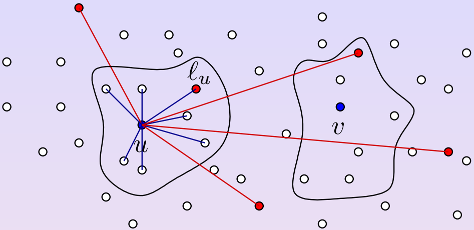
$B_u =$ the set of $\sqrt{n \ln n}$ closest nodes from u

Example: Stretch-3 for Arbitrary Metric



$B_u =$ the set of $\sqrt{n \ln n}$ closest nodes from u
$L =$ a hitting of $\{B_u \mid u \in V\}$ of size $\leqslant \sqrt{n \ln n}$

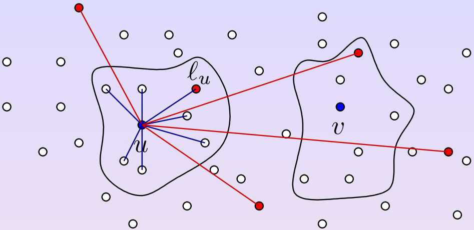
Example: Stretch-3 for Arbitrary Metric



$B_u =$ the set of $\sqrt{n \ln n}$ closest nodes from u
$L =$ a hitting of $\{B_u \mid u \in V\}$ of size $\leqslant \sqrt{n \ln n}$

Overlay: $u \to w$, $\forall w \in B_u$ and $u \to \ell$, $\forall \ell \in L$

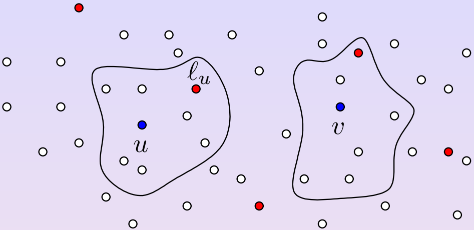
Example: Stretch-3 for Arbitrary Metric



$B_u =$ the set of $\sqrt{n \ln n}$ closest nodes from u
$L =$ a hitting of $\{B_u \mid u \in V\}$ of size $\leqslant \sqrt{n \ln n}$

Overlay: $u \to w$, $\forall w \in B_u$ and $u \to \ell$, $\forall \ell \in L$
$\Rightarrow |E| \leqslant \sum_u (|B_u| + |L|) = \tilde{O}(n^{3/2})$

Example: Stretch-3 for Arbitrary Metric

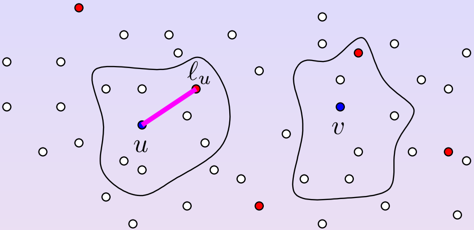


$B_u =$ the set of $\sqrt{n \ln n}$ closest nodes from u
$L =$ a hitting of $\{B_u \mid u \in V\}$ of size $\leqslant \sqrt{n \ln n}$

Overlay: $u \to w$, $\forall w \in B_u$ and $u \to \ell$, $\forall \ell \in L$
$\Rightarrow |E| \leqslant \sum_u (|B_u| + |L|) = \tilde{O}(n^{3/2})$

ROUTING: IF $v \in B_u$, ROUTE $u \to v$, ELSE $u \to \ell_u \to v$

Example: Stretch-3 for Arbitrary Metric

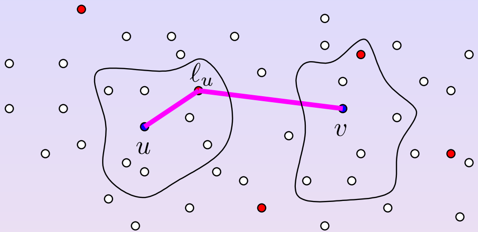


$B_u =$ the set of $\sqrt{n \ln n}$ closest nodes from u
$L =$ a hitting of $\{B_u \mid u \in V\}$ of size $\leqslant \sqrt{n \ln n}$

Overlay: $u \to w$, $\forall w \in B_u$ and $u \to \ell$, $\forall \ell \in L$
$\Rightarrow |E| \leqslant \sum_u (|B_u| + |L|) = \tilde{O}(n^{3/2})$

Routing: If $v \in B_u$, route $u \to v$, else $u \to \ell_u \to v$

Example: Stretch-3 for Arbitrary Metric



$B_u =$ the set of $\sqrt{n \ln n}$ closest nodes from u
$L =$ a hitting of $\{B_u \mid u \in V\}$ of size $\leqslant \sqrt{n \ln n}$

Overlay: $u \to w$, $\forall w \in B_u$ and $u \to \ell$, $\forall \ell \in L$
$\Rightarrow |E| \leqslant \sum_u (|B_u| + |L|) = \tilde{O}(n^{3/2})$

ROUTING: IF $v \in B_u$, ROUTE $u \to v$, ELSE $u \to \ell_u \to v$

Rem: $\ell_u \to v$ is not necessarily easy to implement in the graph model (usually simulated with some tree routings)

Some Results in the Metric Model

Both labeled and name-independent variants exist ...

metric	stretch	average degree	
Euclidian	$O(1)$	$O(1)$	[Abraham,Malkhi/Hassin,Peleg]

Some Results in the Metric Model

Both labeled and name-independent variants exist ...

metric	stretch	average degree	
Euclidian	$O(1)$	$O(1)$	[Abraham,Malkhi/Hassin,Peleg]
doubling-α dim.	$1+\varepsilon$	$\tilde{O}(\log\Delta)$	[Talwar/Chan et al./Slivkins]
	$1+\varepsilon$	$\tilde{O}(1)$	[Abraham et al.]

Undirected vs. Directed

(graph model only!)

Problem: there is no stretch-space trade-off for routing in *directed* graphs! The stretch maybe not bounded if $o(n)$ bits of memory are used, even in strongly connected digraphs [Thorup,Zwick]

Undirected vs. Directed

(graph model only!)

Problem: there is no stretch-space trade-off for routing in *directed* graphs! The stretch maybe not bounded if $o(n)$ bits of memory are used, even in strongly connected digraphs [Thorup,Zwick]

New measure: roundtrip stretch factor

$$\text{stretch} \;=\; \frac{|\text{route}(u,v)| + |\text{route}(v,u)|}{\text{dist}(u,v) + \text{dist}(v,u)}$$

Rem: $\text{dist}(u,v) + \text{dist}(v,u)$ is now a distance function

Some Results for Arbitrary Digraphs

Labeled: [Roditty,Thorup,Zwick - SODA '02]

stretch$=4k+\varepsilon$	stretch$=3$
space$=\tilde{O}(\varepsilon^{-1}kn^{1/k}\log\Delta)$	space$=\tilde{O}(\sqrt{n}\,)$
labels$=o(\varepsilon^{-1}k\log^2 n\log\Delta)$	labels$=o(\log^2 n)$

Name-independent: [Arias,Cowen,Laing - PODC '03]

stretch$=O(k^2)$	stretch$=6$
space$=\tilde{O}(\varepsilon^{-1}kn^{1/k}\log\Delta)$	space$=\tilde{O}(\sqrt{n}\,)$
labels$=o(\varepsilon^{-1}k^2\log^2 n\log\Delta)$	labels$=o(\log^2 n)$

Lower bound: if stretch <2, then $\Omega(n)$ bits is required

Open Questions: For Arbitrary Networks

Q1: Labeled: stretch $\leqslant 4k-5$ for $\tilde{O}(n^{1/k})$ memory. Optimal only for $k=1,2$. The lower bound on the stretch is $\leqslant 2k-1$. For $k=4$, the lower bound is not known to be $2k-1$ (the Erdös conjecture is proved only for $k=1,2,3,5$)

Open Questions: For Arbitrary Networks

Q1: Labeled: stretch $\leqslant 4k-5$ for $\tilde{O}(n^{1/k})$ memory. Optimal only for $k=1,2$. The lower bound on the stretch is $\leqslant 2k-1$. For $k=4$, the lower bound is not known to be $2k-1$ (the Erdös conjecture is proved only for $k=1,2,3,5$)

Q2: Name-independent = labeled ???
For $k=1,2$, the same bounds hold.

Open Questions: For Arbitrary Networks

Q1: Labeled: stretch $\leqslant 4k-5$ for $\tilde{O}(n^{1/k})$ memory. Optimal only for $k=1,2$. The lower bound on the stretch is $\leqslant 2k-1$. For $k=4$, the lower bound is not known to be $2k-1$ (the Erdös conjecture is proved only for $k=1,2,3,5$)

Q2: Name-independent = labeled ???
For $k=1,2$, the same bounds hold.

Q3: Directed = Undirected???

Open Questions: For Specific Networks

Q4: Trees unweighted name-indep: what's the best stretch with $\tilde{O}(1)$ memory? Currently stretch ≈ 17.

Open Questions: For Specific Networks

Q4: Trees unweighted name-indep: what's the best stretch with $\tilde{O}(1)$ memory? Currently stretch ≈ 17.

Q5: Labeled treewidth-k & shortest path: $o(k \log^2 n)$-bit labels? True for trees $k = 1$ [Fraigniaud,G.] and weighted outerplanar $k = 2$ [Dieng, G.]: $\Theta(\log^2 n / \log \log n)$ bits are enough and necessary.

Open Questions: For Specific Networks

Q4: Trees unweighted name-indep: what's the best stretch with $\tilde{O}(1)$ memory? Currently stretch ≈ 17.

Q5: Labeled treewidth-k & shortest path: $o(k \log^2 n)$-bit labels? True for trees $k = 1$ [Fraigniaud,G.] and weighted outerplanar $k = 2$ [Dieng, G.]: $\Theta(\log^2 n / \log\log n)$ bits are enough and necessary.

Q6: Shortest path in planar with $\tilde{O}(1)$ labels: $\Omega(n^{1/3})$ ... $O(n)$ (currently $7.18n$ bits [Lu '02])

Future Works (1/2)

W1: $\tilde{O}(\deg(u))$ with stretch $O(1)$ for general graphs?

Future Works (1/2)

W1: $\tilde{O}(\deg(u))$ with stretch $O(1)$ for general graphs?

W2: Bounded degree?
(sparse graphs are known to be non-compact.
Bounded degree nodes "increase" distances, so
stretch tends to 1. No lower bounds is known.
Bounded degree includes expanders ...)

Future Works (1/2)

W1: $\tilde{O}(\deg(u))$ with stretch $O(1)$ for general graphs?

W2: Bounded degree?
(sparse graphs are known to be non-compact. Bounded degree nodes “increase” distances, so stretch tends to 1. No lower bounds is known. Bounded degree includes expanders ...)

W3: Routing with additive stretch?
(initial works in random power law networks [Brady,Cowen '06]. The addtive stretch and the polylog labels depend on the graph parameter only. Works well in practice. Connection with distance labeling)

Future Works (2/2)

W4: Average stretch? ε-slack routing?
(labeled and name-indep differ. Average stretch & additive stretch are interesting in practice)

Future Works (2/2)

W4: Average stretch? ε-slack routing?
(labeled and name-indep differ. Average stretch & additive stretch are interesting in practice)

W5: Dynamic routing: Yes [Korman,Peleg] but not yet compact ...

Future Works (2/2)

W4: Average stretch? ε-slack routing?
(labeled and name-indep differ. Average stretch & additive stretch are interesting in practice)

W5: Dynamic routing: Yes [Korman,Peleg] but not yet compact ...

W6: Distributed algorithms for constructing tables? Yes [Frederickson'90] for some speficic graphs (planar). Distributed implementation is possible but ... complicated!

Thank you!