Towards Communication-Efficient Peer-to-Peer Networks

Khalid Hourani[♦], **William K. Moses Jr.**[★], Gopal Pandurangan[♦]

[♦]University of Houston, [★]Durham University

Based on work presented at ESA 2024

November 1st, 2024

Outline

Context for the Work

- Peer-to-Peer (P2P) networks - many applications, much studied.

Context for the Work

- Peer-to-Peer (P2P) networks - many applications, much studied.
- Desirable properties of P2P networks:

Context for the Work

- Peer-to-Peer (P2P) networks - many applications, much studied.
- Desirable properties of P2P networks:
  - Connectivity
  - Low diameter
  - High expansion
  - Robustness to adversarial deletions

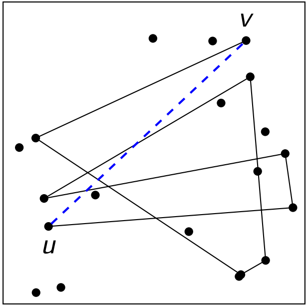
Context for the Work

- Peer-to-Peer (P2P) networks - many applications, much studied.
- Desirable properties of P2P networks:
  - Connectivity
  - Low diameter
  - High expansion
  - Robustness to adversarial deletions
- **As a result:** P2P network is typically a random graph, e.g., Bitcoin [11].

Context for the Work

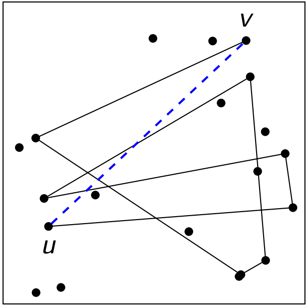
- Peer-to-Peer (P2P) networks - many applications, much studied.
- Desirable properties of P2P networks:
  - Connectivity
  - Low diameter
  - High expansion
  - Robustness to adversarial deletions
- **As a result:** P2P network is typically a random graph, e.g., Bitcoin [11].
- **To capture latency:** P2P network can be modeled as a random graph **embedded in an underlying Euclidean space** [5, 11].

The Issue



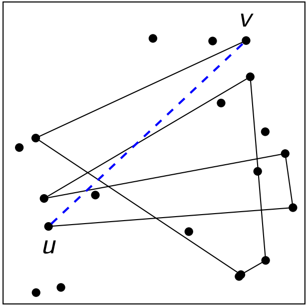
- **Pictured:** Such an embedding of P2P n/w in unit grid.
  Note: Only edges in shortest path from node u to node v shown with black lines.

The Issue



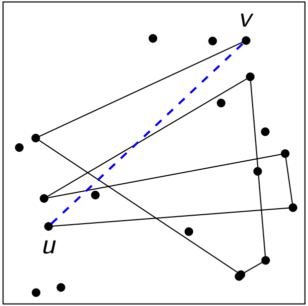
- **Pictured:** Such an embedding of P2P n/w in unit grid.
  Note: Only edges in shortest path from node u to node v shown with black lines.
- **What's the issue?**

The Issue



- **Pictured:** Such an embedding of P2P n/w in unit grid.
  Note: Only edges in shortest path from node u to node v shown with black lines.
- **What's the issue?**
  The network doesn't respect the underlying (Internet) communication topology:
  Shortest path distance between u & $v \gg$ Euclidean distance between u & v.

The Issue



- **Pictured:** Such an embedding of P2P n/w in unit grid.
  Note: Only edges in shortest path from node u to node v shown with black lines.
- **What's the issue?**
  The network doesn't respect the underlying (Internet) communication topology:
  Shortest path distance between u & $v \gg$ Euclidean distance between u & v.
- **The crux of our work:** We construct a n/w that respects this underlying metric.

What's Been Done

In General

- Distributed Hash Tables (DHTs): P2P network that respects underlying metric [1, 2, 3, 4, 6, 9].

What's Been Done

In General

- Distributed Hash Tables (DHTs): P2P network that respects underlying metric [1, 2, 3, 4, 6, 9].
  Issue: Requires global knowledge of costs b/w all pairs of nodes.

What's Been Done

In General

- Distributed Hash Tables (DHTs): P2P network that respects underlying metric [1, 2, 3, 4, 6, 9].
  Issue: Requires global knowledge of costs b/w all pairs of nodes.
- Converting arbitrary graph into an expander [7, 10, 12].

What's Been Done

In General

- Distributed Hash Tables (DHTs): P2P network that respects underlying metric [1, 2, 3, 4, 6, 9].
  Issue: Requires global knowledge of costs b/w all pairs of nodes.
- Converting arbitrary graph into an expander [7, 10, 12].
  Issue: Doesn't respect the underlying metric (i.e., geographic distance/latency b/w nodes).

What's Been Done

In General

- Distributed Hash Tables (DHTs): P2P network that respects underlying metric [1, 2, 3, 4, 6, 9].
  Issue: Requires global knowledge of costs b/w all pairs of nodes.
- Converting arbitrary graph into an expander [7, 10, 12].
  Issue: Doesn't respect the underlying metric (i.e., geographic distance/latency b/w nodes).

Closely Related Work

What's Been Done

In General

- Distributed Hash Tables (DHTs): P2P network that respects underlying metric [1, 2, 3, 4, 6, 9].
  Issue: Requires global knowledge of costs b/w all pairs of nodes.
- Converting arbitrary graph into an expander [7, 10, 12].
  Issue: Doesn't respect the underlying metric (i.e., geographic distance/latency b/w nodes).

Closely Related Work

- [5, 8] - empirical evidence that nodes on Internet can be embedded in low-dimensional metric space: distance b/w any 2 nodes accurately captures propagation delay b/w them.

What's Been Done

In General

- Distributed Hash Tables (DHTs): P2P network that respects underlying metric [1, 2, 3, 4, 6, 9].
  Issue: Requires global knowledge of costs b/w all pairs of nodes.
- Converting arbitrary graph into an expander [7, 10, 12].
  Issue: Doesn't respect the underlying metric (i.e., geographic distance/latency b/w nodes).

Closely Related Work

- [5, 8] - empirical evidence that nodes on Internet can be embedded in low-dimensional metric space: distance b/w any 2 nodes accurately captures propagation delay b/w them.
- [11] - P2P n/w construction protocol that respects underlying metric. Empirical evaluation.

Our Contributions

- Theoretical framework to model & analyze P2P n/w protocols.

Our Contributions

- Theoretical framework to model & analyze P2P n/w protocols.
- **Close-Weaver Protocol:** takes expander graph G, outputs topology G^* with good properties that also respects underlying metric.

Our Contributions

- Theoretical framework to model & analyze P2P n/w protocols.
- **Close-Weaver Protocol:** takes expander graph G, outputs topology G^* with good properties that also respects underlying metric.
    - **Fast** - $O(\text{polylog } n)$ rounds.

Our Contributions

- Theoretical framework to model & analyze P2P n/w protocols.
- **Close-Weaver Protocol:** takes expander graph G, outputs topology G^* with good properties that also respects underlying metric.
    - **Fast** - $O(\text{polylog } n)$ rounds.
    - **Requires only local knowledge** (nodes know their location & distance to nbrs in G).

Our Contributions

- Theoretical framework to model & analyze P2P n/w protocols.
- **Close-Weaver Protocol:** takes expander graph G, outputs topology G^* with good properties that also respects underlying metric.
  - **Fast** - $O(\text{polylog}\, n)$ rounds.
  - **Requires only local knowledge** (nodes know their location & distance to nbrs in G).
  - **Scalable** - nodes send & receive $O(\text{polylog}\, n)$ bits per round and communicate with $O(\text{polylog}\, n)$ nodes per round.

Our Contributions

- Theoretical framework to model & analyze P2P n/w protocols.
- **Close-Weaver Protocol:** takes expander graph G, outputs topology G^* with good properties that also respects underlying metric.
  - **Fast** - $O(\text{polylog}\, n)$ rounds.
  - **Requires only local knowledge** (nodes know their location & distance to nbrs in G).
  - **Scalable** - nodes send & receive $O(\text{polylog}\, n)$ bits per round and communicate with $O(\text{polylog}\, n)$ nodes per round.
  - **Robust** - no single point of action; no construction of tree-like structure.

Our Contributions

- Theoretical framework to model & analyze P2P n/w protocols.
- **Close-Weaver Protocol:** takes expander graph G, outputs topology G^* with good properties that also respects underlying metric.
  - **Fast** - $O(\text{polylog}\, n)$ rounds.
  - **Requires only local knowledge** (nodes know their location & distance to nbrs in G).
  - **Scalable** - nodes send & receive $O(\text{polylog}\, n)$ bits per round and communicate with $O(\text{polylog}\, n)$ nodes per round.
  - **Robust** - no single point of action; no construction of tree-like structure.
- Efficient routing & broadcast algorithms on G^*, including **Compass-Cast**.

Plan for Rest of the Talk

Plan for Rest of the Talk

1. Framework to understand results: Model + Metrics

Plan for Rest of the Talk

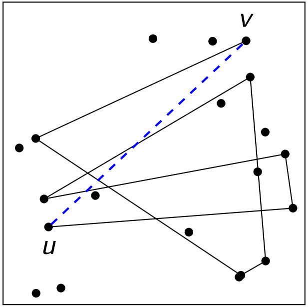
1. Framework to understand results: Model + Metrics
2. Construction of desired graph: Close-Weaver Protocol
   - Focus on alg. ideas
   - Mention of key points of analysis

Plan for Rest of the Talk

1. Framework to understand results: Model + Metrics
2. Construction of desired graph: Close-Weaver Protocol
   - Focus on alg. ideas
   - Mention of key points of analysis
3. Broadcast with optimal bounds (up to polylog n factors): Compass-Cast
   - Focus on alg. ideas

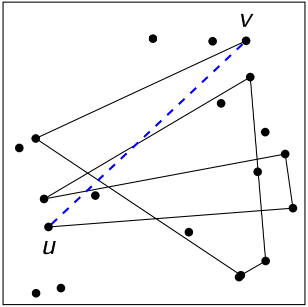
Outline

Model



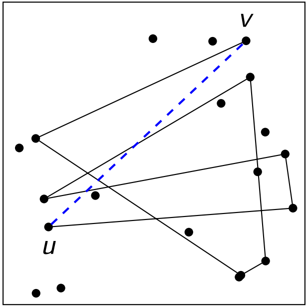
- Unit grid; n nodes embedded uniformly at random.

Model



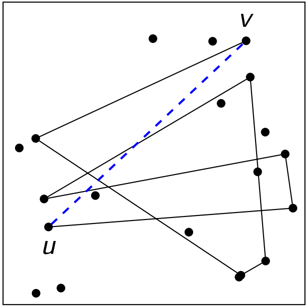
- Unit grid; n nodes embedded uniformly at random.
- I/p: $O(1)$-regular exander graph G (sufficient if G has random connectivity topology modeled by a bounded degree graph).

Model



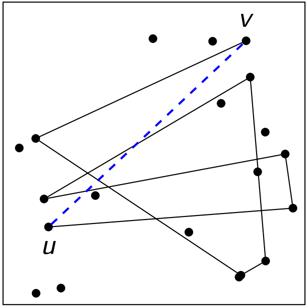
- Unit grid; n nodes embedded uniformly at random.
- I/p: $O(1)$-regular exander graph G (sufficient if G has random connectivity topology modeled by a bounded degree graph).
- Each node u knows its own ID, IDs of its nbrs in G, & value of n.

Model



- Unit grid; n nodes embedded uniformly at random.
- I/p: $O(1)$-regular exander graph G (sufficient if G has random connectivity topology modeled by a bounded degree graph).
- Each node u knows its own ID, IDs of its nbrs in G, & value of n.
- Given another node v's ID, a node u can determine its distance to v.

Model



- Unit grid; n nodes embedded uniformly at random.
- I/p: $O(1)$-regular exander graph G (sufficient if G has random connectivity topology modeled by a bounded degree graph).
- Each node u knows its own ID, IDs of its nbrs in G, & value of n.
- Given another node v's ID, a node u can determine its distance to v.
- Time - synchronous rounds; Communication - message passing

How to Measure Quality of Close-Weaver

- **Running time** - small ($O(\text{polylog } n)$).

How to Measure Quality of Close-Weaver

- **Running time** - small ($O(\text{polylog}\, n)$).
- **Max. degree of final topology** G^* - small ($O(\text{polylog}\, n)$).

How to Measure Quality of Close-Weaver

- **Running time** - small ($O(\text{polylog}\, n)$).
- **Max. degree of final topology** G^* - small ($O(\text{polylog}\, n)$).
- Does final topology G^* **respect underlying distance metric**?

How to Measure Quality of Close-Weaver

- **Running time** - small ($O(\text{polylog}\, n)$).
- **Max. degree of final topology** G^* - small ($O(\text{polylog}\, n)$).
- Does final topology G^* **respect underlying distance metric**?
  Rephrased: Does G^* contain a random geometric graph with good parameters?

How to Measure Quality of Close-Weaver

- **Running time** - small ($O(\text{polylog}\, n)$).
- **Max. degree of final topology** G^* - small ($O(\text{polylog}\, n)$).
- Does final topology G^* **respect underlying distance metric**?
  Rephrased: Does G^* contain a random geometric graph with good parameters?
  ($G(n, \rho)$ where $\rho = \Theta(\sqrt{\log n/n})$)

How to Measure Quality of Close-Weaver

- **Running time** - small ($O(\text{polylog}\, n)$).
- **Max. degree of final topology** G^* - small ($O(\text{polylog}\, n)$).
- Does final topology G^* **respect underlying distance metric**?
  Rephrased: Does G^* contain a random geometric graph with good parameters?
  ($G(n, \rho)$ where $\rho = \Theta(\sqrt{\log n / n})$)
- **Is final topology** G^* **useful**?

How to Measure Quality of Close-Weaver

- **Running time** - small ($O(\text{polylog}\, n)$).
- **Max. degree of final topology** G^* - small ($O(\text{polylog}\, n)$).
- Does final topology G^* **respect underlying distance metric**?
  Rephrased: Does G^* contain a random geometric graph with good parameters?
  ($G(n, \rho)$ where $\rho = \Theta(\sqrt{\log n / n})$)
- **Is final topology** G^* **useful**?
  Rephrased: Do we have efficient routing & broadcast algorithms that use G^*?

Outline

Algorithm

Algorithm Close-Weaver Construction Protocol

1: **for each** node u and phase i in $\{1, 2, \ldots, \kappa - 1\}$ **do**
2: u performs $\Theta(\log n)$ lazy random walks of length $2\tau = \Theta(\log n)$ in $G_u(i-1)$
3: u connects to $\Theta(\log n)$ nodes where random walks are successful
4: **for each** node u **do**
5: u initiates $\Theta(\log^2 n)$ lazy random walks in $B_u(r^\kappa)$ and connects to nodes where walk ends successfully

Important Notes:

- r - constant value $\in (0, 1)$, e.g., 0.25
- $B_u(r^i)$ – area of $\cap$ of unit square and the square of side-length r^i centered at node u
- κ chosen s.t. $r^{2\kappa} n = \Theta(\log n)$
- $G(i-1)$ - graph formed at end of phase $i-1$; $G(0) \equiv G$
  $G_u(i-1)$ - subgraph of $G(i-1)$ induced by all nodes & edges within $B_u(r^{i-1})$
- **Final graph** $G^* = \cup_{i=0}^{\kappa} G(i)$

Algorithm (Lines 2-3)

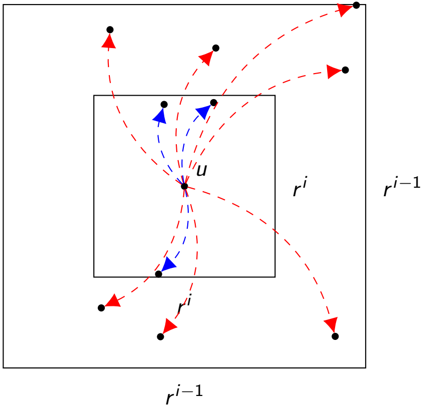


Figure: Node u initiates random walks within the square $B_u(r^{i-1})$. Random walks which terminate within square $B_u(r^i)$ are *successful* (shown in blue).

Analysis

Theorem

The Close-Weaver protocol takes an embedded d-regular expander graph and constructs a graph in $O(\log^3 n)$ rounds such that:
(i) its degree is $O(\log^2 n)$ with high probability and
(ii) it contains a random geometric graph $G(n, \rho)$ (where $\rho = \Theta(\sqrt{(\log n)/n})$) w.h.p.

Analysis

Theorem

The Close-Weaver protocol takes an embedded d-regular expander graph and constructs a graph in $O(\log^3 n)$ rounds such that:
(i) its degree is $O(\log^2 n)$ with high probability and
(ii) it contains a random geometric graph $G(n, \rho)$ (where $\rho = \Theta(\sqrt{(\log n)/n})$) w.h.p.

Proof Sketch

- **Lemma:** $\forall u, \forall 0 \leq i \leq \kappa - 1$, $G_u(i)$ is an expander w.h.p.

Analysis

Theorem

The Close-Weaver protocol takes an embedded d-regular expander graph and constructs a graph in $O(\log^3 n)$ rounds such that:
(i) its degree is $O(\log^2 n)$ with high probability and
(ii) it contains a random geometric graph $G(n, \rho)$ (where $\rho = \Theta(\sqrt{(\log n)/n})$) w.h.p.

Proof Sketch

- **Lemma:** $\forall u, \forall 0 \leq i \leq \kappa - 1$, $G_u(i)$ is an expander w.h.p.
- **Lemma:** In each phase, for each node, $\Theta(\log n)$ random walks are successfull w.h.p.

Analysis

Theorem

The Close-Weaver protocol takes an embedded d-regular expander graph and constructs a graph in $O(\log^3 n)$ rounds such that:
(i) its degree is $O(\log^2 n)$ with high probability and
(ii) it contains a random geometric graph $G(n, \rho)$ (where $\rho = \Theta(\sqrt{(\log n)/n})$) w.h.p.

Proof Sketch

- **Lemma:** $\forall u, \forall 0 \leq i \leq \kappa - 1$, $G_u(i)$ is an expander w.h.p.
- **Lemma:** In each phase, for each node, $\Theta(\log n)$ random walks are successfull w.h.p.
- **Lemma:** In each phase, for each node u, $\#$ of nodes that connect to u is $\Theta(\log n)$ w.h.p.

Analysis

Theorem

The Close-Weaver protocol takes an embedded d-regular expander graph and constructs a graph in $O(\log^3 n)$ rounds such that:
(i) its degree is $O(\log^2 n)$ with high probability and
(ii) it contains a random geometric graph $G(n, \rho)$ (where $\rho = \Theta(\sqrt{(\log n)/n})$) w.h.p.

Proof Sketch

- **Lemma:** $\forall u, \forall 0 \leq i \leq \kappa - 1$, $G_u(i)$ is an expander w.h.p.
- **Lemma:** In each phase, for each node, $\Theta(\log n)$ random walks are successfull w.h.p.
- **Lemma:** In each phase, for each node u, # of nodes that connect to u is $\Theta(\log n)$ w.h.p.
- **Lemma:** For each node u, u connected to all $\Theta(\log n)$ nbrs in $B_u(r^\kappa)$ in $G_u(\kappa)$

Analysis

Theorem

The Close-Weaver protocol takes an embedded d-regular expander graph and constructs a graph in $O(\log^3 n)$ rounds such that:
(i) its degree is $O(\log^2 n)$ with high probability and
(ii) it contains a random geometric graph $G(n, \rho)$ (where $\rho = \Theta(\sqrt{(\log n)/n})$) w.h.p.

Proof Sketch

- **Lemma:** $\forall u, \forall 0 \leq i \leq \kappa - 1$, $G_u(i)$ is an expander w.h.p.
- **Lemma:** In each phase, for each node, $\Theta(\log n)$ random walks are successfull w.h.p.
- **Lemma:** In each phase, for each node u, $\#$ of nodes that connect to u is $\Theta(\log n)$ w.h.p.
- **Lemma:** For each node u, u connected to all $\Theta(\log n)$ nbrs in $B_u(r^\kappa)$ in $G_u(\kappa)$
  Random geometric graph: So $G(\kappa)$ is $G(n, \rho)$ with desired ρ

Analysis

Theorem

The Close-Weaver protocol takes an embedded d-regular expander graph and constructs a graph in $O(\log^3 n)$ rounds such that:
(i) its degree is $O(\log^2 n)$ with high probability and
(ii) it contains a random geometric graph $G(n, \rho)$ (where $\rho = \Theta(\sqrt{(\log n)/n})$) w.h.p.

Proof Sketch

- **Lemma:** $\forall u, \forall 0 \leq i \leq \kappa - 1$, $G_u(i)$ is an expander w.h.p.
- **Lemma:** In each phase, for each node, $\Theta(\log n)$ random walks are successfull w.h.p.
- **Lemma:** In each phase, for each node u, # of nodes that connect to u is $\Theta(\log n)$ w.h.p.
- **Lemma:** For each node u, u connected to all $\Theta(\log n)$ nbrs in $B_u(r^\kappa)$ in $G_u(\kappa)$
  Random geometric graph: So $G(\kappa)$ is $G(n, \rho)$ with desired ρ
- $O(\log^2 n)$ **max. degree** - $O(\log n)$ phases, each phase adds $O(\log n)$ edges per node

Analysis

Theorem

The Close-Weaver protocol takes an embedded d-regular expander graph and constructs a graph in $O(\log^3 n)$ rounds such that:
(i) its degree is $O(\log^2 n)$ with high probability and
(ii) it contains a random geometric graph $G(n, \rho)$ (where $\rho = \Theta(\sqrt{(\log n)/n})$) w.h.p.

Proof Sketch

- **Lemma:** $\forall u, \forall 0 \leq i \leq \kappa - 1$, $G_u(i)$ is an expander w.h.p.
- **Lemma:** In each phase, for each node, $\Theta(\log n)$ random walks are successfull w.h.p.
- **Lemma:** In each phase, for each node u, $\#$ of nodes that connect to u is $\Theta(\log n)$ w.h.p.
- **Lemma:** For each node u, u connected to all $\Theta(\log n)$ nbrs in $B_u(r^\kappa)$ in $G_u(\kappa)$
  Random geometric graph: So $G(\kappa)$ is $G(n, \rho)$ with desired ρ
- $O(\log^2 n)$ **max. degree** - $O(\log n)$ phases, each phase adds $O(\log n)$ edges per node
- **Running time** from another lemma: time required to complete random walks

Outline

Overview

- **Goal:** Broadcast from source to all nodes
  1. Fast - $O(1)$ distance to reach any node
  2. Small # of hops - $O(\log n)$ hops
  3. Efficient (low congestion) - sum of weights of edges used - $O(\sqrt{n \log^3 n})$

Overview

- **Goal:** Broadcast from source to all nodes
  1. Fast - $O(1)$ distance to reach any node
  2. Small # of hops - $O(\log n)$ hops
  3. Efficient (low congestion) - sum of weights of edges used - $O(\sqrt{n \log^3 n})$
- **Simple alg:** Broadcast along G^*

Overview

- **Goal:** Broadcast from source to all nodes
  1. Fast - $O(1)$ distance to reach any node
  2. Small # of hops - $O(\log n)$ hops
  3. Efficient (low congestion) - sum of weights of edges used - $O(\sqrt{n \log^3 n})$
- **Simple alg:** Broadcast along G^*
- **Issue:** Too many edges used/congestion of network

Overview

- **Goal:** Broadcast from source to all nodes
  1. Fast - $O(1)$ distance to reach any node
  2. Small # of hops - $O(\log n)$ hops
  3. Efficient (low congestion) - sum of weights of edges used - $O(\sqrt{n \log^3 n})$
- **Simple alg:** Broadcast along G^*
- **Issue:** Too many edges used/congestion of network
- **Compass-Cast:** Solves broadcast & given 3 metrics are optimal up to polylog n factors

Overview

- **Goal:** Broadcast from source to all nodes
  1. Fast - $O(1)$ distance to reach any node
  2. Small # of hops - $O(\log n)$ hops
  3. Efficient (low congestion) - sum of weights of edges used - $O(\sqrt{n \log^3 n})$
- **Simple alg:** Broadcast along G^*
- **Issue:** Too many edges used/congestion of network
- **Compass-Cast:** Solves broadcast & given 3 metrics are optimal up to polylog n factors
- **Additional requirement:** Nodes need to know their own location (& one round additional preprocessing after creating G^*).

Overview

- **Goal:** Broadcast from source to all nodes
  1. Fast - $O(1)$ distance to reach any node
  2. Small # of hops - $O(\log n)$ hops
  3. Efficient (low congestion) - sum of weights of edges used - $O(\sqrt{n \log^3 n})$
- **Simple alg:** Broadcast along G^*
- **Issue:** Too many edges used/congestion of network
- **Compass-Cast:** Solves broadcast & given 3 metrics are optimal up to polylog n factors
- **Additional requirement:** Nodes need to know their own location (& one round additional preprocessing after creating G^*).

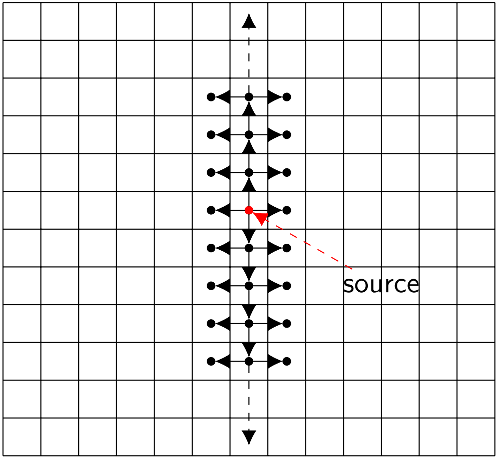
- Define H_i: partition of unit grid into $1/r^i \times 1/r^i$ grid
- **Idea:** Propagate the message to one node in each square of H_κ, then use $G(\kappa)$ (random geometric graph) to transmit messages twice.

Compass-Cast: 3 Phase Algorithm

- **Phase 1:** msg from source propagated to at least one node in each square of H_2
- **Phase 2:** Recursively perform: Node u in square of H_i fwds msg to one node per square of H_{i+1} that intersects with u's square
- **Phase 3:** All nodes with msg transmit on $G(\kappa)$ (random geometric graph) twice

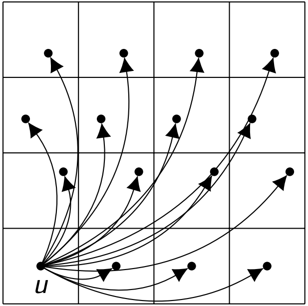
Compass-Cast: 3 Phase Algorithm

- **Phase 1:** msg from source propagated to at least one node in each square of H_2
- **Phase 2:** Recursively perform: Node u in square of H_i fwds msg to one node per square of H_{i+1} that intersects with u's square
- **Phase 3:** All nodes with msg transmit on $G(\kappa)$ (random geometric graph) twice



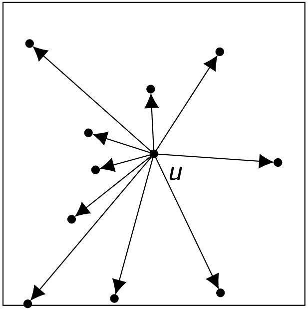
Compass-Cast: 3 Phase Algorithm

- **Phase 1:** msg from source propagated to at least one node in each square of H_2
- **Phase 2:** Recursively perform: Node u in square of H_i fwds msg to one node per square of H_{i+1} that intersects with u's square
- **Phase 3:** All nodes with msg transmit on $G(\kappa)$ (random geometric graph) twice



Compass-Cast: 3 Phase Algorithm

- **Phase 1:** msg from source propagated to at least one node in each square of H_2
- **Phase 2:** Recursively perform: Node u in square of H_i fwds msg to one node per square of H_{i+1} that intersects with u's square
- **Phase 3:** All nodes with msg transmit on $G(\kappa)$ (random geometric graph) twice



Outline

Conclusions & Future Work

Conclusions

1. In this talk, we looked at Close-Weaver (alg + analysis) & Compass-Broadcast (alg).
2. In paper, you can find more details about those two, as well as on Geometric-Flooding and Greedy-Routing.

Future Work

1. Reducing the max. degree of any node in the constructed network (while still allowing efficient broadcast & routing).
2. Possibly removing/weakening the assumption that all nodes are distributed in the grid uniformly at random.

References I

C. Greg Plaxton, Rajmohan Rajaraman, and Andréa W. Richa.
Accessing nearby copies of replicated objects in a distributed environment.
Theory Comput. Syst., 32(3):241–280, 1999.

Antony Rowstron and Peter Druschel.
Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems.
In *Middleware 2001: IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg, Germany, November 12–16, 2001 Proceedings 2*, pages 329–350. Springer, 2001.

David R. Karger and Matthias Ruhl.
Finding nearest neighbors in growth-restricted metrics.
In John H. Reif, editor, *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 741–750. ACM, 2002.

Ittai Abraham, Dahlia Malkhi, and Oren Dobzinski.
LAND: stretch $(1 + \text{epsilon})$ locality-aware networks for dhts.
In J. Ian Munro, editor, *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, New Orleans, Louisiana, USA, January 11-14, 2004*, pages 550–559. SIAM, 2004.

References II

Frank Dabek, Russ Cox, M. Frans Kaashoek, and Robert Tappan Morris.
Vivaldi: a decentralized network coordinate system.
In Raj Yavatkar, Ellen W. Zegura, and Jennifer Rexford, editors, *ACM SIGCOMM*, pages 15–26, 2004.

Kirsten Hildrum, John Kubiatowicz, Sean Ma, and Satish Rao.
A note on the nearest neighbor in growth-restricted metrics.
In J. Ian Munro, editor, *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, New Orleans, Louisiana, USA, January 11-14, 2004*, pages 560–561. SIAM, 2004.

Dana Angluin, James Aspnes, Jiang Chen, Yinghua Wu, and Yitong Yin.
Fast construction of overlay networks.
In Phillip B. Gibbons and Paul G. Spirakis, editors, *SPAA 2005: Proceedings of the 17th Annual ACM Symposium on Parallelism in Algorithms and Architectures, July 18-20, 2005, Las Vegas, Nevada, USA*, pages 145–154. ACM, 2005.

Zi Hu, John S. Heidemann, and Yuri Pradkin.
Towards geolocation of millions of IP addresses.
In *Proceedings of the 12th ACM SIGCOMM Internet Measurement Conference, IMC*, pages 123–130, 2012.

References III

Maximilian Drees, Robert Gmyr, and Christian Scheideler.
Churn- and dos-resistant overlay networks based on network reconfiguration.
In Christian Scheideler and Seth Gilbert, editors, *Proceedings of the 28th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA 2016, Asilomar State Beach/Pacific Grove, CA, USA, July 11-13, 2016*, pages 417–427. ACM, 2016.

Seth Gilbert, Gopal Pandurangan, Peter Robinson, and Amitabh Trehan.
Dconstructor: Efficient and robust network construction with polylogarithmic overhead.
In Yuval Emek and Christian Cachin, editors, *PODC '20: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, August 3-7, 2020*, pages 438–447. ACM, 2020.

Yifan Mao, Soubhik Deb, Shaileshh Bojja Venkatakrishnan, Sreeram Kannan, and Kannan Srinivasan.
Perigee: Efficient peer-to-peer network design for blockchains.
In *ACM Symposium on Principles of Distributed Computing (PODC)*, pages 428–437, 2020.

Thorsten Götte, Kristian Hinnenthal, Christian Scheideler, and Julian Werthmann.
Time-optimal construction of overlay networks.
In Avery Miller, Keren Censor-Hillel, and Janne H. Korhonen, editors, *ACM Symposium on Principles of Distributed Computing (PODC)*, pages 457–468, 2021.

The End