Compact Routing with Name Independence

Kofi Laing

joint work with:
M. Arias, L. Cowen, R. Rajaraman, O. Taka
Computer Science Departments,
Tufts and NorthEastern Universities
Outline

• Problem Definition: Memory vs Stretch

• Past Work

• Low-Stretch Schemes with Sublinear space
  ★ Single Source Scheme
  ★ Schemes A, B & C

• Tradeoff Schemes with Sublinear space
  ★ Exponential Stretch
  ★ Polynomial Stretch

• Result Summary & Conclusions
Routing Problem

Diagram showing a network with nodes and edges labeled with distances.
Routing Problem

a
b
c
d
e
f g

a: 2
b: 1
c: 2
d: 2
e: 1
f: 1
g: −
Routing Problem

a: 2
b: 1
c: 2
d: 2
e: 1
f: 1
g: −
Routing Problem

a: 2
b: 1
c: 2
d: 2
e: 1
f: 1
g: −
Routing Problem

- a: 3
- b: 2
- c: 3
- d: −
- e: 2
- f: 2
- g: 1
Routing Problem

a: 3  
b: 2  
c: 3  
d: -  
e: 2  
f: 2  
g: 1
Routing Problem

\[ a: 3 \\
 b: 2 \\
 c: 3 \\
 d: - \\
 e: 2 \\
 f: 2 \\
 g: 1 \]
Routing Problem

Graph with nodes labeled a, b, c, d, e, f, g, and edges with weights.

- Node a connected to nodes 1, 2, 3, 8
- Node b connected to nodes 1
- Node c connected to nodes 1, 2, 3, 6
- Node d connected to nodes 2, 4, 21
- Node e connected to nodes 1, 2, 4, 5
- Node f connected to nodes 1, 2, 7
- Node g connected to nodes 1, 2, 6

Weights:
- Node a: 1
- Node b: 1
- Node c: -
- Node d: 2
- Node e: 3
- Node f: 1
- Node g: 2
Routing Problem

\[ \begin{array}{c}
 a & b & c & d & e & f & g \\
 1 & 1 & - & 2 & 3 & 1 & 2 \\
 8 & 10 & 7 & 6 & 2 & 4 & 126 \\
\end{array} \]
Routing Problem

Shortest path routing possible with $O(n \log n)$ sized local tables.
Compact Routing Problem

Given a graph $G = (V, E)$ with positive weighted edges

**Defn:** $d(u, v)$: shortest distance from $u$ to $v$

**Defn:** $p_A(u, v)$: distance of specified path from $u$ to $v$
Compact Routing Problem

Given a graph $G = (V, E)$ with positive weighted edges

**Defn:** $d(u, v)$: shortest distance from $u$ to $v$

**Defn:** $p_A(u, v)$: distance of specified path from $u$ to $v$

**Defn:** stretch: $\max_{(u,v) \in V^2} = \frac{p_A(u,v)}{d(u,v)}$

Problem: finding good tradeoffs of maximum local space for stretch
Node Naming Models

- Name-dependent: named with $O(\log^2 n)$ sized addresses
Node Naming Models

- Name-dependent: named with $O(\log^2 n)$ sized addresses
- permutation: named $\{0, \ldots, n - 1\}$ by algorithm

Row Major Labelling
Name Dependent
or Permutation
Node Naming Models

- Name-dependent: named with $O(\log^2 n)$ sized addresses
  - permutation: named $\{0, \ldots, n - 1\}$ by algorithm
- Name-independent: named $\{0, \ldots, n - 1\}$ adversarially.

Arbitrary Permutation
Name Independent !!!

2  10  5  15
6 13  8  1
3  0 12 9
14 11  4  7
Port Naming Models
(Burhman et al)

**Defn:** Neighbors: Named \(\{0, \ldots, n - 1\}\) for neighbors
Defn: Neighbors: Named \(\{0, \ldots, n - 1\}\) for neighbors

Defn: Fixed: Named \(\{1, \ldots, \text{deg}(v)\}\) adversarially
Port Naming Models
(Burhman et al)

**Defn:** Neighbors: Named \( \{0, \ldots, n - 1\} \) for neighbors

**Defn:** Fixed: Named \( \{1, \ldots, \text{deg}(v)\} \) adversarially

**Defn:** Free: Named \( \{1, \ldots, \text{deg}(v)\} \) by algorithm
**Defn:** Neighbors: Named \( \{0, \ldots, n - 1\} \) for neighbors

**Defn:** Fixed: Named \( \{1, \ldots, \text{deg}(v)\} \) adversarially

**Defn:** Free: Named \( \{1, \ldots, \text{deg}(v)\} \) by algorithm
Summary of Name-dependent Results
Special Networks

- Rings and Trees (Santoro & Khabib 85)
- Complete Networks, Grids (van Leeuwen & Tan 87)
- Small Separators (Frederickson & Jarnadan 88)
- Planar Networks (Frederickson & Jarnadan 89)

<table>
<thead>
<tr>
<th>Graph</th>
<th>Stretch</th>
<th>Space</th>
<th>Header</th>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trees</td>
<td>1</td>
<td>(\tilde{O}(\sqrt{n}))</td>
<td>(O(\log n))</td>
<td>Cowen 01</td>
</tr>
<tr>
<td>Trees</td>
<td>1</td>
<td>(\tilde{O}(1))</td>
<td>(O(\log^2 n))</td>
<td>TZ01 &amp; FG01</td>
</tr>
</tbody>
</table>
### Previous Name-dependent Results

**Upper Bounds, Universal**

**Defn:** \( \tilde{O}(g(n)) \) means \( \{ f(n) | \exists c, k, n_0 \text{ such that } 0 \leq f(n) \leq cg(n) \log^k(n) \text{ for all } n \geq n_0 \} \).

<table>
<thead>
<tr>
<th>Stretch</th>
<th>Space</th>
<th>Citation, Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>12k + 3</td>
<td>( O(n^{1+1/k}) ) TOTAL</td>
<td>Peleg &amp; Upfal 88</td>
</tr>
<tr>
<td>5</td>
<td>( \tilde{O}(n^{1/2}) ) local</td>
<td>Eliam, Gavoille &amp; Peleg 98</td>
</tr>
<tr>
<td>3</td>
<td>( \tilde{O}(n^{2/3}) ) local</td>
<td>Cowen 99</td>
</tr>
<tr>
<td>3</td>
<td>( \tilde{O}(n^{1/2}) ) local</td>
<td>Thorup and Zwick 01</td>
</tr>
<tr>
<td>(2k − 1)</td>
<td>( O(n^{1/k}) ) local</td>
<td><em>handshaking</em>, TZ01</td>
</tr>
<tr>
<td>(4k − 5)</td>
<td>( O(n^{1/k}) ) local</td>
<td>without <em>handshaking</em>, TZ01</td>
</tr>
</tbody>
</table>
Lower Bounds

Gavoille & Gengler 97: $\Omega(n)$ sized tables for max stretch $< 3$.

Buhrman, Hoepman and Vitanyi 99:

- $\Omega(n^2 \log n)$ bits TOTAL for $s < 2$ (Name-Independent)
- $(n^2/2) \log n$ bits TOTAL for $s < 2$ (Name-Independent, Fixed Ports)
- $\Omega(n^2)$ bits TOTAL for $s < 2$ (Name-Independent and Neighbor Ports, or Fixed Ports or Free Ports)
Compact Routing is Possible! ABLP-89

- Arbitrary Graphs (Universality), arbitrary edge weights
- Constant size stretch (independent of $n$)
- Polylogarithmic-sized routing headers,
Compact Routing is Possible! ABLP-89

- Arbitrary Graphs (Universality), arbitrary edge weights
- Constant size stretch (independent of $n$)
- Polylogarithmic-sized routing headers,
- Topology-independent node names
Compact Routing is Possible! ABLP-89

- Arbitrary Graphs (Universality), arbitrary edge weights
- Constant size stretch (independent of $n$)
- Polylogarithmic-sized routing headers,
- Topology-independent node names
- Balanced Sublinear-space routing tables
Compact Routing is Possible! ABLP-89

- Arbitrary Graphs (Universality), arbitrary edge weights
- Constant size stretch (independent of $n$)
- Polylogarithmic-sized routing headers,
- Topology-independent node names
- Balanced Sublinear-space routing tables
- Distributed Routing Table Construction
## New and Known (Minimum) Results
*(Name Independent Routing Only)*

<table>
<thead>
<tr>
<th></th>
<th>Table size</th>
<th>Header Size</th>
<th>Stretch</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABLP-89</td>
<td>$\tilde{O} \left( n^{2/3} \right) $</td>
<td>$O(\log n)$</td>
<td>486</td>
</tr>
<tr>
<td>ABLP-89</td>
<td>$\tilde{O} \left( n^{1/2} \right) $</td>
<td>$O(\log n)$</td>
<td>2592</td>
</tr>
<tr>
<td></td>
<td>Table size</td>
<td>Header Size</td>
<td>Stretch</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>ABLP-89</td>
<td>$\tilde{O}(n^{2/3})$</td>
<td>$O(\log n)$</td>
<td>486</td>
</tr>
<tr>
<td>ABLP-89</td>
<td>$\tilde{O}(n^{1/2})$</td>
<td>$O(\log n)$</td>
<td>2592</td>
</tr>
<tr>
<td>AP-90</td>
<td>$\tilde{O}(n^{2/3})$</td>
<td>$O(\log n)$</td>
<td>624</td>
</tr>
<tr>
<td>AP-90</td>
<td>$\tilde{O}(n^{1/2})$</td>
<td>$O(\log n)$</td>
<td>1088</td>
</tr>
</tbody>
</table>
## New and Known (Minimum) Results (Name Independent Routing Only)

<table>
<thead>
<tr>
<th></th>
<th>Table size</th>
<th>Header Size</th>
<th>Stretch</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABLP-89</td>
<td>(\tilde{O} \left( n^{2/3} \right) )</td>
<td>(O(\log n))</td>
<td>486</td>
</tr>
<tr>
<td>ABLP-89</td>
<td>(\tilde{O} \left( n^{1/2} \right) )</td>
<td>(O(\log n))</td>
<td>2592</td>
</tr>
<tr>
<td>AP-90</td>
<td>(\tilde{O} \left( n^{2/3} \right) )</td>
<td>(O(\log n))</td>
<td>624</td>
</tr>
<tr>
<td>AP-90</td>
<td>(\tilde{O} \left( n^{1/2} \right) )</td>
<td>(O(\log n))</td>
<td>1088</td>
</tr>
<tr>
<td><strong>Our Scheme A</strong></td>
<td>(\tilde{O} \left( n^{1/2} \right) )</td>
<td>(O(\log^2 n))</td>
<td>5</td>
</tr>
<tr>
<td><strong>Our Scheme B</strong></td>
<td>(\tilde{O} \left( n^{1/2} \right) )</td>
<td>(O(\log n))</td>
<td>7</td>
</tr>
<tr>
<td><strong>Our Scheme C</strong></td>
<td>(\tilde{O} \left( n^{2/3} \right) )</td>
<td>(O(\log n))</td>
<td>5</td>
</tr>
</tbody>
</table>
# New and Known Tradeoff Schemes
*(Name Independent Routing Only)*

<table>
<thead>
<tr>
<th></th>
<th>Tables</th>
<th>Headers</th>
<th>Stretch</th>
</tr>
</thead>
<tbody>
<tr>
<td>P89</td>
<td>$\tilde{O}(n^{2/k})$</td>
<td>$O(\log n)$</td>
<td>$O(2^k)$</td>
</tr>
</tbody>
</table>
### New and Known Tradeoff Schemes
**(Name Independent Routing Only)**

<table>
<thead>
<tr>
<th></th>
<th>Tables</th>
<th>Headers</th>
<th>Stretch</th>
</tr>
</thead>
<tbody>
<tr>
<td>P89</td>
<td>$\tilde{O} \left( \frac{n^2}{k} \right)$</td>
<td>$O(\log n)$</td>
<td>$O(2^k)$</td>
</tr>
<tr>
<td>ABLP-89</td>
<td>$\tilde{O} \left( \frac{n^2}{k} \right)$</td>
<td>$O(\log n)$</td>
<td>$O(k^2 3^k)$</td>
</tr>
</tbody>
</table>
### New and Known Tradeoff Schemes
(Name Independent Routing Only)

<table>
<thead>
<tr>
<th></th>
<th>Tables</th>
<th>Headers</th>
<th>Stretch</th>
</tr>
</thead>
<tbody>
<tr>
<td>P89</td>
<td>( \tilde{O}(n^2/k) )</td>
<td>( O(\log n) )</td>
<td>( O(2^k) )</td>
</tr>
<tr>
<td>ABLP-89</td>
<td>( \tilde{O}(n^2/k) )</td>
<td>( O(\log n) )</td>
<td>( O(k^2 3^k) )</td>
</tr>
<tr>
<td>AP-90</td>
<td>( \tilde{O}(n^2/k) )</td>
<td>( O(\log n) )</td>
<td>( 64k^2 + 16k )</td>
</tr>
</tbody>
</table>
New and Known Tradeoff Schemes
(Name Independent Routing Only)

<table>
<thead>
<tr>
<th></th>
<th>Tables</th>
<th>Headers</th>
<th>Stretch</th>
</tr>
</thead>
<tbody>
<tr>
<td>P89</td>
<td>$\tilde{O}(\frac{n^2}{k})$</td>
<td>$O(\log n)$</td>
<td>$O(2^k)$</td>
</tr>
<tr>
<td>ABLP-89</td>
<td>$\tilde{O}(\frac{n^2}{k})$</td>
<td>$O(\log n)$</td>
<td>$O(k^2 3^k)$</td>
</tr>
<tr>
<td>AP-90</td>
<td>$\tilde{O}(\frac{n^2}{k})$</td>
<td>$O(\log n)$</td>
<td>$64k^2 + 16k$</td>
</tr>
<tr>
<td>Our Expo</td>
<td>$\tilde{O}(\frac{n^2}{k})$</td>
<td>$O(\log^2 n)$</td>
<td>$1 + (k - 1)(2^k - 2)$</td>
</tr>
<tr>
<td>Our Poly</td>
<td>$\tilde{O}(\frac{n^2}{k})$</td>
<td>$O(\log^2 n)$</td>
<td>$16k^2 + 4k$</td>
</tr>
</tbody>
</table>
Lemma 1. There exists a name-independent routing scheme on any tree $T$ with space $\tilde{O}(\sqrt{n})$, and $O(\log n)$-sized headers which achieves stretch 3 for paths from root.
Lemma 1. There exists a name-independent routing scheme on any tree $T$ with space $\tilde{O}(\sqrt{n})$, and $O(\log n)$-sized headers which achieves stretch 3 for paths from root.

Ingredients:

<table>
<thead>
<tr>
<th>Graph</th>
<th>Stretch</th>
<th>Space</th>
<th>Header</th>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trees</td>
<td>1</td>
<td>$\tilde{O}(\sqrt{n})$</td>
<td>$O(\log n)$</td>
<td>Cowen 01</td>
</tr>
</tbody>
</table>
Lemma 1. There exists a name-independent routing scheme on any tree $T$ with space $\tilde{O}(\sqrt{n})$, and $O(\log n)$-sized headers which achieves stretch 3 for paths from root.

Ingredients:

<table>
<thead>
<tr>
<th>Graph</th>
<th>Stretch</th>
<th>Space</th>
<th>Header</th>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trees</td>
<td>1</td>
<td>$\tilde{O}(\sqrt{n})$</td>
<td>$O(\log n)$</td>
<td>Cowen 01</td>
</tr>
</tbody>
</table>

- a distributed dictionary, first defined by Peleg
Lemma 1. There exists a name-independent routing scheme on any tree $T$ with space $\tilde{O}(\sqrt{n})$, and $O(\log n)$-sized headers which achieves stretch 3 for paths from root.

Ingredients:

<table>
<thead>
<tr>
<th>Graph</th>
<th>Stretch</th>
<th>Space</th>
<th>Header</th>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trees</td>
<td>1</td>
<td>$\tilde{O}(\sqrt{n})$</td>
<td>$O(\log n)$</td>
<td>Cowen 01</td>
</tr>
</tbody>
</table>

- a distributed dictionary, first defined by Peleg
- new randomized block assignment of ranges of addresses
Lemma 1. There exists a name-independent routing scheme on any tree $T$ with space $\tilde{O}(\sqrt{n})$, and $O(\log n)$-sized headers which achieves stretch 3 for paths from root.
Lemma 1. There exists a name-independent routing scheme on any tree $T$ with space $\tilde{O}(\sqrt{n})$, and $O(\log n)$-sized headers which achieves stretch 3 for paths from root.
Lemma 1. There exists a name-independent routing scheme on any tree $T$ with space $\tilde{O}(\sqrt{n})$, and $O(\log n)$-sized headers which achieves stretch 3 for paths from root.
Lemma 1. There exists a name-independent routing scheme on any tree $T$ with space $\tilde{O}(\sqrt{n})$, and $O(\log n)$-sized headers which achieves stretch 3 for paths from root.

$B_0 = (00,01,02,03)$
$B_1 = (10,11,12,13)$
$B_2 = (20,21,22,23)$
$B_3 = (30,31,32,33)$

$N(s)$
Lemma 1. There exists a name-independent routing scheme on any tree $T$ with space $\tilde{O}(\sqrt{n})$, and $O(\log n)$-sized headers which achieves stretch 3 for paths from root.
Lemma 1. There exists a name-independent routing scheme on any tree $T$ with space $\tilde{O}(\sqrt{n})$, and $O(\log n)$-sized headers which achieves stretch 3 for paths from root.

- $B_0 = (00, 01, 02, 03)$
- $B_1 = (10, 11, 12, 13)$
- $B_2 = (20, 21, 22, 23)$
- $B_3 = (30, 31, 32, 33)$
Lemma 1. There exists a name-independent routing scheme on any tree $T$ with space $\tilde{O}(\sqrt{n})$, and $O(\log n)$-sized headers which achieves stretch 3 for paths from root.

- $B_0 = (00, 01, 02, 03)$
- $B_1 = (10, 11, 12, 13)$
- $B_2 = (20, 21, 22, 23)$
- $B_3 = (30, 31, 32, 33)$
Lemma 1. There exists a name-independent routing scheme on any tree $T$ with space $\tilde{O}(\sqrt{n})$, and $O(\log n)$-sized headers which achieves stretch 3 for paths from root.

$B_0 = (00,01,02,03)$

$B_1 = (10,11,12,13)$

$B_2 = (20,21,22,23)$

$B_3 = (30,31,32,33)$
Using Trees in General Networks

Recall Existing Algorithms:

<table>
<thead>
<tr>
<th>Graph</th>
<th>Stretch</th>
<th>Space</th>
<th>Header</th>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trees</td>
<td>1</td>
<td>$\tilde{O}(\sqrt{n})$</td>
<td>$O(\log n)$</td>
<td>Cowen 01</td>
</tr>
<tr>
<td>Trees</td>
<td>1</td>
<td>$O(1)$</td>
<td>$O(\log^2 n)$</td>
<td>TZ01 &amp; FG01</td>
</tr>
</tbody>
</table>

Possible Approaches??

- Cowen Single Source Algorithm from each point
- FG or TZ Single Source Algorithm from each point
Directions Using Landmarks

- Directions: Go to the Landmark, and from there, ...

- Landmark: a location which is
  - easy to find
  - easy to specify directions from
Directions Using Landmarks

- Directions: Go to the Landmark, and from there, ...

- Landmark: a location which is
  - easy to find
  - easy to specify directions from

- Make tree rooted at each landmark
Lovasz Hitting Set Lemma

- undirected (weighted) $G$ with $n$ nodes, $m$ edges
Lovasz Hitting Set Lemma

- undirected (weighted) $G$ with $n$ nodes, $m$ edges
- *neighborhood ball* $N(u)$: $n^{1/2}$ nodes closest to $u$
Lovasz Hitting Set Lemma

- undirected (weighted) $G$ with $n$ nodes, $m$ edges
- neighborhood ball $N(u)$: $n^{1/2}$ nodes closest to $u$
- Lovasz Lemma: $\exists$ hitting set $L \subset V$:
  - $\forall v, N(v) \cap L \neq \emptyset$
  - $|L| = O(n^{1/2} \log n)$
  - $L$ computed greedily in $\tilde{O}(m + n^{3/2})$ time
Neighborhoods centered everywhere!

**Lemma 2.** Let \( \{B_i|0 \leq i < \sqrt{n}\} \) denote a set of blocks. \( \exists \) sets \( S_v \) of blocks for nodes \( v \), so that

- \( \forall v \in G, \forall B_i, \text{there exists an } S_j \in N(v) \text{ with } B_i \in S_j \)
- \( \forall v \in G, |S_v| = O(\log n) \)
General Networks

$\sqrt{n}$-Block Distribution Lemma

Neighborhoods centered everywhere!

**Lemma 2.** Let $\{B_i | 0 \leq i < \sqrt{n}\}$ denote a set of blocks. ∃ sets $S_v$ of blocks for nodes $v$, so that

- $\forall v \in G, \forall B_i$, there exists an $S_j \in N(v)$ with $B_i \in S_j$
- $\forall v \in G, |S_v| = O(\log n)$
Neighborhoods centered everywhere!

**Lemma 2.** Let \( \{B_i|0 \leq i < \sqrt{n}\} \) denote a set of blocks. \( \exists \) sets \( S_v \) of blocks for nodes \( v \), so that

- \( \forall v \in G, \forall B_i, \) there exists an \( S_j \in N(v) \) with \( B_i \in S_j \)
- \( \forall v \in G, |S_v| = O(\log n) \)
Neighborhoods centered everywhere!

**Lemma 2.** Let \( \{B_i|0 \leq i < \sqrt{n}\} \) denote a set of blocks. \( \exists \) sets \( S_v \) of blocks for nodes \( v \), so that

- \( \forall v \in G, \forall B_i, \) there exists an \( S_j \in N(v) \) with \( B_i \in S_j \)
- \( \forall v \in G, |S_v| = O(\log n) \)
Neighborhoods centered everywhere!

**Lemma 2.** Let $\{B_i | 0 \leq i < \sqrt{n}\}$ denote a set of blocks. \exists sets $S_v$ of blocks for nodes $v$, so that

- $\forall v \in G, \forall B_i$, there exists an $S_j \in N(v)$ with $B_i \in S_j$
- $\forall v \in G, |S_v| = O(\log n)$
Lemma 3. ABLP89: Given a weighted undirected graph $G$ if $w \in N(u)$ and $v$ is on the shortest path from $u$ to $w$ then $w \in N(v)$.

- helps with optimal routing within $N(u)$
- removes need for tree-based lookup routing
Lemma 3. ABLP89: Given a weighted undirected graph $G$ if $w \in N(u)$ and $v$ is on the shortest path from $u$ to $w$ then $w \in N(v)$.

- helps with optimal routing within $N(u)$
- removes need for tree-based lookup routing
Lemma 3. ABLP89: Given a weighted undirected graph $G$ if $w \in N(u)$ and $v$ is on the shortest path from $u$ to $w$ then $w \in N(v)$.

- helps with optimal routing within $N(u)$
- removes need for tree-based lookup routing
Lemma 3. ABLP89: Given a weighted undirected graph $G$ if $w \in N(u)$ and $v$ is on the shortest path from $u$ to $w$ then $w \in N(v)$.

- helps with optimal routing within $N(u)$
- removes need for tree-based lookup routing
Scheme A: Per Node Storage

1. $\forall v$ in $N(u)$, $(v, e_{uv})$.

2. $\forall l \in L$, $(l, e_{ul})$.

3. $\forall 0 \leq i < \sqrt{n}$, the pair $(i, t)$ where $t \in N(u)$ satisfies $B_i \in S_t$
Scheme A: Per Node Storage

1. $\forall v \in N(u), (v, e_{uv})$.

2. $\forall l \in L, (l, e_{ul})$.

3. $\forall 0 \leq i < \sqrt{n}$, the pair $(i, t)$ where $t \in N(u)$ satisfies $B_i \in S_t$.

4. $\forall B_k \in S_u, \forall j \in B_k$,
   - $l_g \in L$ that minimizes $d(u, l_g) + d(l_g, j)$
   - tree-routing address $R(j)$ in tree $T_{l_g}$.

5. $\forall l \in L$, $Tab(u)$ for tree $T_l$. 
Lemma 4. The stretch of Scheme A is bounded by 5, uses $\tilde{O}(\sqrt{n})$ space, and $O(\log^2 n)$ headers.
Lemma 4. The stretch of Scheme A is bounded by 5, uses $\tilde{O}(\sqrt{n})$ space, and $O(\log^2 n)$ headers.
Lemma 4. The stretch of Scheme A is bounded by 5, uses $\tilde{O}(\sqrt{n})$ space, and $O(\log^2 n)$ headers.
**Lemma 4.** The stretch of Scheme A is bounded by 5, uses $\tilde{O}(\sqrt{n})$ space, and $O(\log^2 n)$ headers.
Lemma 4. The stretch of Scheme A is bounded by 5, uses $\tilde{O}(\sqrt{n})$ space, and $O(\log^2 n)$ headers.
Lemma 4. The stretch of Scheme A is bounded by 5, uses \( \tilde{O}(\sqrt{n}) \) space, and \( O(\log^2 n) \) headers.
Lemma 4. The stretch of Scheme A is bounded by 5, uses $\tilde{O}(\sqrt{n})$ space, and $O(\log^2 n)$ headers.
Lemma 4. The stretch of Scheme A is bounded by 5, uses $\tilde{O}(\sqrt{n})$ space, and $O(\log^2 n)$ headers.
Schemes B & C: Routing Algorithms
Schemes B & C: Routing Algorithms
Schemes B & C: Routing Algorithms

$G$

$N(u)$

$N(w)$

$i$ $u$ $w$

$lw$
Schemes B & C: Routing Algorithms

\[ G \]

\[ N(u) \]

\[ N(w) \]

\[ i \rightarrow u \]

\[ lw \rightarrow w \]
Schemes B & C: Routing Algorithms
Schemes B & C: Routing Algorithms

$G$

$N(u)$

$i$

$u$

$w$

$Iw$

$N(w)$
Schemes B & C: Routing Algorithms

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Stretch</th>
<th>Space</th>
<th>Header Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5</td>
<td>$\mathcal{O}(\sqrt{n})$</td>
<td>$O(\log^2 n)$</td>
</tr>
<tr>
<td>B</td>
<td>7</td>
<td>$\mathcal{O}(\sqrt{n})$</td>
<td>$O(\log n)$</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
<td>$\mathcal{O}(n^{2/3})$</td>
<td>$O(\log n)$</td>
</tr>
</tbody>
</table>
Generalizations to $\tilde{O}(n^{1/k})$ Space

Concept of Incremental Prefix Matching
Generalizations to $\tilde{O}(n^{1/k})$ Space

Concept of Incremental Prefix Matching
Generalizations to $\tilde{O} \left( n^{1/k} \right)$ Space

Concept of Incremental Prefix Matching
Generalizations to $\tilde{O}(n^{1/k})$ Space
Concept of Incremental Prefix Matching
Generalizations to $\tilde{O}(n^{1/k})$ Space
Concept of Incremental Prefix Matching
Generalizations to $\tilde{O}(n^{1/k})$ Space
Concept of Incremental Prefix Matching
Generalizations to $\tilde{O}(n^{1/k})$ Space
Concept of Incremental Prefix Matching
Generalizations to $\tilde{O}(n^{1/k})$ Space
Concept of Incremental Prefix Matching
Generalizations to $\tilde{O}(n^{1/k})$ Space

Concept of Incremental Prefix Matching
Generalizations to $\tilde{O}(n^{1/k})$ Space

Concept of Incremental Prefix Matching
Generalizations to $\tilde{O}(n^{1/k})$ Space

Concept of Incremental Prefix Matching
Generalizations to $\tilde{O}(n^{1/k})$ Space
Concept of Incremental Prefix Matching
Generalizations to $\tilde{O}\left(n^{1/k}\right)$ Space
Concept of Incremental Prefix Matching
Generalizing to $\tilde{O}(n^{1/k})$ Space – Definitions

**Defn:** $\Sigma$: set $\{0, \ldots, n^{1/k} - 1\}$

**Defn:** $N^i(v)$: set of $n^{i/k}$ nodes closest to $v$. 
Generalizing to $\tilde{O} \left( n^{1/k} \right)$ Space – Definitions

**Defn:** $\Sigma$: set $\{0, \ldots, n^{1/k} - 1\}$

**Defn:** $N^i(v)$: set of $n^{i/k}$ nodes closest to $v$.

**Defn:** $\sigma^i(\alpha\beta) = \alpha$ iff $|\alpha| = i$, for $\alpha, \beta \in \Sigma^*$

**Defn:** $\langle v \rangle$: node id of $v$ in base $n^{1/k}$, padded to length $k$.

Examples with 10000 nodes and $k = 4$:

- $\sigma^3(\langle 1234 \rangle) = \sigma^3(1234) = 123$
- $\sigma^3(\langle 3 \rangle) = \sigma^3(0003) = 000$
Generalizing to $\tilde{O}(n^{1/k})$ Space – Definitions

**Defn:** $\Sigma$: set $\{0, \ldots, n^{1/k} - 1\}$

**Defn:** $N^i(v)$: set of $n^{i/k}$ nodes closest to $v$.

**Defn:** $\sigma^i(\alpha\beta) = \alpha$ iff $|\alpha| = i$, for $\alpha, \beta \in \Sigma^*$

**Defn:** $\langle v \rangle$: node id of $v$ in base $n^{1/k}$, padded to length $k$.

Examples with 10000 nodes and $k = 4$:

- $\sigma^3(\langle 1234 \rangle) = \sigma^3(1234) = 123$
- $\sigma^3(\langle 3 \rangle) = \sigma^3(0003) = 000$

**Defn:** A block $B_\alpha$ is a set $\{v | \sigma^{k-1}(\langle v \rangle) = \alpha\}$

- node 1234 is in block $B_{123} = \{1230, \ldots, 1239\}$
Lemma 5. \( \exists \) sets of blocks \( S_v \) of size \( O(\log n) \) for nodes \( v \), so that 
\( \forall v \in G, \forall 0 \leq i < k, \forall \tau \in \Sigma^i, \exists w \in N^i(v) \) with \( B_\alpha \in S_w \) such that \( \sigma^i(B_\alpha) = \tau \).
Lemma 5. \exists sets of blocks $S_v$ of size $O(\log n)$ for nodes $v$, so that
\( \forall v \in G, \forall 0 \leq i < k, \forall \tau \in \Sigma^i, \exists w \in N^i(v) \text{ with } B_\alpha \in S_w \text{ such that } \sigma^i(B_\alpha) = \tau. \)

Example:
Lemma 5. \[ \exists \text{ sets of blocks } S_v \text{ of size } O(\log n) \text{ for nodes } v, \text{ so that } \]
\[ \forall v \in G, \forall 0 \leq i < k, \forall \tau \in \Sigma^i, \]
\[ \exists w \in N^i(v) \text{ with } B_\alpha \in S_w \text{ such that } \sigma^i(B_\alpha) = \tau. \]

Example:
- every \( N^1(v) \) contains all of \( \Sigma \)
Lemma 5. \( \exists \) sets of blocks \( S_v \) of size \( O(\log n) \) for nodes \( v \), so that \( \forall v \in G, \forall 0 \leq i < k, \forall \tau \in \Sigma^i \), \( \exists w \in N^i(v) \) with \( B_\alpha \in S_w \) such that \( \sigma^i(B_\alpha) = \tau \).

Example:

- every \( N^1(v) \) contains all of \( \Sigma \)
- every \( N^2(v) \) contains all of \( \Sigma^2 \)
- and so on
Lemma 5. $\exists$ sets of blocks $S_v$ of size $O(\log n)$ for nodes $v$, so that
$
\forall v \in G, \forall 0 \leq i < k, \forall \tau \in \Sigma^i,$
$\exists w \in N^i(v)$ with $B_\alpha \in S_w$ such that $\sigma^i(B_\alpha) = \tau$.

Example:
- every $N^1(v)$ contains all of $\Sigma$
- every $N^2(v)$ contains all of $\Sigma^2$
- and so on
Lemma 5. \( \exists \) sets of blocks \( S_v \) of size \( O(\log n) \) for nodes \( v \), so that \( \forall v \in G, \forall 0 \leq i < k, \forall \tau \in \Sigma^i, \exists w \in N^i(v) \) with \( B_\alpha \in S_w \) such that \( \sigma^i(B_\alpha) = \tau \).

Example:

- every \( N^1(v) \) contains all of \( \Sigma \)
- every \( N^2(v) \) contains all of \( \Sigma^2 \)
- and so on
Lemma 5. \( \exists \) sets of blocks \( S_v \) of size \( O(\log n) \) for nodes \( v \), so that \( \forall v \in G, \forall 0 \leq i < k, \forall \tau \in \Sigma^i, \exists w \in N^i(v) \) with \( B_\alpha \in S_w \) such that \( \sigma^i(B_\alpha) = \tau \).

Example:
- every \( N^1(v) \) contains all of \( \Sigma \)
- every \( N^2(v) \) contains all of \( \Sigma^2 \)
- and so on
Proof of \( \sqrt[k]{n} \)-Block Distribution Lemma

- By probabilistic method.
- \( f(n, k) \) rounds of “coloring”.
- \( \forall i, |\Sigma^i| = |N^i(u)| = n^i/k \).
- \( X_{i,u,\tau,r} \): event that in round \( r \) \( N^i(u) \) does not contain \( B_\alpha \) for which \( \sigma^i(B_\alpha) = \tau \).
Proof continued

Need to show: \[ \bigcup_{0 \leq i < k} \bigcup_{u \in V} \bigcup_{\tau \in \Sigma} \bigcap_{r=1}^{f(n,k)} X_{i,u,\tau,r} \neq \emptyset \]
Proof continued

Need to show: \( \bigcup_{0 \leq i < k} \bigcup_{u \in V} \bigcup_{\tau \in \Sigma} \bigcap_{r=1}^{f(n,k)} X_{i,u,\tau,r} \neq \emptyset \)

For fixed \( i, u, \tau, r \); \( \Pr[X_{i,u,\tau,r}] = \left( 1 - \frac{1}{n^{i/k}} \right)^{n^{i/k}} \leq e^{-1} \)
Proof continued

Need to show: \[ \bigcup_{0 \leq i < k} \bigcup_{u \in V} \bigcup_{\tau \in \Sigma} \bigcap_{r=1}^{f(n,k)} X_{i,u,\tau,r} \neq \emptyset \]

For fixed \( i, u, \tau, r \); \( \Pr[X_{i,u,\tau,r}] = \left(1 - \frac{1}{n^{i/k}}\right)^{n^{i/k}} \leq e^{-1} \)

For fixed \( i, u, \tau \); \( \Pr\left[\bigcap_{r=1}^{f(n,k)} X_{i,u,\tau,r}\right] \leq e^{-f(n,k)} \)
Proof continued

Need to show: \[ \bigcup_{0 \leq i < k} \bigcup_{u \in V} \bigcup_{\tau \in \Sigma^i} \bigcap_{r=1}^{f(n,k)} X_{i,u,\tau,r} \neq \emptyset \]

For fixed \( i, u, \tau, r \); \( \Pr[X_{i,u,\tau,r}] = \left(1 - \frac{1}{n^{i/k}}\right)^{n^{i/k}} \leq e^{-1} \)

For fixed \( i, u, \tau \); \( \Pr \left[ \bigcap_{r=1}^{f(n,k)} X_{i,u,\tau,r} \right] \leq e^{-f(n,k)} \)

For fixed \( i, u \); \( \Pr \left[ \bigcup_{\tau \in \Sigma^i} \bigcap_{r=1}^{f(n,k)} X_{i,u,\tau,r} \right] \leq n^{i/k} e^{-f(n,k)} \)
Proof continued

Need to show: \( \bigcup_{0 \leq i < k} \bigcup_{u \in V} \bigcup_{\tau \in \Sigma^i} \bigcap_{r=1}^{f(n,k)} X_{i,u,\tau,r} \neq \emptyset \)

For fixed \( i, u, \tau, r \); \( \Pr[X_{i,u,\tau,r}] = \left(1 - \frac{1}{n^{i/k}}\right)^{n^{i/k}} \leq e^{-1} \)

For fixed \( i, u, \tau \); \( \Pr\left[\bigcap_{r=1}^{f(n,k)} X_{i,u,\tau,r}\right] \leq e^{-f(n,k)} \)

For fixed \( i, u \); \( \Pr\left[\bigcup_{\tau \in \Sigma^i} \bigcap_{r=1}^{f(n,k)} X_{i,u,\tau,r}\right] \leq n^{i/k} e^{-f(n,k)} \)

For fixed \( i \);

\( \Pr\left[\bigcup_{u \in V} \bigcup_{\tau \in \Sigma^i} \bigcap_{r=1}^{f(n,k)} X_{i,u,\tau,r}\right] \leq n^{1+i/k} e^{-f(n,k)} \)
Proof continued

Need to show: \( \bigcup_{0 \leq i < k} \bigcup_{u \in V} \bigcup_{\tau \in \Sigma^i} \bigcap_{r=1}^{f(n,k)} X_{i,u,\tau,r} \neq \emptyset \)

For fixed \( i, u, \tau, r \); \( \Pr[X_{i,u,\tau,r}] = \left(1 - \frac{1}{n^{i/k}}\right)^{n^{i/k}} \leq e^{-1} \)

For fixed \( i, u, \tau \); \( \Pr\left[\bigcap_{r=1}^{f(n,k)} X_{i,u,\tau,r}\right] \leq e^{-f(n,k)} \)

For fixed \( i, u \); \( \Pr\left[\bigcup_{\tau \in \Sigma^i} \bigcap_{r=1}^{f(n,k)} X_{i,u,\tau,r}\right] \leq n^{i/k} e^{-f(n,k)} \)

For fixed \( i \);

\[
\Pr\left[\bigcup_{u \in V} \bigcup_{\tau \in \Sigma^i} \bigcap_{r=1}^{f(n,k)} X_{i,u,\tau,r}\right] \leq n^{1+i/k} e^{-f(n,k)} \leq n^2 e^{-f(n,k)}
\]
Clearly, \( \Pr \left[ \bigcup_{0 \leq i < k} \bigcup_{u \in V} \bigcup_{\tau \in \Sigma} \bigcap_{r=1}^{f(n,k)} X_{i,u,\tau,r} \right] \leq k n^2 e^{-f(n,k)} \)
Proof ends

Clearly, \( \Pr \left[ \bigcup_{0 \leq i < k} \bigcup_{u \in V} \bigcup_{\tau \in \Sigma} \bigcap_{r=1}^{f(n,k)} X_{i,u,\tau,r} \right] \leq kn^2 e^{-f(n,k)} \)

Result holds if \( kn^2 e^{-f(n,k)} < 1 \).

Ensured by choosing \( e^{f(n,k)} = 2kn^2 \),

\[
f(n, k) = \ln 2 + \ln k + 2 \ln n = \tilde{O}(1) \quad \square
\]
1. \textit{TZTab}(u)

2. \(\forall v \in N^1(u)\), the pair \((v, e_{uv})\), (first edge on a shortest path from \(u\) to \(v\)).

3. The set \(S'_u = S_u \cup \{B_\gamma\}\) (where \(u \in B_\gamma\)) of \(O(\log n)\) blocks \(B_\alpha\), and for each block \(B_\alpha \in S'_u\), the following:

   (a) \(\forall 0 \leq i < k - 1, \forall \tau \in \Sigma\), store \(TZR(u, v)\), (\(v\) is the nearest node containing a \(B_\beta\) such that
       - \(\sigma^i(B_\beta) = \sigma^i(B_\alpha)\), and
       - the \((i + 1)\) element of \(\sigma^{k-1}(B_\beta)\) is \(\tau\).

   (b) \(\forall \tau \in \Sigma\), store \(TZR(u, v)\),
       - \(v\) satisfies \(\sigma^{k-1}(B_\beta) = \sigma^{k-1}(v)\), and
       - the \(k^{th}\) element of \(\sigma^k(v)\) is \(\tau\).
if $n = 10000$ and $k = 4$, then $n^{1/k} = 10$, 4 digit names

node 2357 stores $O(\log n)$ blocks (including $B_{235}$),
Storage per block – by Example

- if \( n = 10000 \) and \( k = 4 \), then \( n^{1/k} = 10 \), 4 digit names

- node 2357 stores \( O(\log n) \) blocks (including \( B_{235} \)),

- for \( B_{235} \) store \( TZR(2357, v) \) for closest \( v \) containing \( B_\beta \) such that

\[
\begin{align*}
\star \sigma^1(B_\beta) &\in \{0, 1, \ldots, 9\} \\
\star \sigma^2(B_\beta) &\in \{20, 21, \ldots, 29\} \\
\star \sigma^3(B_\beta) &\in \{230, 231, \ldots, 239\} \\
\star \sigma^4(v) &\in \{2350, 2351, \ldots, 2359\}
\end{align*}
\]

**Lemma 6.** The storage requirement of our algorithm is \( \tilde{O}(n^{1/k}) \) for fixed \( k \).
Illustration of Exponential Algorithm

- if \( n = 10000 \) and \( k = 4 \), then \( n^{1/k} = 10 \)
Illustration of Exponential Algorithm

- if $n = 10000$ and $k = 4$, then $n^{1/k} = 10$
• if \( n = 10000 \) and \( k = 4 \), then \( n^{1/k} = 10 \)
• if $n = 10000$ and $k = 4$, then $n^{1/k} = 10$
• if $n = 10000$ and $k = 4$, then $n^{1/k} = 10$
Exponential Stretch Routing
Algorithm ExpRoute

if \((t \in N^1(s))\):
    route to \(t\) using \(e_{ut}\).
else:
    for \(i \leftarrow 0\) until \(v_i = t\) incrementing \(i\) by 1:
        if \((i + 1 < k)\): \(v_{i+1} \leftarrow\) closest to \(v_i\) in \(N^{i+1}(v_i) \cap \{v|\exists B_{\beta} \in S_v : \sigma^{i+1}(B_{\beta}) = \sigma^{i+1}(\langle t \rangle)\}\)
        else: \(v_k \leftarrow t\)
    if \((i = 0)\): route to \(v_1\) using \(e_{uv_1}\)
    else: \((i \geq 1)\) route to \(v_{i+1}\) using \(TZR(v_i, v_{i+1})\)
Lemma 7. Given a packet at $s$, Algorithm ExpRoute always delivers packets to the destination $t$.

Let $h$ – number of hops required for $v_h = t$.

Lemma 8. For $0 \leq i \leq h - 1$, $d(v_i, v_{i+1}) \leq 2^i d(s, t)$.

- The proof is by induction. Basis case trivial.
• $v_i$ is the $i^{th}$ node visited,

• $v_i^*$ is closest node to $s$ such that $\sigma^i(v_i^*) = \sigma^i(t)$. 

Exponential Stretch Analysis – Proof ctd
Exponential Stretch Analysis – Proof ctd

- $v_i$ is the $i^{th}$ node visited,
- $v_i^*$ is closest node to $s$ such that $\sigma^i(v_i^*) = \sigma^i(t)$.

\[
d(v_r, v_{r+1})
\]
Exponential Stretch Analysis – Proof ctd

- $v_i$ is the $i^{th}$ node visited,
- $v_i^*$ is closest node to $s$ such that $\sigma^i(v_i^*) = \sigma^i(t)$.

\[ d(v_r, v_{r+1}) \leq d(v_r, v_r^*) \]
Exponential Stretch Analysis – Proof ctd

- \( v_i \) is the \( i^{th} \) node visited,
- \( v_i^* \) is closest node to \( s \) such that \( \sigma^i(v_i^*) = \sigma^i(t) \).

\[
\begin{align*}
   d(v_r, v_{r+1}) & \leq d(v_r, v_r^*) \\
   & \leq d(v_r, s) + d(s, v_r^*) \\
   & \leq d(s, t) + \sum_{i=0}^{r-1} d(v_i, v_{i+1})
\end{align*}
\]
• $v_i$ is the $i^{th}$ node visited,

• $v_i^*$ is closest node to $s$ such that $\sigma^i(v_i^*) = \sigma^i(t)$. 

\[
d(v_r, v_{r+1}) \\
\leq d(v_r, v_{r+1}^*) \\
\leq d(v_r, s) + d(s, v_{r+1}^*) \\
\leq d(s, t) + \sum_{i=0}^{r-1} d(v_i, v_{i+1})
\]
Exponential Stretch Analysis – Proof ctd

- $v_i$ is the $i^{th}$ node visited,
- $v_i^*$ is closest node to $s$ such that $\sigma^i(v_i^*) = \sigma^i(t)$.

$$d(v_r, v_{r+1}) \leq d(v_r, v_{r+1}^*) \leq d(v_r, s) + d(s, v_{r+1}^*) \leq d(s, t) + \sum_{i=0}^{r-1} d(v_i, v_{i+1})$$
• $v_i$ is the $i^{th}$ node visited,
• $v_i^*$ is closest node to $s$ such that $\sigma^i(v_i^*) = \sigma^i(t)$.
• $v_i$ is the $i^{th}$ node visited,

• $v_i^*$ is closest node to $s$ such that $\sigma^i(v_i^*) = \sigma^i(t)$.

\[
d(v_r, v_{r+1}) \leq d(v_r, v_r^*) \\
\leq d(v_r, s) + d(s, v_r^*) \\
\leq d(s, t) + \sum_{i=0}^{r-1} d(v_i, v_{i+1}) \\
\leq d(s, t) \left[ 1 + \sum_{i=0}^{r-1} 2^i \right] \\
\leq 2^r d(s, t) \square
\]
Exponential Stretch Analysis – Proof ctd

$p'(u, v)$ – path from $u$ to $v$, using shortest distances between $v_i$ and $v_{i+1}$.

**Lemma 1.** *For all $s, t$, $p'(s, t) \leq (2^k - 1)d(s, t)$.***
$p'(u, v)$ – path from $u$ to $v$, using shortest distances between $v_i$ and $v_{i+1}$.

**Lemma 1.** For all $s, t$, $p'(s, t) \leq (2^k - 1)d(s, t)$.

**Theorem 1.** Algorithm ExpRoute is correct, uses space $\tilde{O}(n^{1/k})$, and gives stretch $1 + (2k - 1)(2^k - 2)$.

**Proof.**

- from $s = v_0$ to $v_1$, we use shortest path (stretch 1).
- remaining segments is $(2^k - 2) \times (2k - 1)$
Lemma 9. $\forall k \geq 1$, there is a universal name-independent routing algorithm with space $\tilde{O}\left(k^2 n^\frac{2}{k} \log D\right)$, stretch $16k^2 + 4k$, where $D$ is diameter of network.

Ingredients

- Underlying Name-dependent Optimal Tree Routing scheme (TZ or FG)
- Simple Distributed Dictionary
- Awerbuch and Peleg Covers
- Prefix Matching idea
Preliminaries

- $\hat{N}^m(v)$ neighborhood of nodes within distance $m$
- $Diam(G)$ maximum distance between any pair of nodes
- $Rad(v, G)$ maximum distance from any node to $v$
- $Rad(G)$ minimum radius of all nodes
- $Center(G)$ node with radius equals graph radius

Observations:

- Maximum Radius of any node is Diameter
- Graph Diameter is at most twice Graph Radius
Lemma 10. (Awerbuch and Peleg): Given a weighted graph and integers $i$ and $k$, a cover $C$ exists such that
Lemma 10. (Awerbuch and Peleg): Given a weighted graph and integers $i$ and $k$, a cover $C$ exists such that

- each $\hat{N}^{2^i}(v)$
Lemma 10. (Awerbuch and Peleg): Given a weighted graph and integers $i$ and $k$, a cover $C$ exists such that

- each $\hat{N}^{2^i}(v)$ is contained in a cluster $C$
- each cluster $C \in C$ has $\text{Diam}(C) \leq (4k + 1)2^i$
Lemma 10. (Awerbuch and Peleg): Given a weighted graph and integers \( i \) and \( k \), a cover \( C \) exists such that

- each \( \tilde{N}^{2^i}(v) \) is contained in a cluster \( C \)
- each cluster \( C \in C \) has \( \text{Diam}(C) \leq (4k + 1)2^i \)
- each node \( v \) is in at most \( kn^{1/k} \) clusters
Many levels of Covers

Construct a cover for each $i \in \{1, \ldots, \lfloor \log(Diam(G)) \rfloor \}$
Many levels of Covers

Construct a cover for each $i \in \{1, \ldots, \lceil \log (\text{Diam}(G)) \rceil \}$

Define neighborhoods of diameter $2^i$ at each level
Many levels of Covers

Construct a cover for each $i \in \{1, \ldots, \lceil \log (Diam(G)) \rceil \}$

Define neighborhoods of diameter $2^i$ at each level

Each node $v$ chooses “home” cluster $C_i(v)$ at each level
Routing within a cluster (Level \( i \))

Construct shortest path tree, source at \( \text{Center}(C) \),

\[ \text{root} \]
Routing within a cluster (Level \(i\))

Construct shortest path tree, source at \(Center(C)\), truncated by cluster.
Routing within a cluster (Level $i$)

Construct shortest path tree, source at $Center(C')$, truncated by cluster.

Use optimal tree routing (TZ01), space $\tilde{O}(1)$, headers $O(\log^2 n)$.

Recall $Diam(C') \leq (4k + 1)2^i$
Each vertex $u$ stores the following:

1. An identifier for $u$’s home cluster at level $i$.

2. For every cluster $C_i$ such that $u \in C_i$
   
   (a) $Tab(C_i, u)$
   
   (b) For every $\tau \in \Sigma$,
       for every $j = 0, \ldots, k - 1$,
       $R(C_i, v)$, where $v \in C_i$ is the nearest node such that $\sigma^j(\langle u \rangle) = \sigma^j(\langle v \rangle)$ and the $(j + 1)$ element of $v$ is $\tau$, if such node $v$ exists.

**Lemma 11.** The total space requirement is $\tilde{O}(k^2 n^{\frac{2}{k}} \log(Diam(G)))$.

Note: polynomial-sized weights imply $\tilde{O}\left(n^{\frac{2}{k}}\right)$ (for constant $k$).
High Level Routing Algorithm
Algorithm PolyRoute

Try each level $i \in \{1, \ldots, \lceil \log (Diam(G)) \rceil \}$

In each level route a star pattern

\[
\begin{align*}
\text{s}=v_0 & \quad \text{t}=v_4=1234
\end{align*}
\]
High Level Routing Algorithm
Algorithm PolyRoute

Try each level $i \in \{1, \ldots, \lceil \log (Diam(G)) \rceil \}$

In each level route a star pattern
High Level Routing Algorithm
Algorithm PolyRoute

Try each level $i \in \{1, \ldots, \lceil \log (Diam(G)) \rceil \}$

In each level route a star pattern
High Level Routing Algorithm
Algorithm PolyRoute

Try each level \( i \in \{1, \ldots, \lceil \log (Diam(G)) \rceil \} \)

In each level route a star pattern
High Level Routing Algorithm
Algorithm PolyRoute

Try each level \( i \in \{1, \ldots, \lceil \log(Diam(G)) \rceil \} \)

In each level route a star pattern
High Level Routing Algorithm
Algorithm PolyRoute

Try each level $i \in \{1, \ldots, \lceil \log (\text{Diam}(G)) \rceil \}$

In each level route a star pattern

\[ s = v_0 \]
\[ t = v_4 = 1234 \]
\[ v_1 = 1999 \]
\[ v_2 = 1299 \]
\[ v_3 = 1239 \]

Failure!!
High Level Routing Algorithm
Algorithm PolyRoute

Try each level $i \in \{1, \ldots, \lceil \log (Diam(G)) \rceil \}$

In each level route a star pattern

```
s=v0  t=v4=1234
v1=1999
v2=1299
v3=1239
```
High Level Routing Algorithm
Algorithm PolyRoute

Try each level $i \in \{1, \ldots, \lceil \log(Diam(G)) \rceil \}$

In each level route a star pattern
Try each level $i \in \{1, \ldots, \lceil \log (Diam(G)) \rceil \}$

In each level route a star pattern
High Level Routing Algorithm
Algorithm PolyRoute

Try each level \( i \in \{1, \ldots, \lceil \log(Diam(G)) \rceil \} \)

In each level route a star pattern

\[\begin{align*}
\text{s=v0} & \\
\text{v1=1999} & \\
\text{v2=1299} & \\
\text{v3=1239} & \\
\text{root} & \\
\text{t=v4=1234} & \\
\end{align*}\]
High Level Routing Algorithm
Algorithm PolyRoute

Try each level \( i \in \{1, \ldots, \lceil \log(Diam(G)) \rceil \} \)

In each level route a star pattern
High Level Routing Algorithm
Algorithm PolyRoute

Try each level $i \in \{1, \ldots, \lceil \log (Diam(G)) \rceil \}$

In each level route a star pattern
Algorithm PolyRoute

Stretch Analysis

- $d$ is distance to destination
- Destination is found at level $i \leq \log (2d)$
- Distance travelled in $C_i$ given by

$$
\leq Diam(C_i) \times k \\
\leq 2^i(4k + 1)k \\
\leq 2d(4k + 1)k \\
= (8k^2 + 2k)d
$$

- Stretch is $16k^2 + 4k$
Other Results

- Roundtrip Routing in Directed Graphs
  - Stretch 6 for $\tilde{O}(\sqrt{n})$ space, $O(\log n)$ headers
  - Stretch $k + 2^{k/2}(k + \epsilon)$ for $\tilde{O}(n^{2/k})$ space, $O(\log^2 n)$ headers
  - Stretch $16k^2 + 8k - 8$ for $\tilde{O}(n^{2/k})$ space, $O(\log^2 n)$ headers
Open Problems

- Proving Lower Bounds for Name Independent Model
  - close gap between 3 and 5

- Efficient Table Construction
  - Sequential
  - Distributed with space constraints

- Static Networks → Dynamic Networks

- Applications in Resource Location?
END
Questions ? ?? ???