Geographic Routing: GPSR

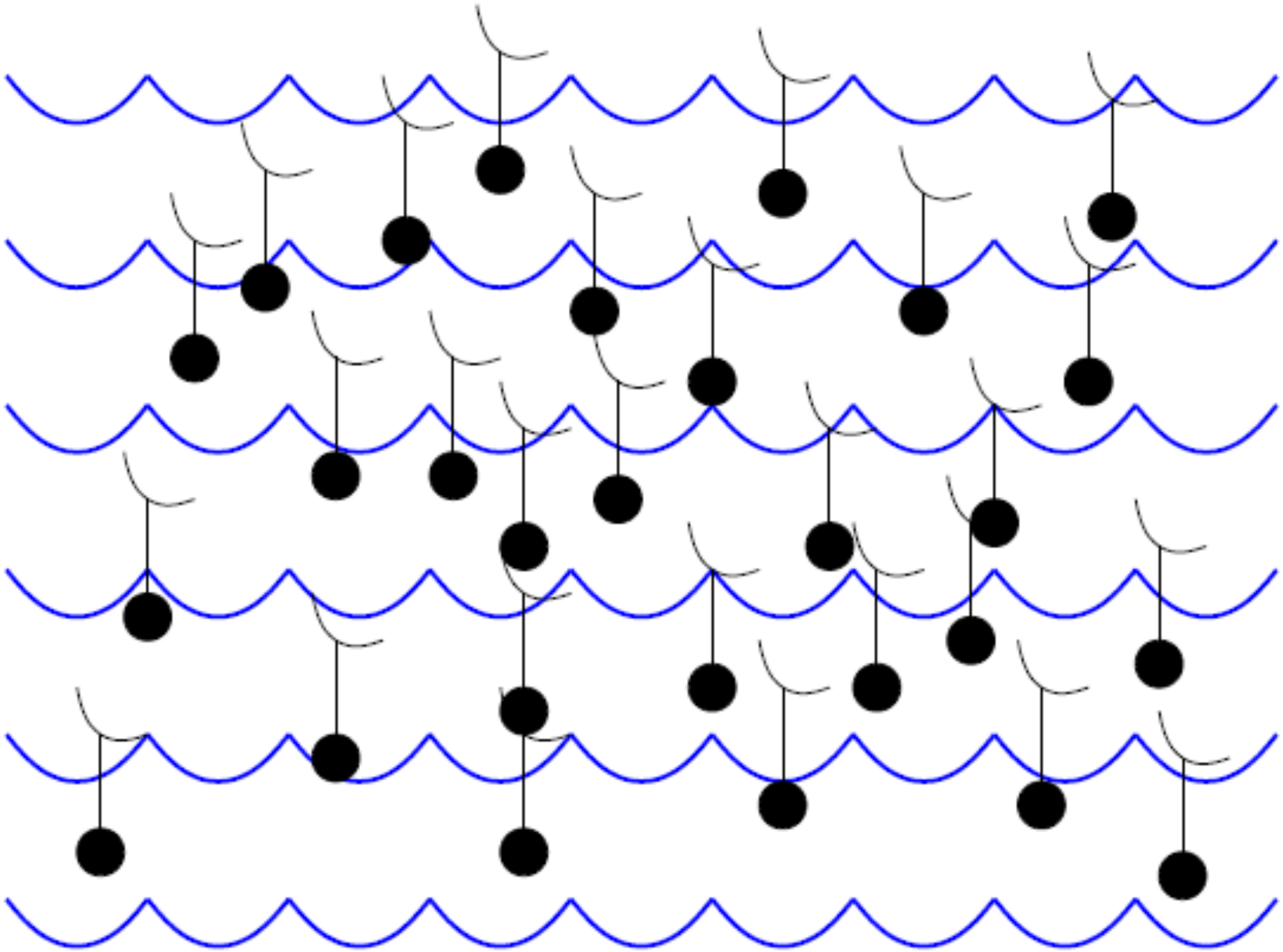
Brad Karp
UCL Computer Science

CS M038 / GZ06
19th January, 2011

Context: Ad hoc Routing

- Early 90s: availability of off-the-shelf wireless network cards and laptops
- 1994: first papers on Destination-Sequenced Distance Vector (DSDV) routing and Dynamic Source Routing (DSR) spark *tremendous* interest in routing on mobile wireless (ad hoc) networks
- 1998: Broch et al.'s comparison of leading ad hoc routing protocol proposals in ns-2 simulator in MobiCom
- [2000: GPSR in MobiCom]
- 2000: Estrin et al.'s Directed Diffusion in MobiCom sparks interest in wireless sensor networks

Original Motivation (2000):
Mobile Sensornets



Original Motivation (2000): Rooftop Networks

- Potentially lower-cost alternative to cellular architecture (no backhaul to every base station)

Motivation (2009): Sensornets

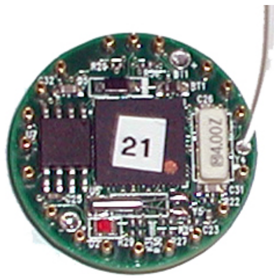
- Many sensors, widely dispersed
- Sensor: radio, transducer(s), CPU, storage, battery
- Multiple wireless hops, forwarding sensor-to-sensor to a base station

What communication primitives will thousand- or million-node sensornets need?

“Scalability” in Sensor Networks

- Resource constraints drive metrics

- State per node: minimize
- Energy consumed: minimize
- Bandwidth consumed: minimize



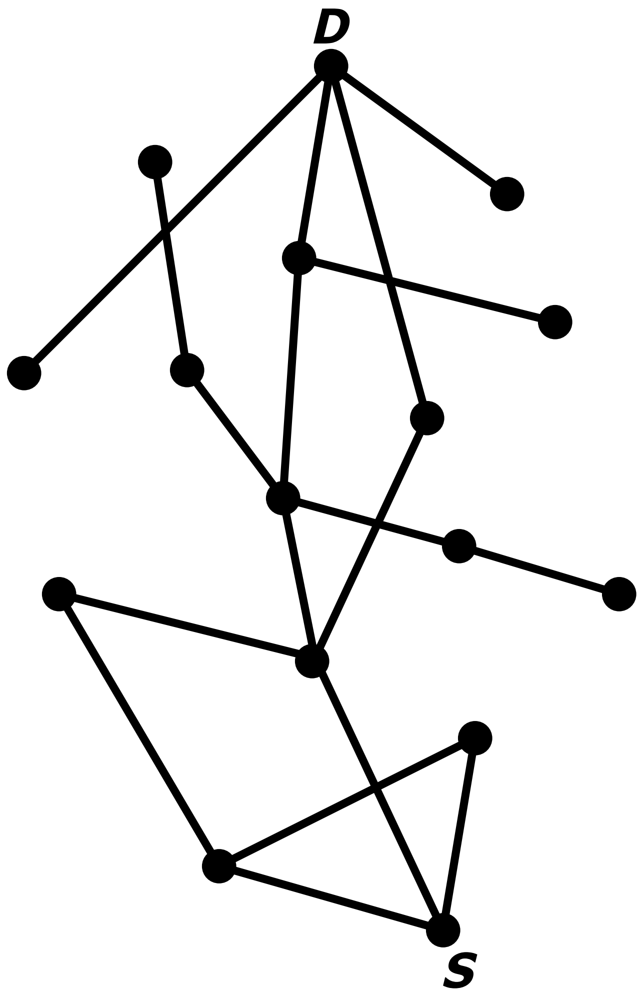
- System scale in nodes: maximize
- Operation success rate: maximize

Outline

- Motivation
- Context
- Algorithm
  - Greedy forwarding
  - Graph planarization
  - Perimeter forwarding
- Evaluation in simulation
- Footnotes
  - Open questions
  - Foibles of simulation

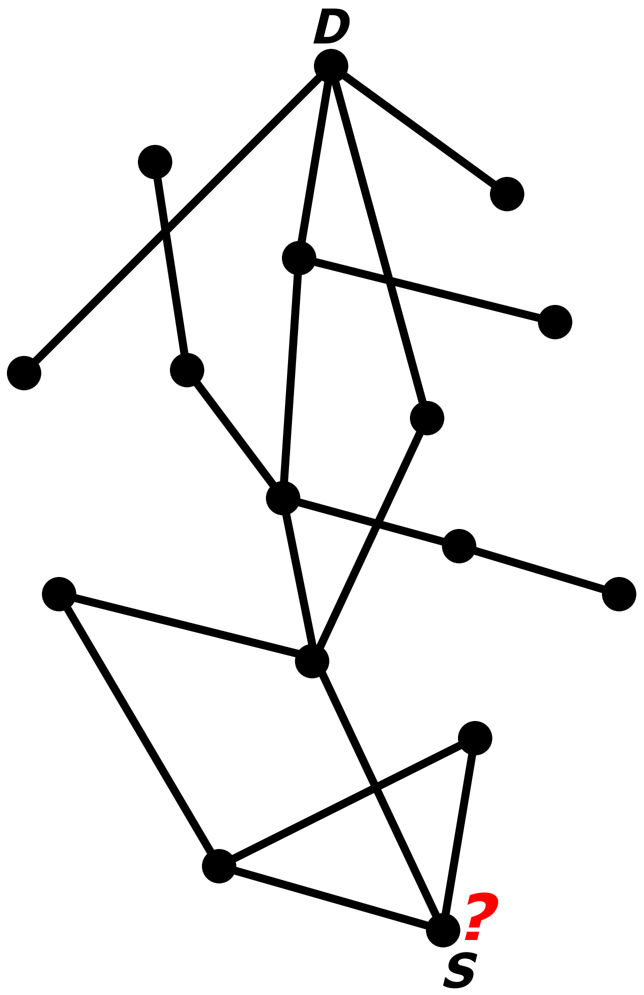
The Routing Problem

- Each router has unique ID
- Packets stamped with destination node ID
- Router must choose next hop for received packet
- Routers communicate to accumulate state for use in forwarding decisions
- Routes change with topology
- Evaluation metrics:
  - Routing protocol message cost
  - Data delivery success rate
  - Route length (hops)
  - Per-router state



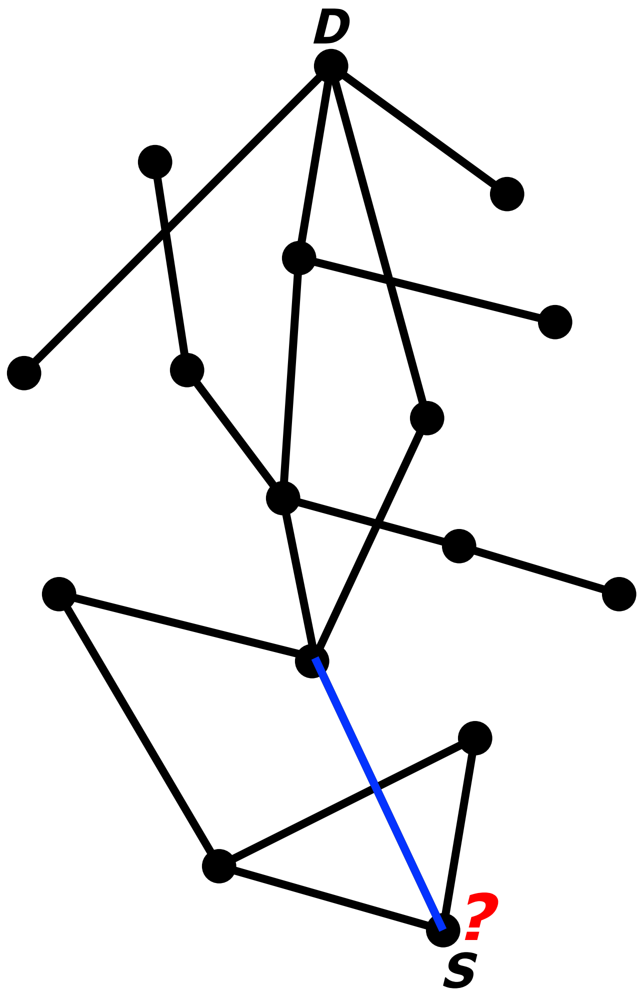
The Routing Problem

- Each router has unique ID
- Packets stamped with destination node ID
- Router must choose next hop for received packet
- Routers communicate to accumulate state for use in forwarding decisions
- Routes change with topology
- Evaluation metrics:
  - Routing protocol message cost
  - Data delivery success rate
  - Route length (hops)
  - Per-router state



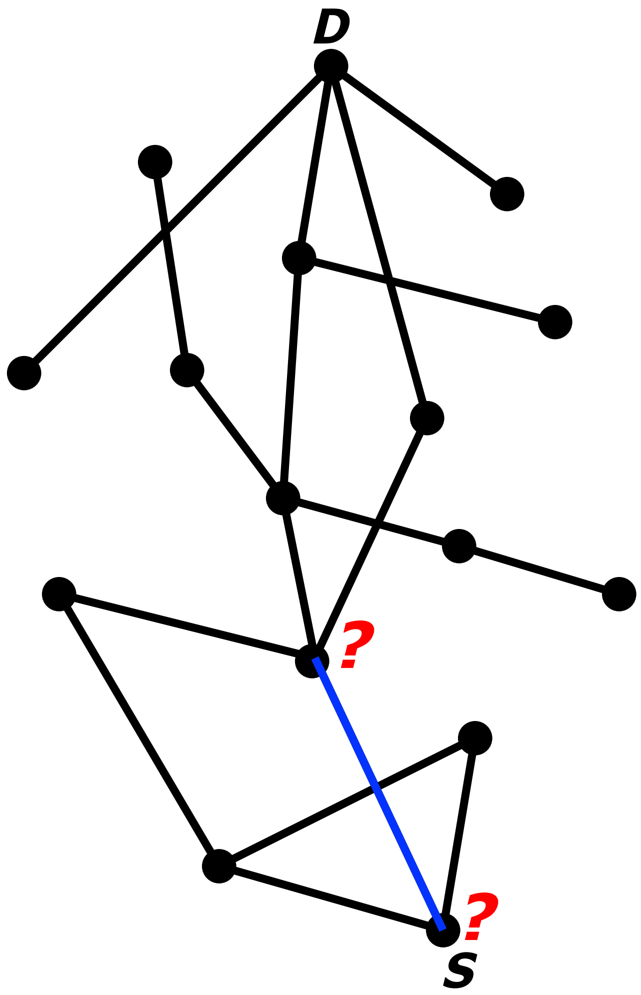
The Routing Problem

- Each router has unique ID
- Packets stamped with destination node ID
- Router must choose next hop for received packet
- Routers communicate to accumulate state for use in forwarding decisions
- Routes change with topology
- Evaluation metrics:
  - Routing protocol message cost
  - Data delivery success rate
  - Route length (hops)
  - Per-router state



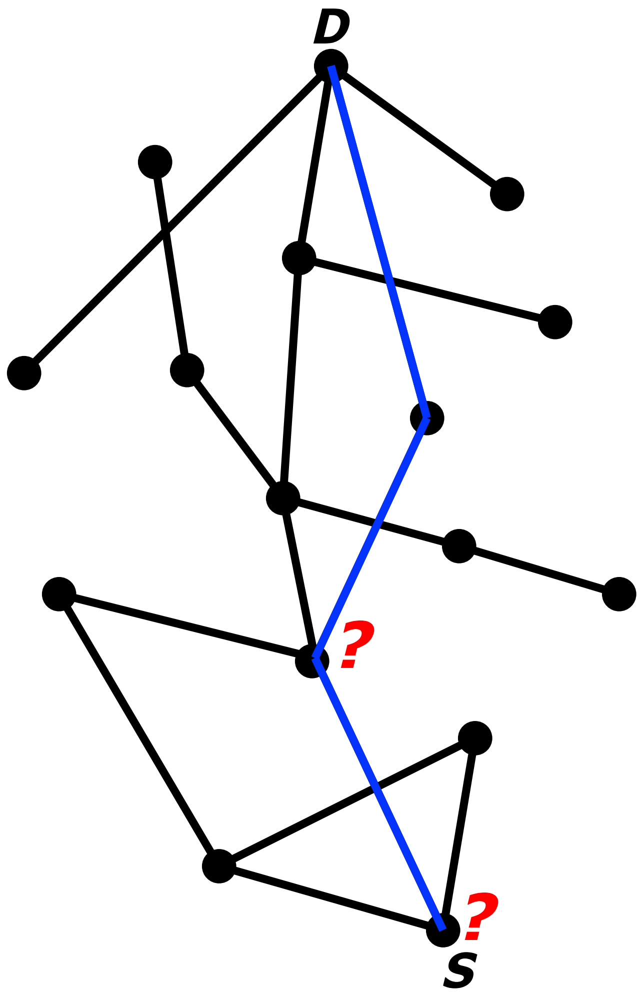
The Routing Problem

- Each router has unique ID
- Packets stamped with destination node ID
- Router must choose next hop for received packet
- Routers communicate to accumulate state for use in forwarding decisions
- Routes change with topology
- Evaluation metrics:
  - Routing protocol message cost
  - Data delivery success rate
  - Route length (hops)
  - Per-router state



The Routing Problem

- Each router has unique ID
- Packets stamped with destination node ID
- Router must choose next hop for received packet
- Routers communicate to accumulate state for use in forwarding decisions
- Routes change with topology
- Evaluation metrics:
  - Routing protocol message cost
  - Data delivery success rate
  - Route length (hops)
  - Per-router state



Why Are Topologies Dynamic?

- Node failure
  - Battery depletion
  - Hardware malfunction
  - Physical damage (harsh environment)
- Link failure
  - Changing RF interference sources
  - Mobile obstacles change multi-path fading
- Node mobility
  - In-range neighbor set constantly changing
  - Extreme case for routing scalability
  - Not commonly envisioned for sensor networks

Routing: Past Approaches, Scaling

- Wired, Intra-domain Internet routing:
  - Link-state and Distance-vector: shortest paths in hops
  - LS: push full topology map to all routers, O(L) state
  - DV: push distances across network diameter, O(N) state
  - Each link change must be communicated to all routers, or loops/disconnection result [Zaumen, Garcia-Luna, '91]

- Dynamic Source Routing (DSR), ad hoc routing:
  - Flood queries on-demand to learn source routes
  - Cache replies

Scaling Routing (cont'd)

- Dominant factors in cost of DV, LS, DSR:
  - rate of change of topology (bandwidth)
  - number of routers in routing domain (b/w, state)
- Scaling strategies:
  - Hierarchy: at AS boundaries (BGP) or on finer scale (OSPF)
    - Goal: reduce number of routers in routing domain
    - Assumption: address aggregation
  - Caching: store source routes overheard (DSR)
    - Goal: limit propagation of future queries
    - Assumption: source route remains fixed while cached

Scaling Routing (cont'd)

- Dominant factors in cost of DV, LS, DSR:
  - rate of change of topology (bandwidth)
  - number of routers in routing domain (b/w, state)

Today: Internet routing scales because of **IP prefix aggregation;** **not easily applicable** in sensornets

Can we achieve per-node state independent of N?

Can we reduce bandwidth spent communicating topology changes?

Greedy Perimeter Stateless Routing (GPSR)

Central idea: Machines can know their geographic locations.
Route using **geography**.

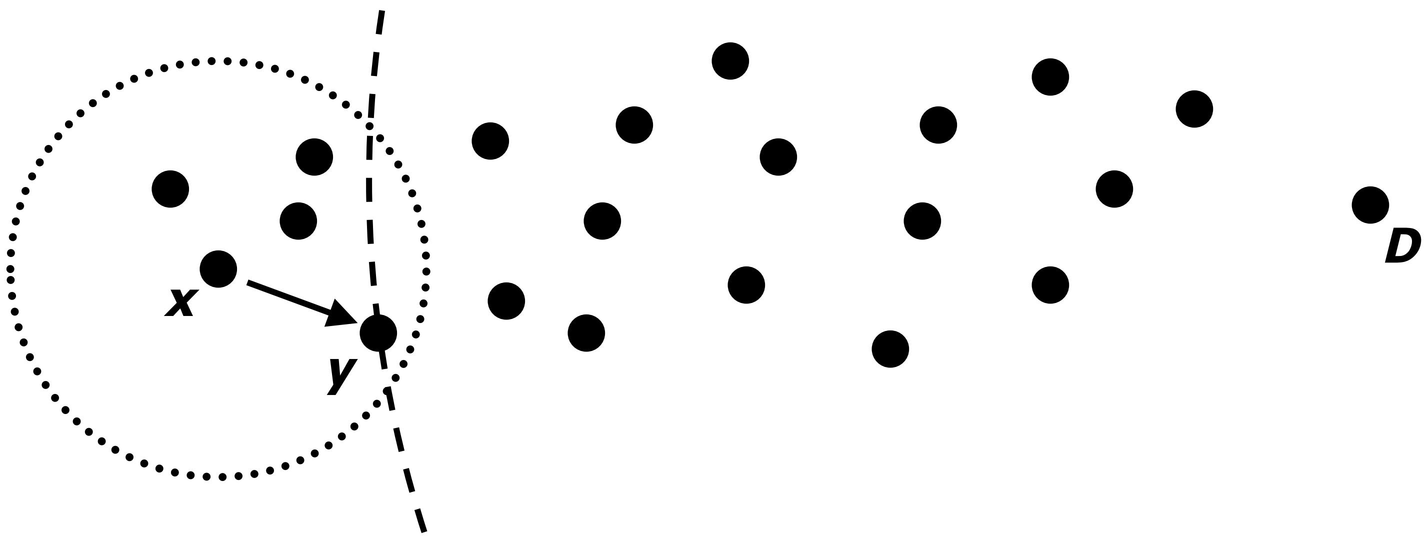
- Packet destination field: location of destination
- Nodes all know own positions, *e.g.,*
  - by GPS (outdoors)
  - by surveyed position (for non-mobile nodes)
  - by short-range localization (indoors, [AT&T Camb, 1997], [Priyantha *et al.,* 2000])
  - *&c.*
- Assume an efficient node location registration/ lookup system (e.g., GLS [Li et al., 2000]) to support host-centric addressing

Assumptions

- Bi-directional radio links (unidirectional links may be *blacklisted*)
- Network nodes placed roughly in a plane
- Radio propagation in free space; distance from transmitter determines signal strength at receiver
- Fixed, uniform radio transmitter power

Greedy Forwarding

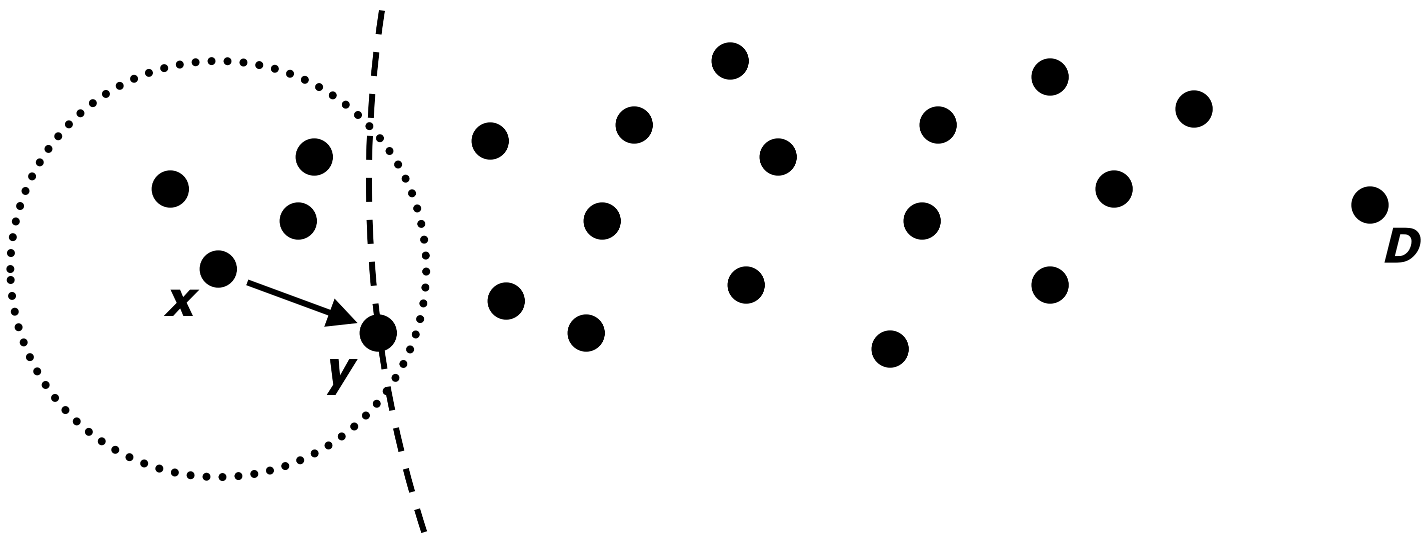
- Nodes learn immediate neighbors' positions from beaconing/piggybacking on data packets
- Locally optimal, greedy next hop choice:
  - Neighbor geographically nearest destination



Greedy Forwarding

- Nodes learn immediate neighbors' positions from beaconing/piggybacking on data packets

Neighbor must be **strictly closer** to avoid loops

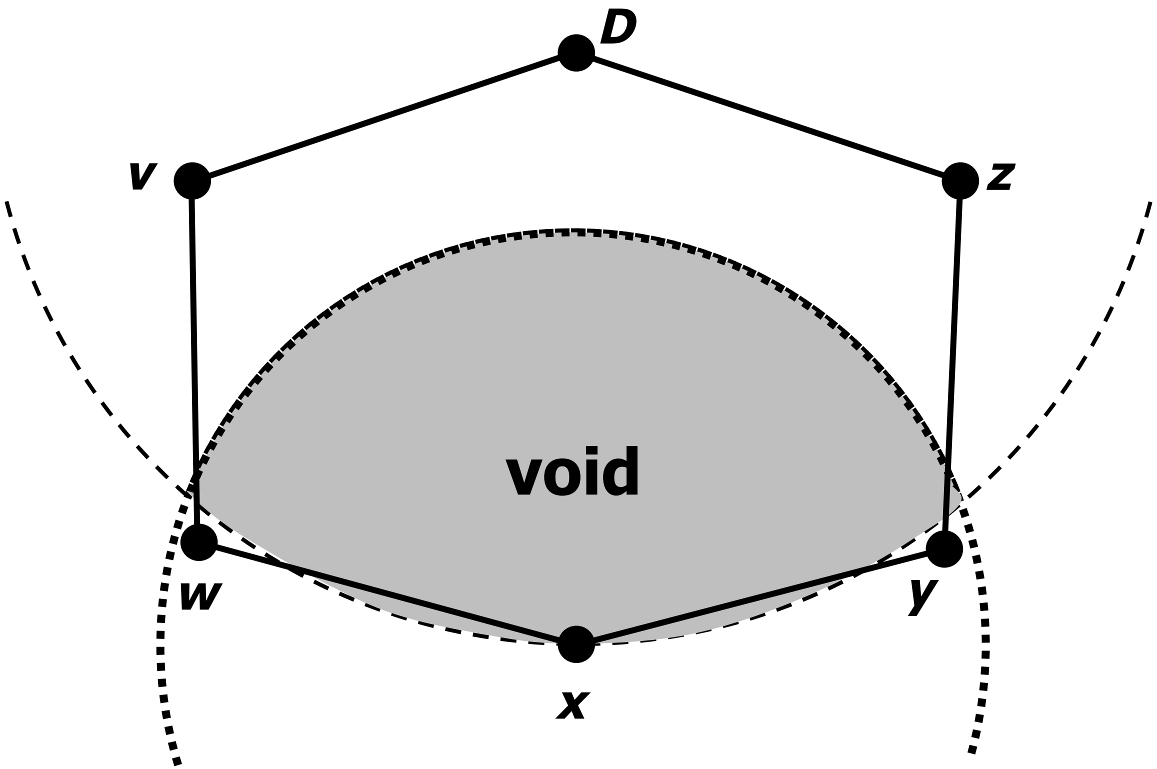


In Praise of Geography

- Self-describing
- As node density increases, shortest path tends toward Euclidean straight line between source and destination
- Node's state concerns only one-hop neighbors:
  - Low per-node state: O(density)
  - Low routing protocol overhead: state pushed only one hop

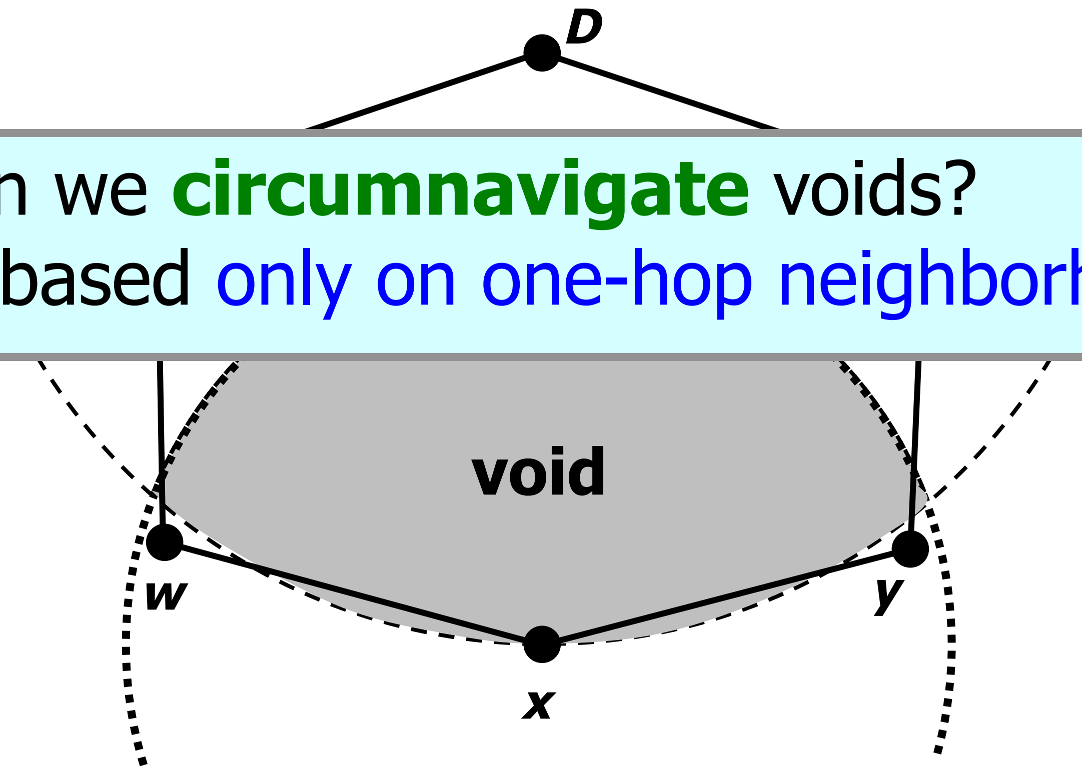
Greedy Forwarding Failure

Greedy forwarding not always possible! Consider:



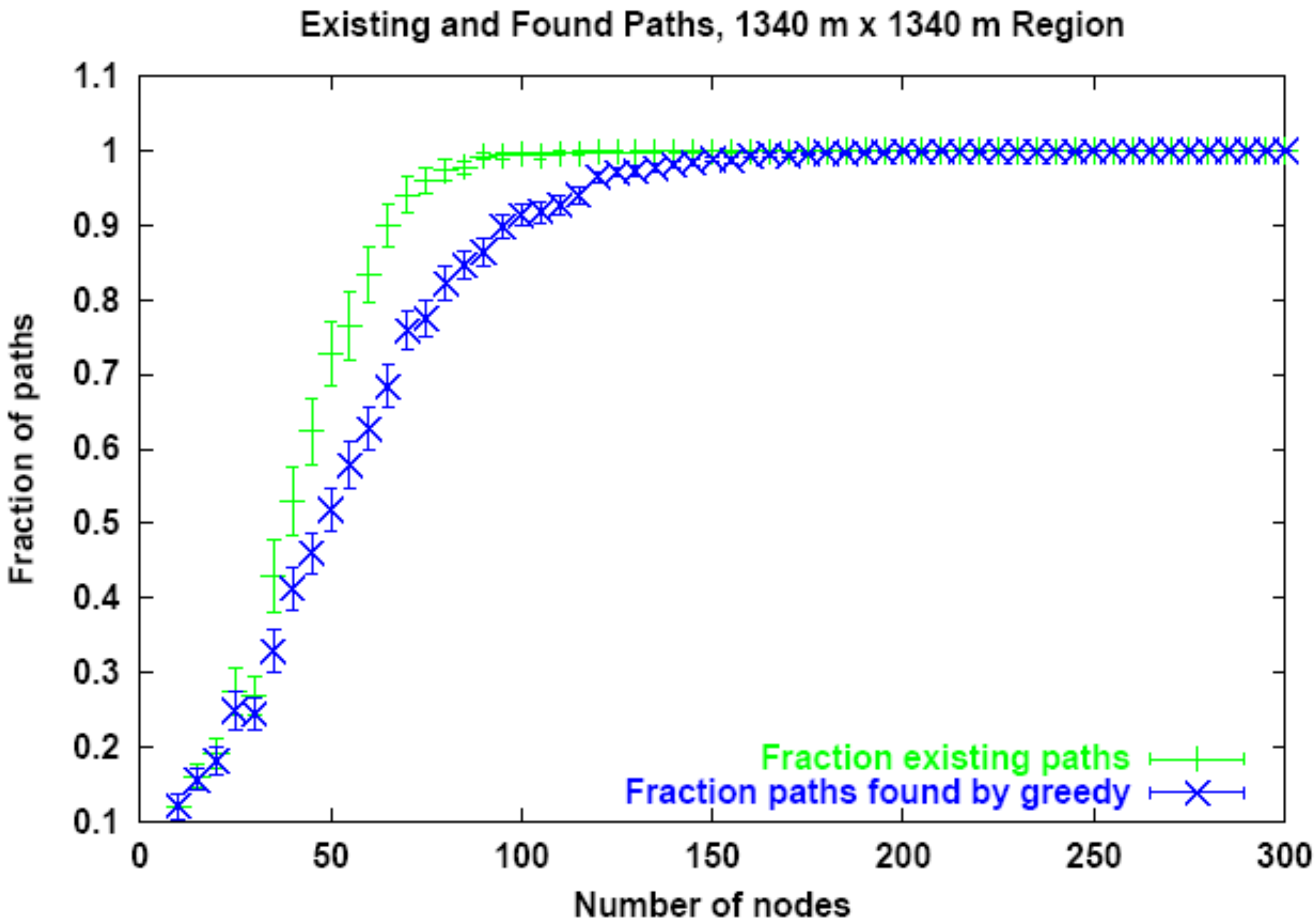
Greedy Forwarding Failure

Greedy forwarding not always possible! Consider:

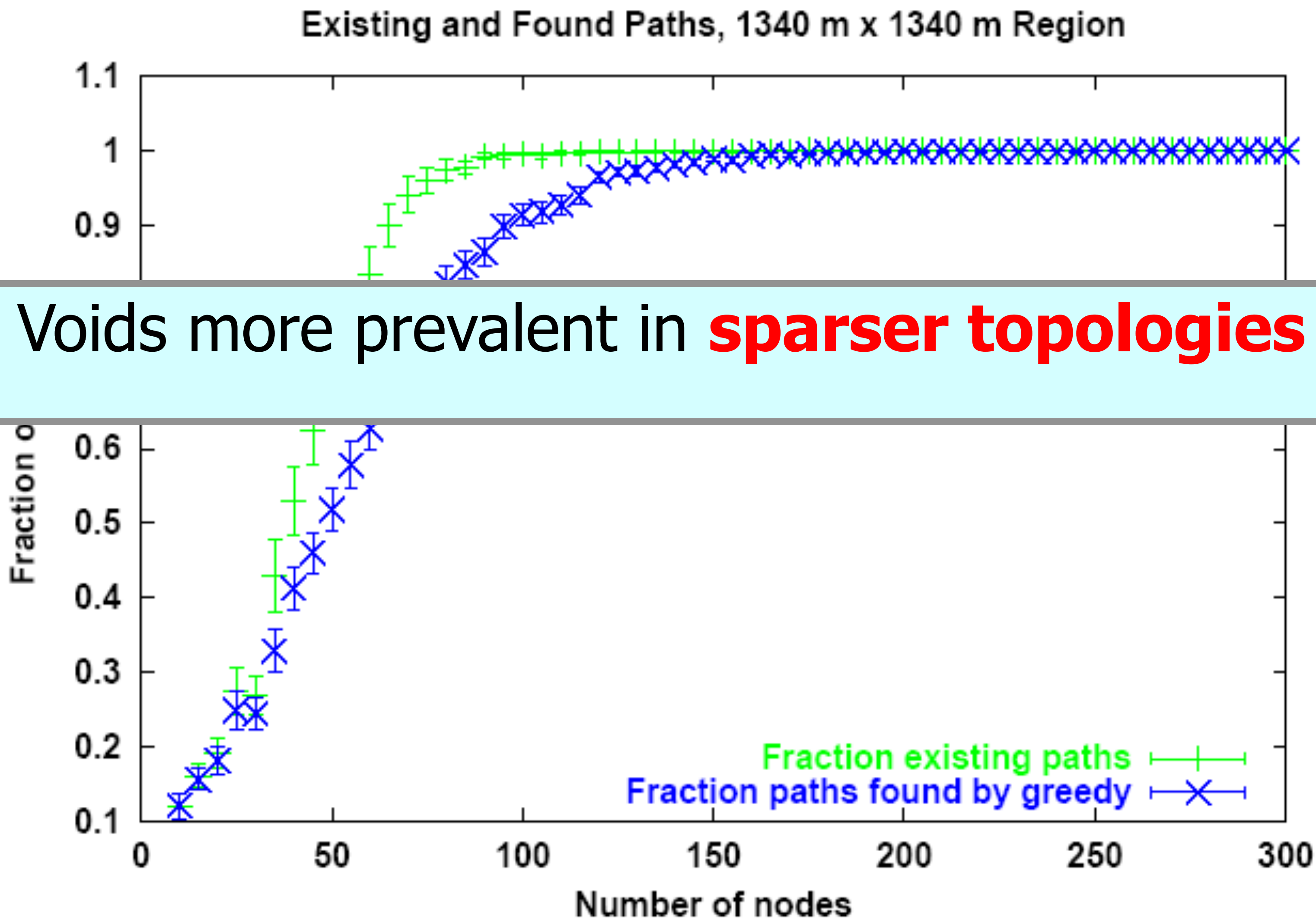


How can we **circumnavigate** voids?
...based only on one-hop neighborhood?

Node Density and Voids



Node Density and Voids

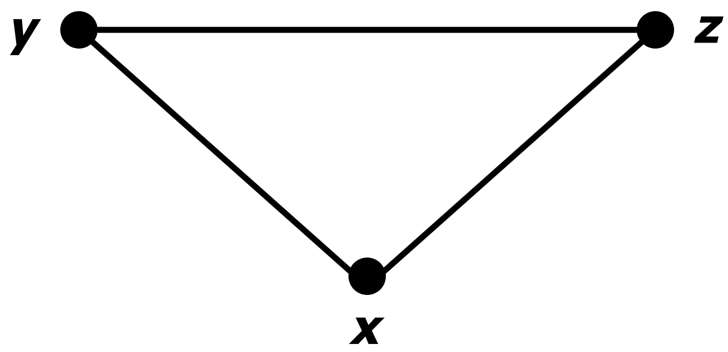


Voids more prevalent in **sparser topologies**

Void Traversal: The Right-hand Rule

Well-known graph traversal: right-hand rule

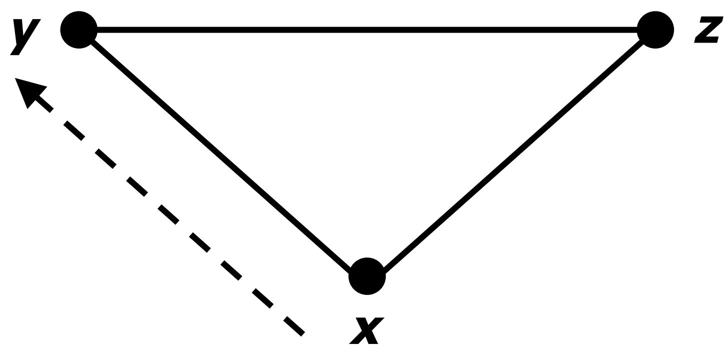
Requires only neighbors' positions



Void Traversal: The Right-hand Rule

Well-known graph traversal: right-hand rule

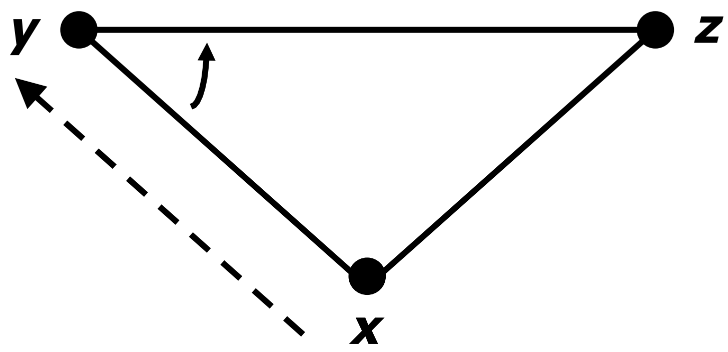
Requires only neighbors' positions



Void Traversal: The Right-hand Rule

Well-known graph traversal: right-hand rule

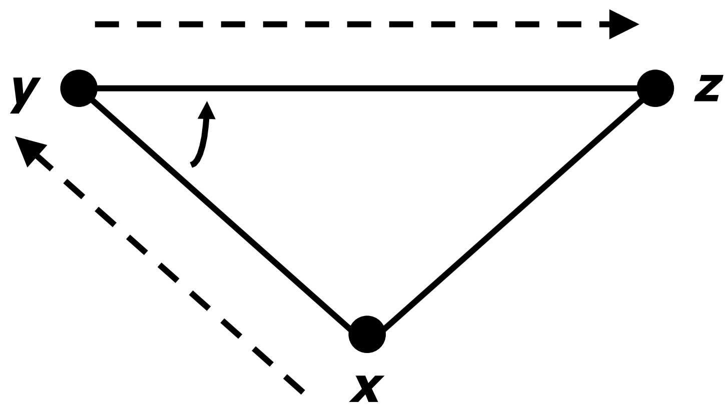
Requires only neighbors' positions



Void Traversal: The Right-hand Rule

Well-known graph traversal: right-hand rule

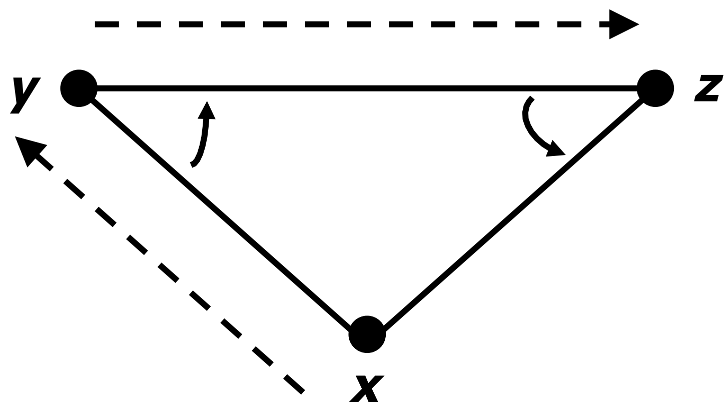
Requires only neighbors' positions



Void Traversal: The Right-hand Rule

Well-known graph traversal: right-hand rule

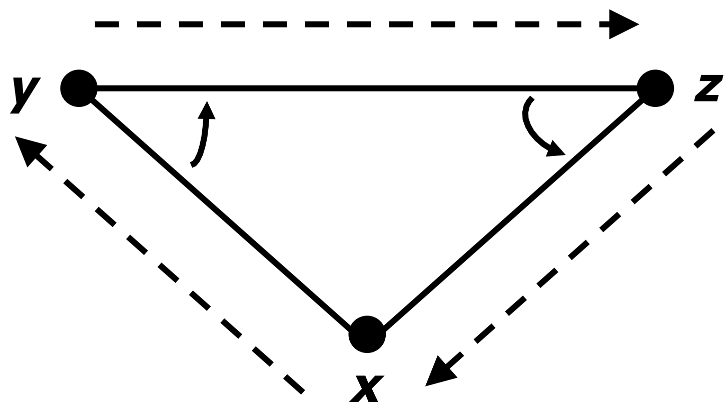
Requires only neighbors' positions



Void Traversal: The Right-hand Rule

Well-known graph traversal: right-hand rule

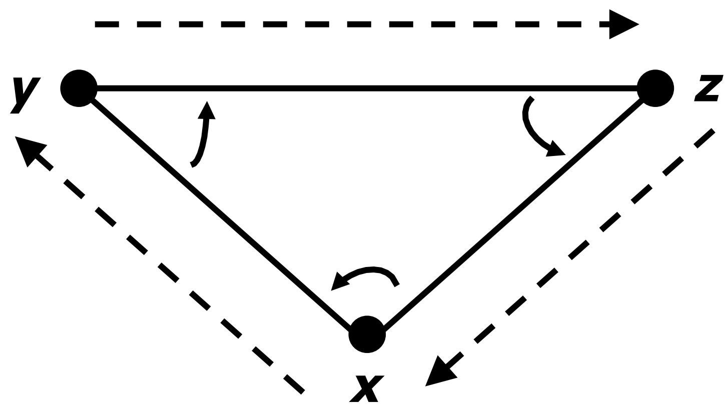
Requires only neighbors' positions



Void Traversal: The Right-hand Rule

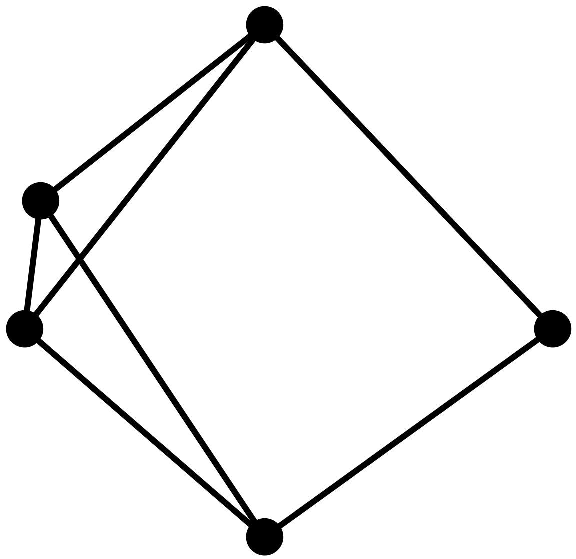
Well-known graph traversal: right-hand rule

Requires only neighbors' positions



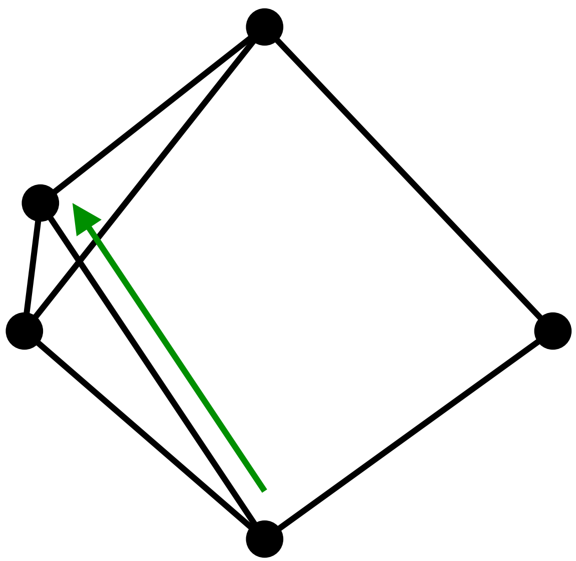
Planar vs. Non-planar Graphs

On graphs with edges that cross **(non-planar graphs),** right-hand rule may not tour enclosed face boundary



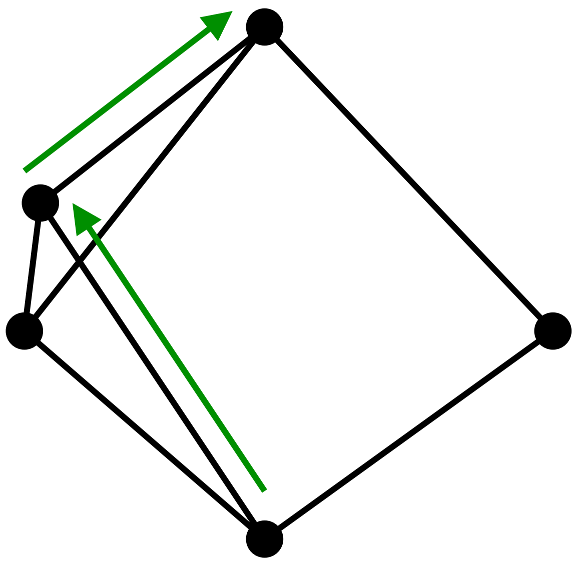
Planar vs. Non-planar Graphs

On graphs with edges that cross **(non-planar graphs),** right-hand rule may not tour enclosed face boundary



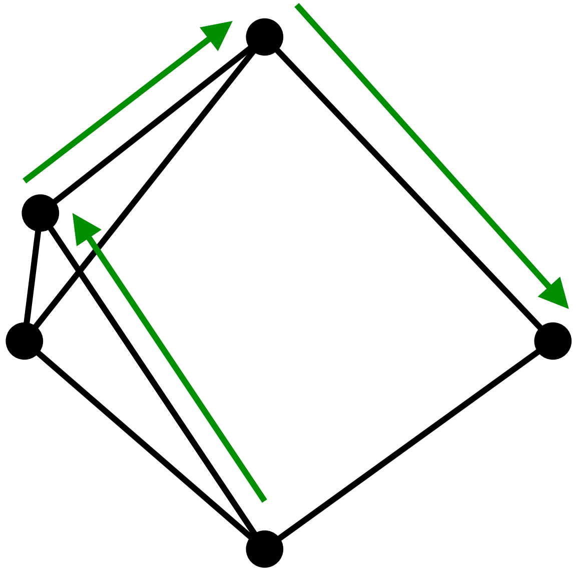
Planar vs. Non-planar Graphs

On graphs with edges that cross **(non-planar graphs),** right-hand rule may not tour enclosed face boundary



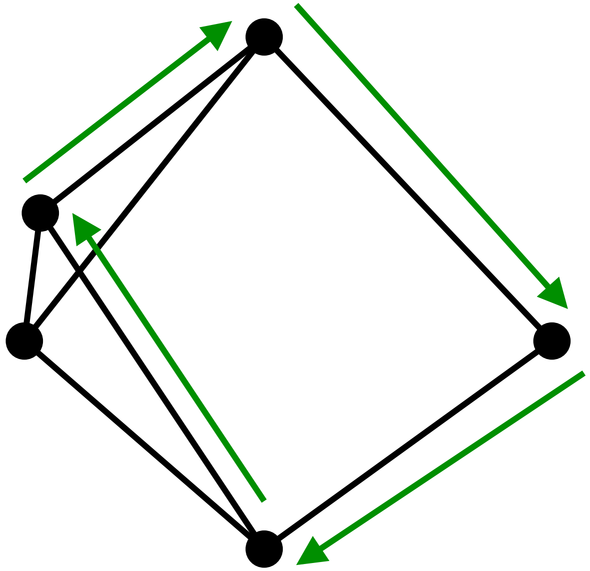
Planar vs. Non-planar Graphs

On graphs with edges that cross **(non-planar graphs),** right-hand rule may not tour enclosed face boundary



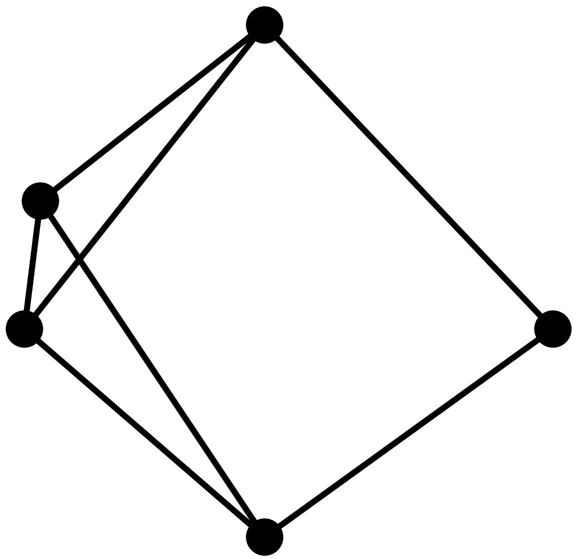
Planar vs. Non-planar Graphs

On graphs with edges that cross **(non-planar graphs),** right-hand rule may not tour enclosed face boundary



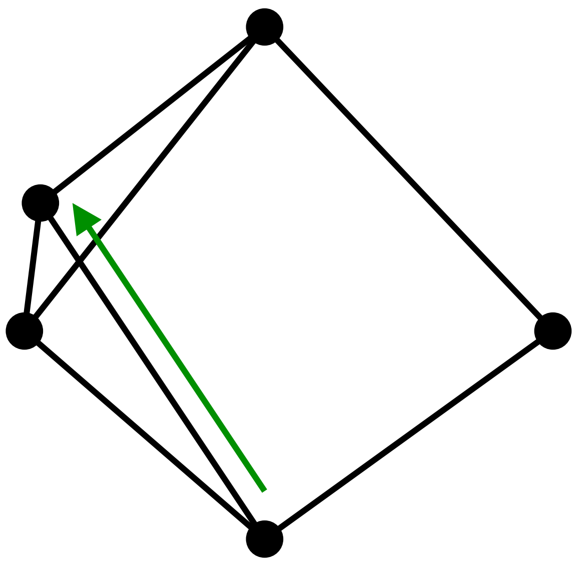
Planar vs. Non-planar Graphs

On graphs with edges that cross **(non-planar graphs),** right-hand rule may not tour enclosed face boundary



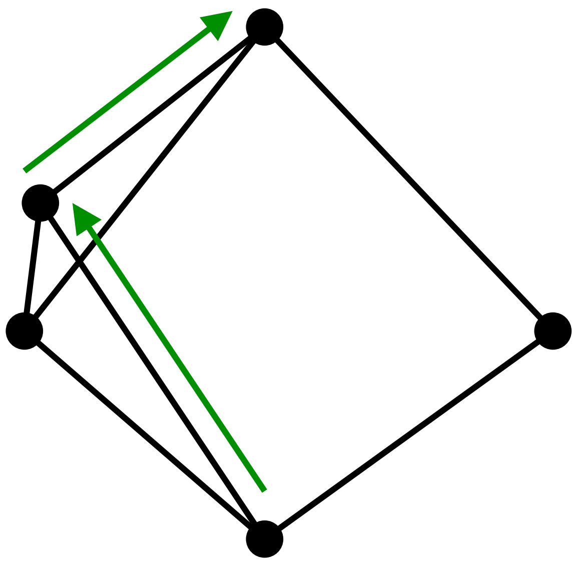
Planar vs. Non-planar Graphs

On graphs with edges that cross **(non-planar graphs),** right-hand rule may not tour enclosed face boundary



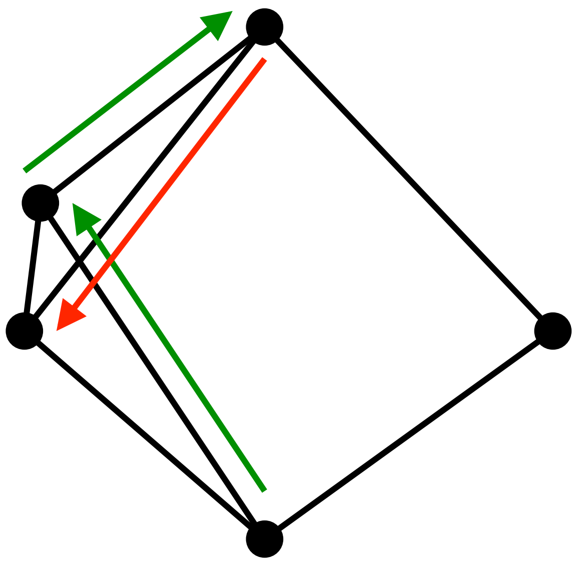
Planar vs. Non-planar Graphs

On graphs with edges that cross **(non-planar graphs),** right-hand rule may not tour enclosed face boundary



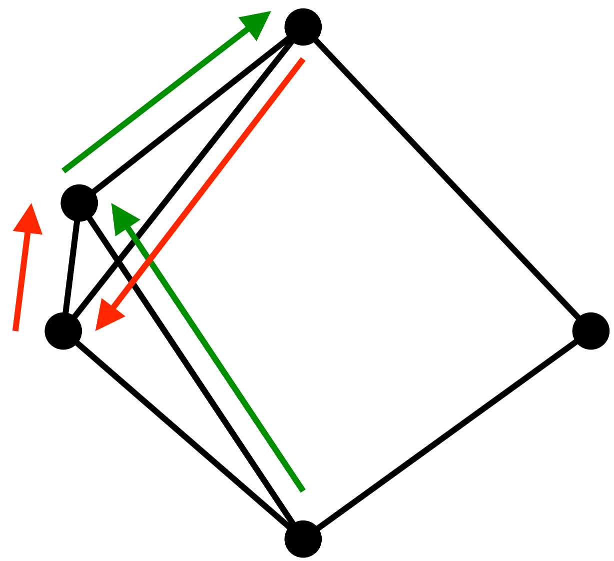
Planar vs. Non-planar Graphs

On graphs with edges that cross **(non-planar graphs),** right-hand rule may not tour enclosed face boundary



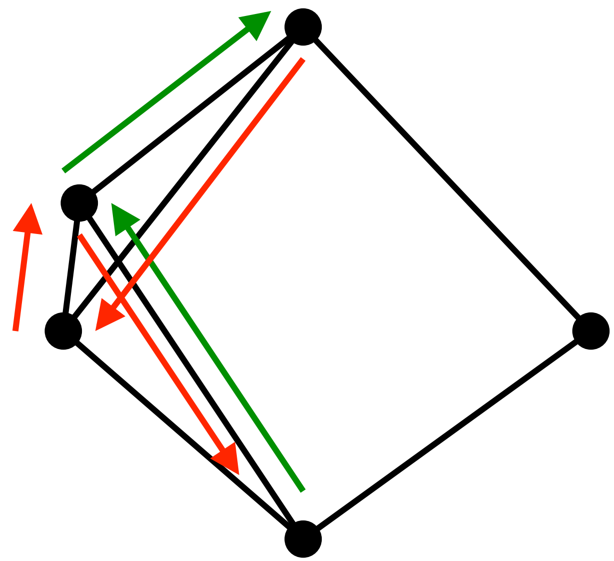
Planar vs. Non-planar Graphs

On graphs with edges that cross **(non-planar graphs),** right-hand rule may not tour enclosed face boundary



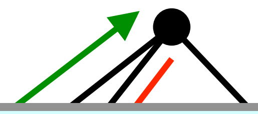
Planar vs. Non-planar Graphs

On graphs with edges that cross **(non-planar graphs),** right-hand rule may not tour enclosed face boundary



Planar vs. Non-planar Graphs

On graphs with edges that cross **(non-planar graphs),** right-hand rule may not tour enclosed face boundary



How to remove crossing edges without partitioning graph?

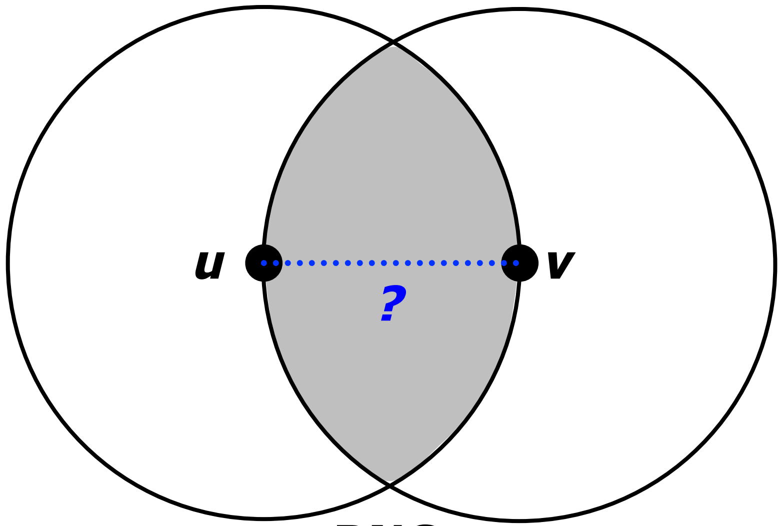
And using only single-hop neighbors' positions?

Planarized Graphs

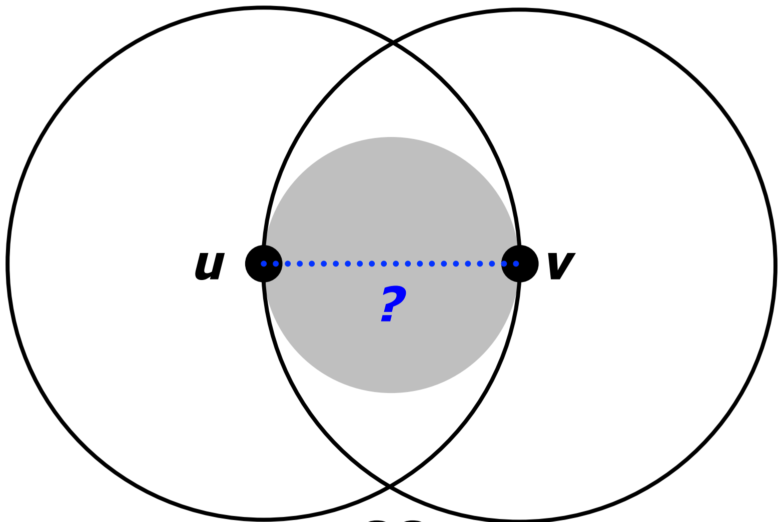
Relative Neighborhood Graph (RNG) [Toussaint, '80] and Gabriel Graph (GG) [Gabriel, '69]: long-known planar graphs

Assume edge exists between any pair of nodes separated by less than threshold distance (*i.e.,* nominal radio range)

RNG and GG can be constructed from only neighbors' positions, and can be shown not to partition network!



RNG



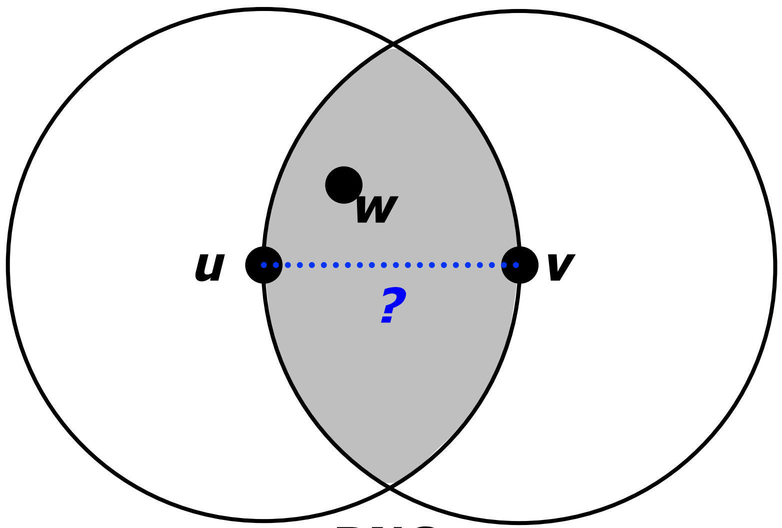
GG

Planarized Graphs

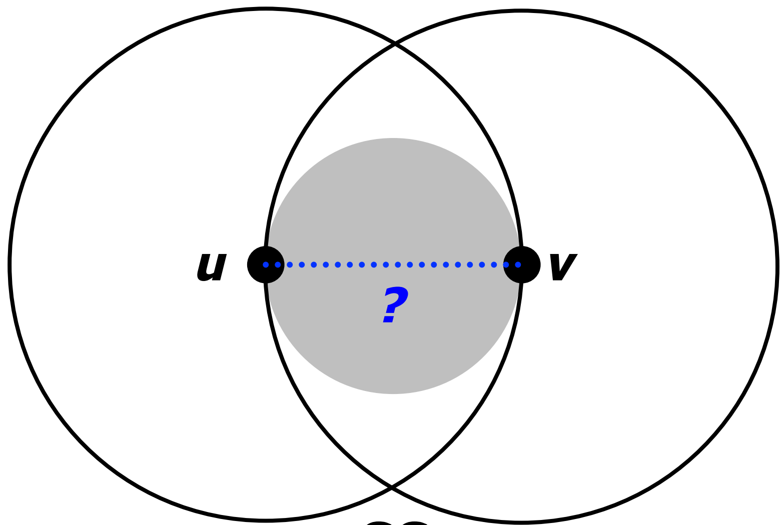
Relative Neighborhood Graph (RNG) [Toussaint, '80] and Gabriel Graph (GG) [Gabriel, '69]: long-known planar graphs

Assume edge exists between any pair of nodes separated by less than threshold distance (*i.e.,* nominal radio range)

RNG and GG can be constructed from only neighbors' positions, and can be shown not to partition network!



RNG



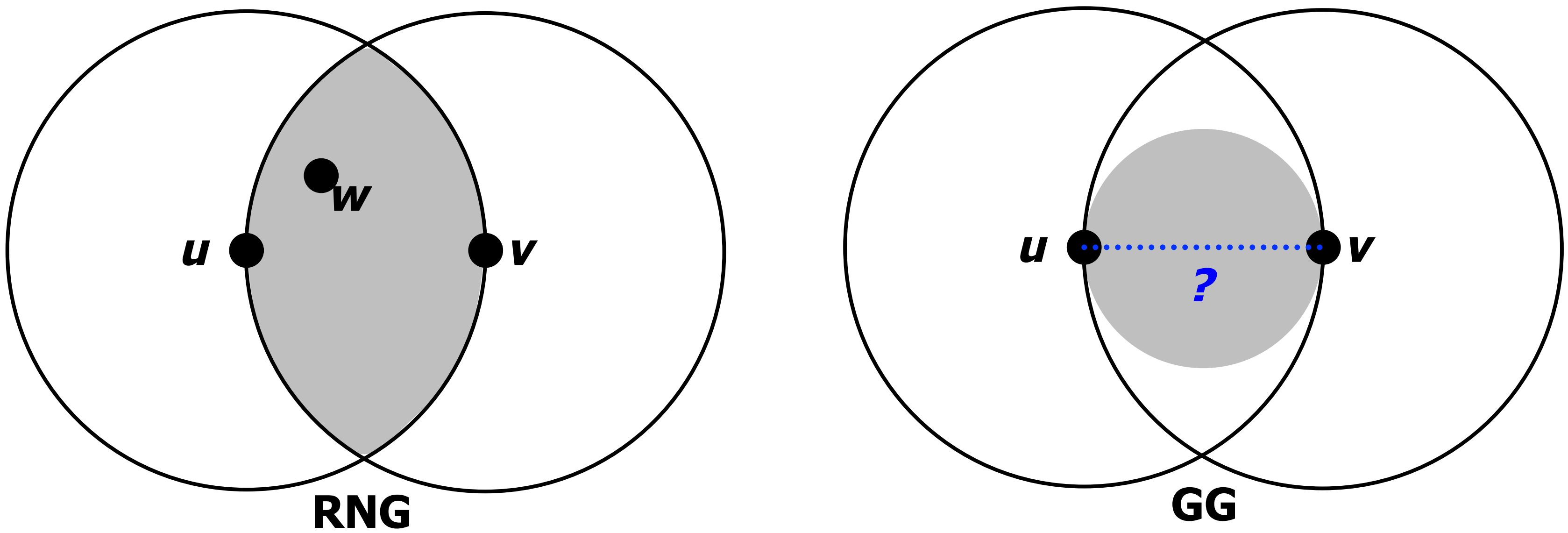
GG

Planarized Graphs

Relative Neighborhood Graph (RNG) [Toussaint, '80] and Gabriel Graph (GG) [Gabriel, '69]: long-known planar graphs

Assume edge exists between any pair of nodes separated by less than threshold distance (*i.e.,* nominal radio range)

RNG and GG can be constructed from only neighbors' positions, and can be shown not to partition network!

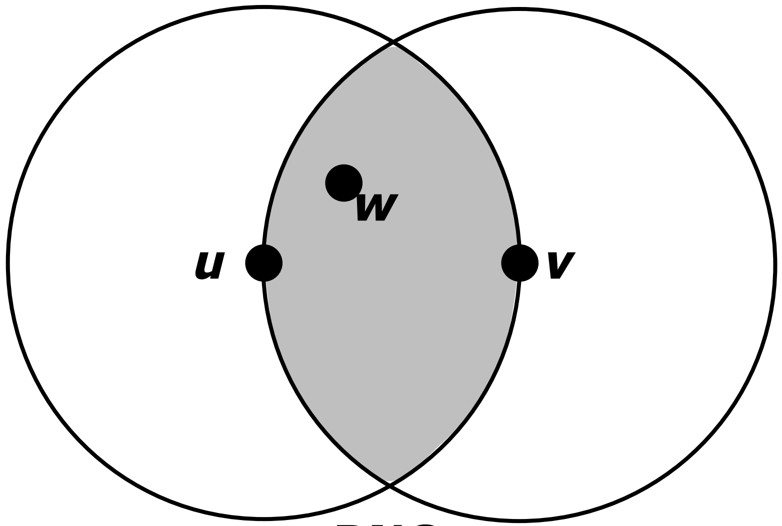


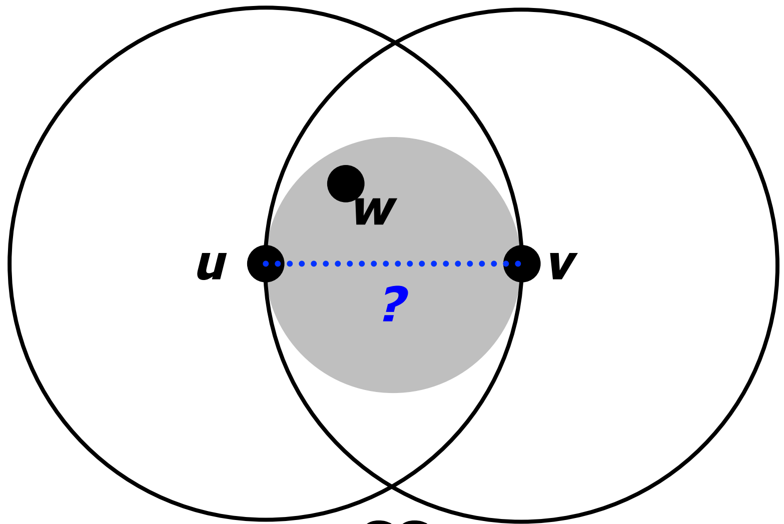
Planarized Graphs

Relative Neighborhood Graph (RNG) [Toussaint, '80] and Gabriel Graph (GG) [Gabriel, '69]: long-known planar graphs

Assume edge exists between any pair of nodes separated by less than threshold distance (*i.e.,* nominal radio range)

RNG and GG can be constructed from only neighbors' positions, and can be shown not to partition network!



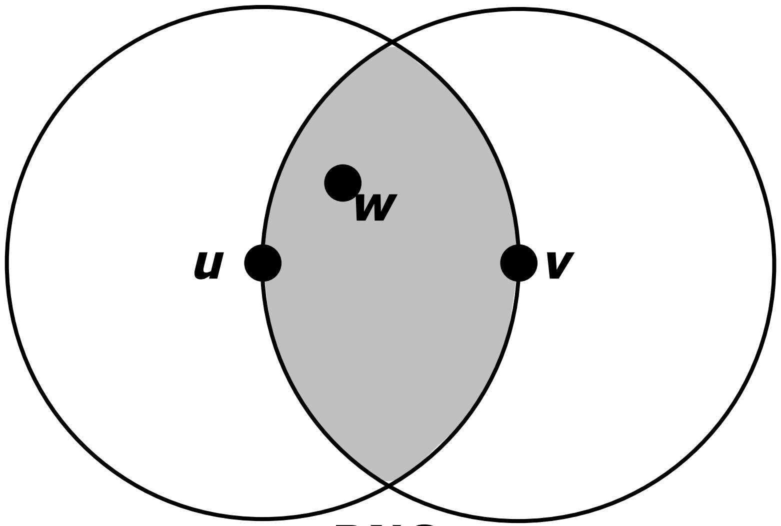


Planarized Graphs

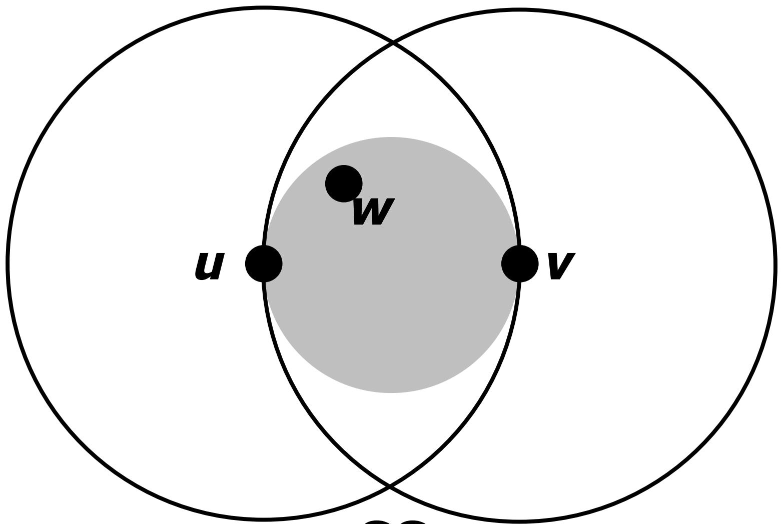
Relative Neighborhood Graph (RNG) [Toussaint, ’80] and Gabriel Graph (GG) [Gabriel, ’69]: long-known planar graphs

Assume edge exists between any pair of nodes separated by less than threshold distance (*i.e.,* nominal radio range)

RNG and GG can be constructed from only neighbors’ positions, and can be shown not to partition network!



RNG



GG

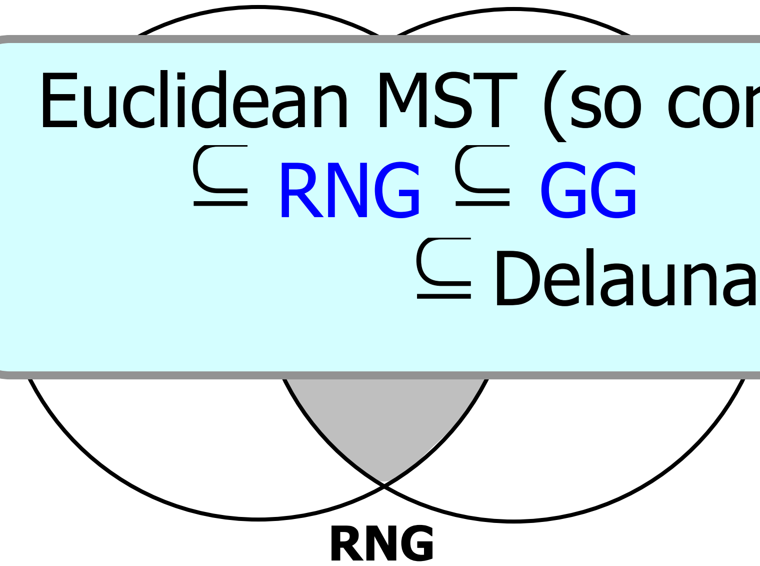
Planarized Graphs

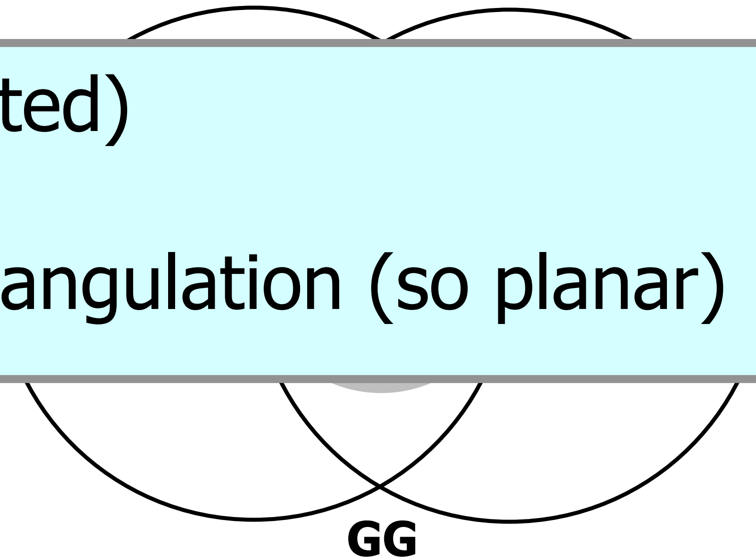
Relative Neighborhood Graph (RNG) [Toussaint, '80] and Gabriel Graph (GG) [Gabriel, '69]: long-known planar graphs

Assume edge exists between any pair of nodes separated by less than threshold distance (*i.e.*, nominal radio range)

RNG and GG can be constructed from only neighbors' positions, and can be shown not to partition network!

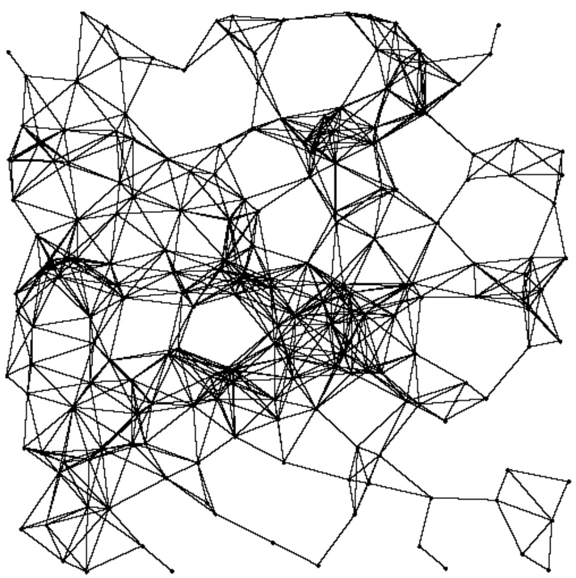
Euclidean MST (so connected)
$\subseteq$ RNG $\subseteq$ GG
$\subseteq$ Delaunay Triangulation (so planar)



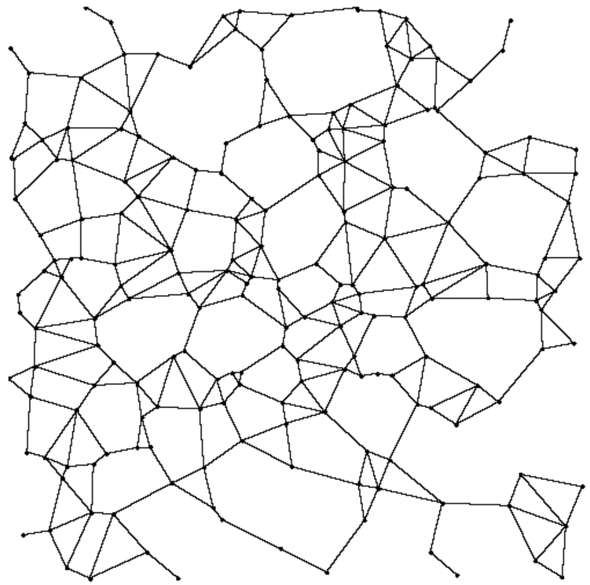


Planarized Graphs: Example

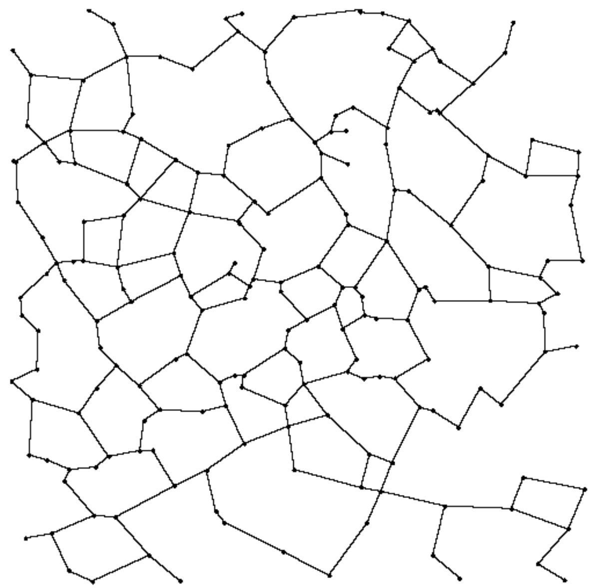
200 nodes, placed uniformly at random on 2000-by-2000-meter region; 250-meter radio range



Full Graph



GG Subgraph

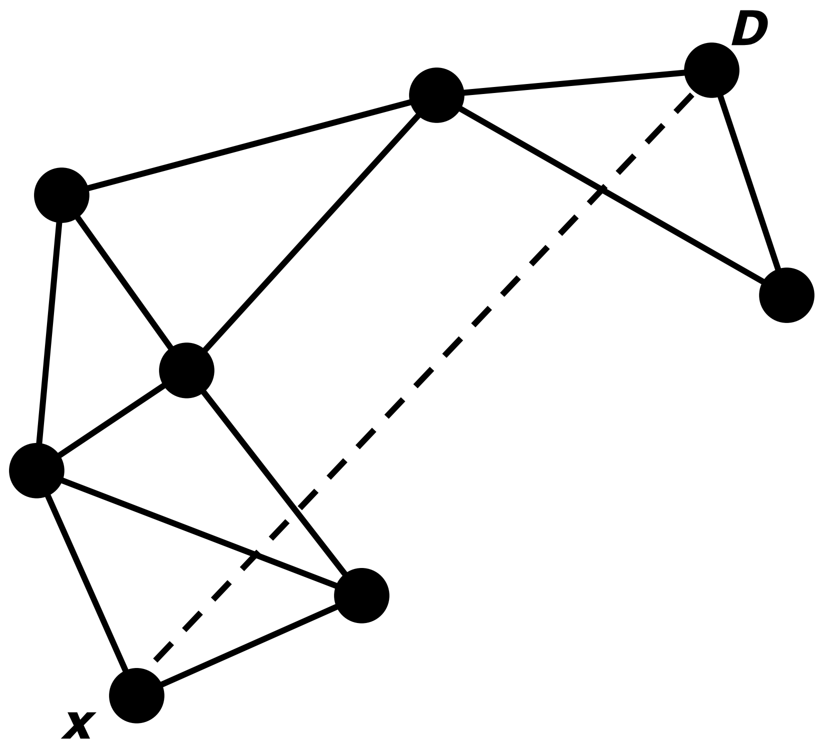


RNG Subgraph

Full Greedy Perimeter Stateless Routing

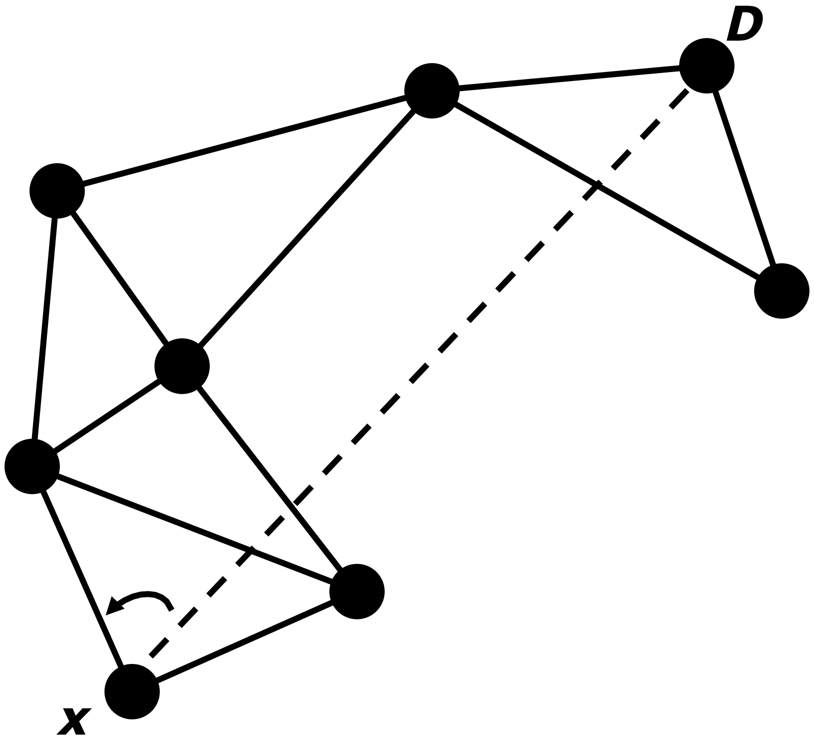
- All packets begin in greedy mode
- Greedy mode uses full graph
- Upon greedy failure, node marks its location in packet, marks packet in perimeter mode
- Perimeter mode packets follow simple planar graph traversal:
  - Forward along successively closer faces by right-hand rule, until reaching destination
  - Packets return to greedy mode upon reaching node closer to destination than perimeter mode entry point

Perimeter Mode Forwarding Example



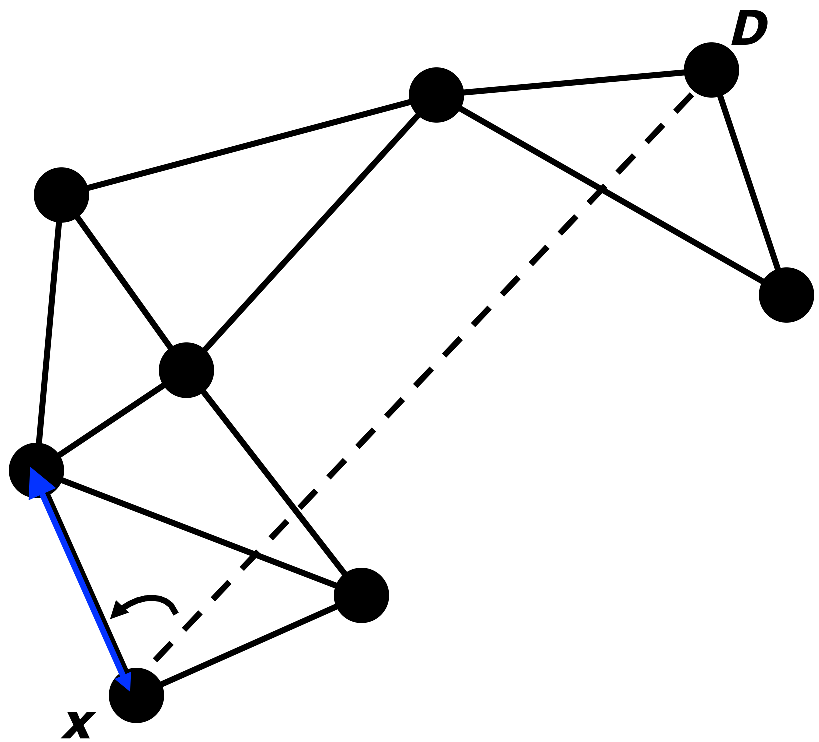
- Traverse face closer to *D* along $\overline{xD}$ by right-hand rule, until crossing $\overline{xD}$
- Repeat with next-closer face, *&c.*

Perimeter Mode Forwarding Example



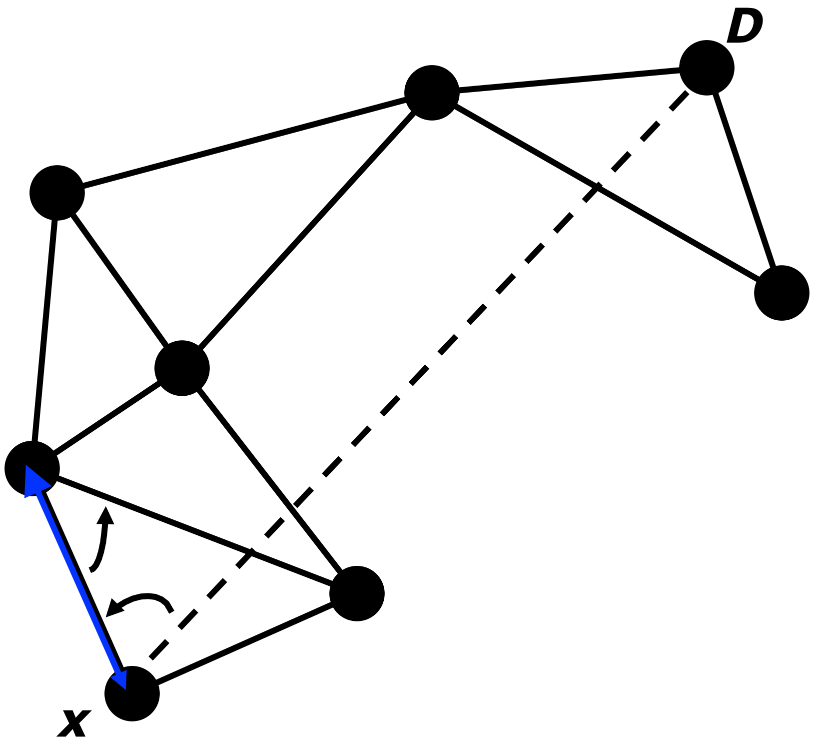
- Traverse face closer to *D* along $\overline{xD}$ by right-hand rule, until crossing $\overline{xD}$
- Repeat with next-closer face, *&c.*

Perimeter Mode Forwarding Example



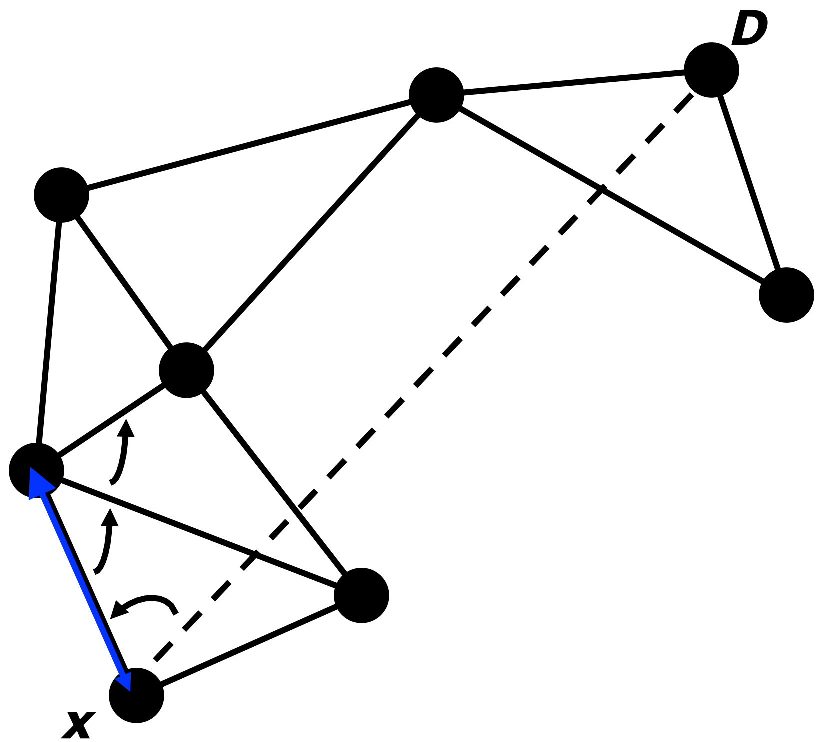
- Traverse face closer to *D* along $\overline{xD}$ by right-hand rule, until crossing $\overline{xD}$
- Repeat with next-closer face, *&c.*

Perimeter Mode Forwarding Example



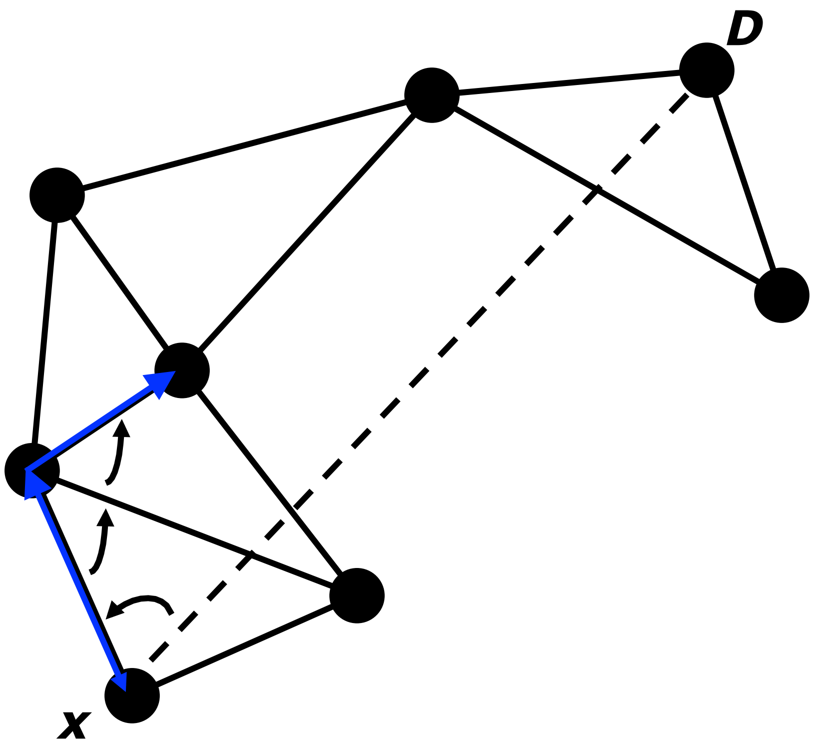
- Traverse face closer to *D* along $\overline{xD}$ by right-hand rule, until crossing $\overline{xD}$
- Repeat with next-closer face, *&c.*

Perimeter Mode Forwarding Example



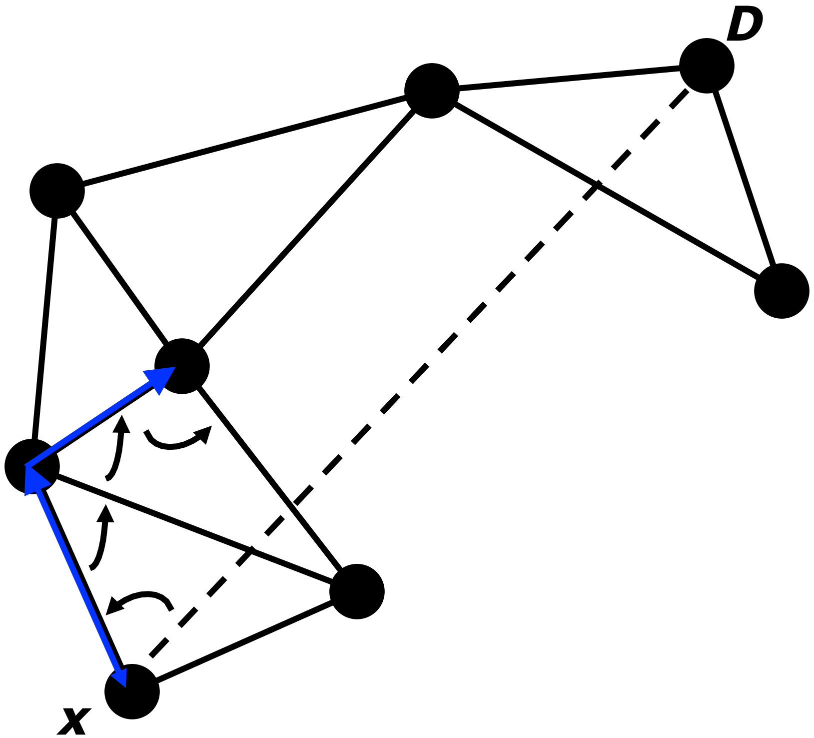
- Traverse face closer to *D* along $\overline{xD}$ by right-hand rule, until crossing $\overline{xD}$
- Repeat with next-closer face, *&c.*

Perimeter Mode Forwarding Example



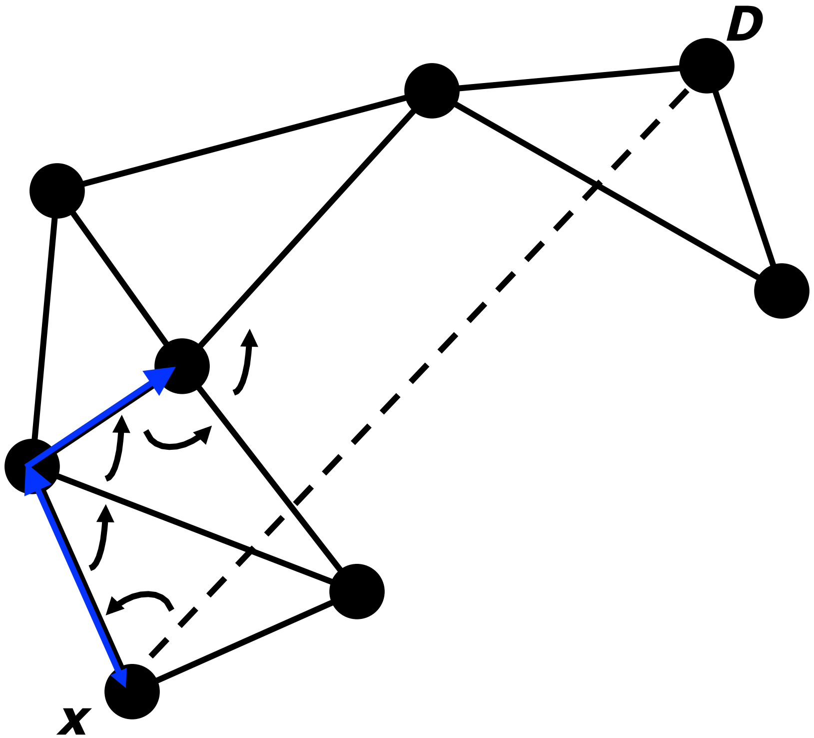
- Traverse face closer to *D* along $\overline{xD}$ by right-hand rule, until crossing $\overline{xD}$
- Repeat with next-closer face, *&c.*

Perimeter Mode Forwarding Example



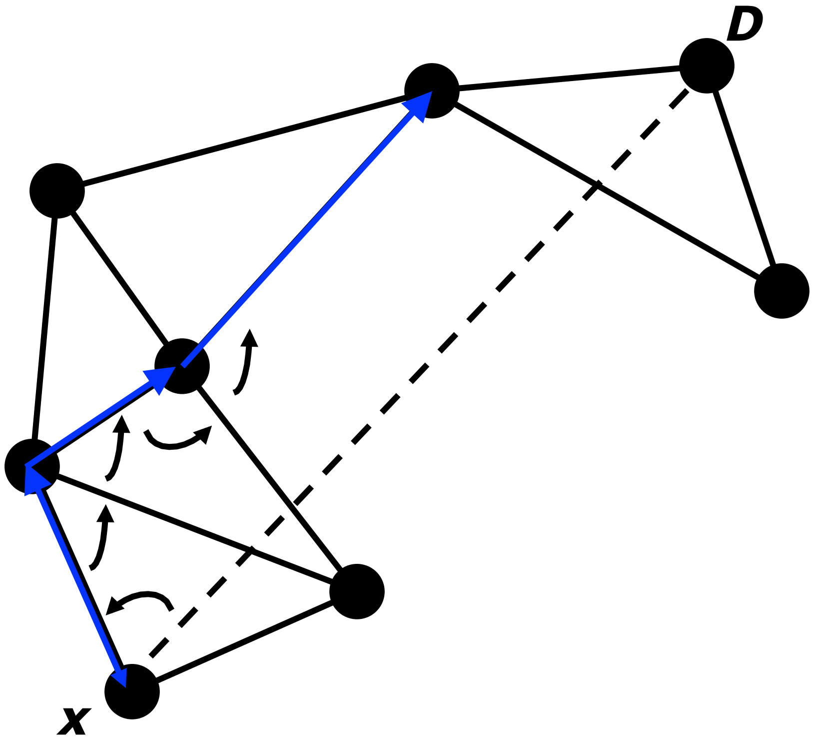
- Traverse face closer to *D* along $\overline{xD}$ by right-hand rule, until crossing $\overline{xD}$
- Repeat with next-closer face, *&c.*

Perimeter Mode Forwarding Example



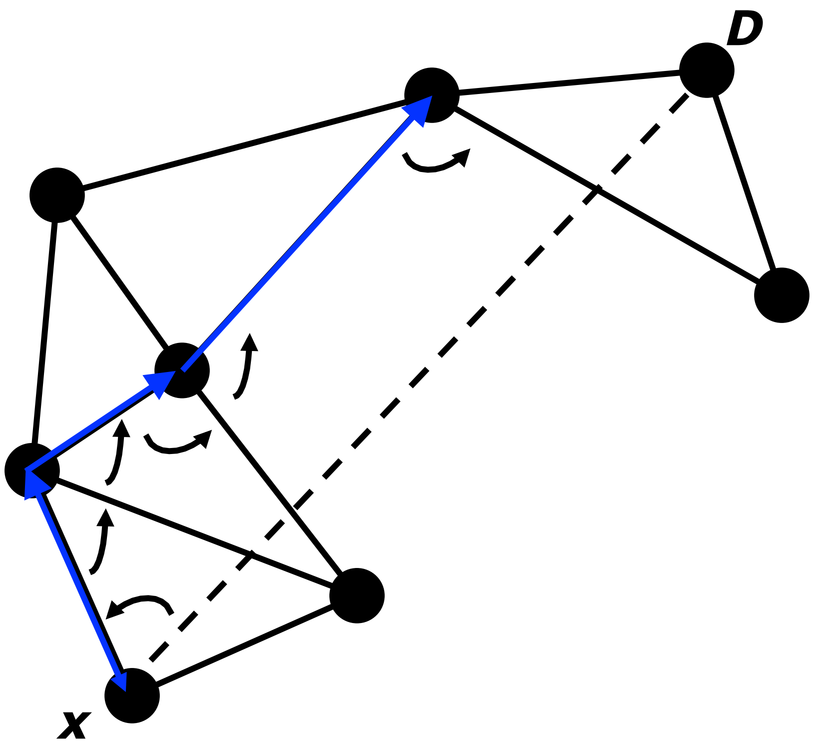
- Traverse face closer to *D* along $\overline{xD}$ by right-hand rule, until crossing $\overline{xD}$
- Repeat with next-closer face, *&c.*

Perimeter Mode Forwarding Example



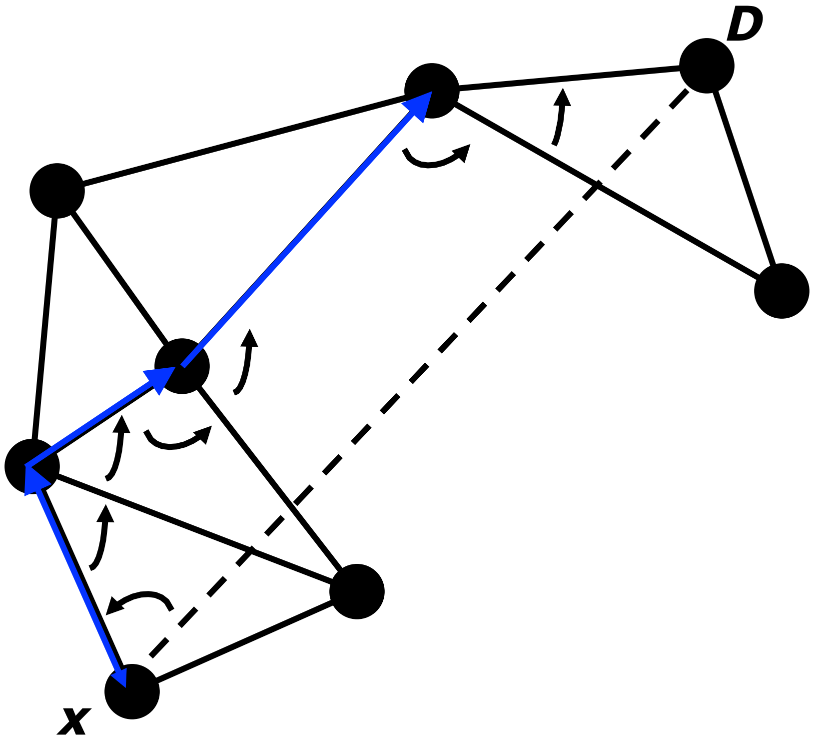
- Traverse face closer to *D* along $\overline{xD}$ by right-hand rule, until crossing $\overline{xD}$
- Repeat with next-closer face, *&c.*

Perimeter Mode Forwarding Example



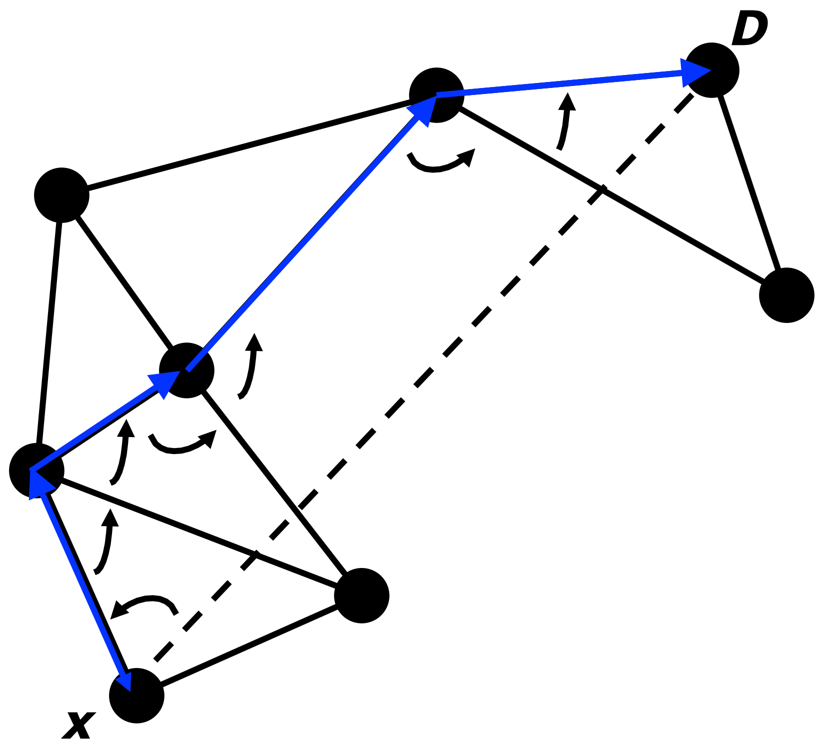
- Traverse face closer to *D* along $\overline{xD}$ by right-hand rule, until crossing $\overline{xD}$
- Repeat with next-closer face, *&c.*

Perimeter Mode Forwarding Example



- Traverse face closer to *D* along $\overline{xD}$ by right-hand rule, until crossing $\overline{xD}$
- Repeat with next-closer face, *&c.*

Perimeter Mode Forwarding Example



- Traverse face closer to *D* along $\overline{xD}$ by right-hand rule, until crossing $\overline{xD}$
- Repeat with next-closer face, *&c.*

Protocol Tricks for Dynamic Networks

- Use of MAC-layer failure feedback: As in DSR [Broch, Johnson, '98], interpret retransmit failure reports from 802.11 MAC as indication neighbor gone out-of-range
- Interface queue traversal and packet purging: Upon MAC retransmit failure for a neighbor, remove packets to that neighbor from IFQ to avoid head-of-line blocking of 802.11 transmitter during retries
- Promiscuous network interface: Reduce beacon load and keep positions stored in neighbor tables current by tagging all packets with forwarding node's position
- Planarization triggers: Re-planarize upon acquisition of new neighbor and every loss of former neighbor, to keep planarization up-to-date as topology changes

Outline

- Motivation
- Context
- Algorithm
  - Greedy forwarding
  - Graph planarization
  - Perimeter forwarding
- Evaluation in simulation
- Footnotes
  - Open questions
  - Foibles of simulation

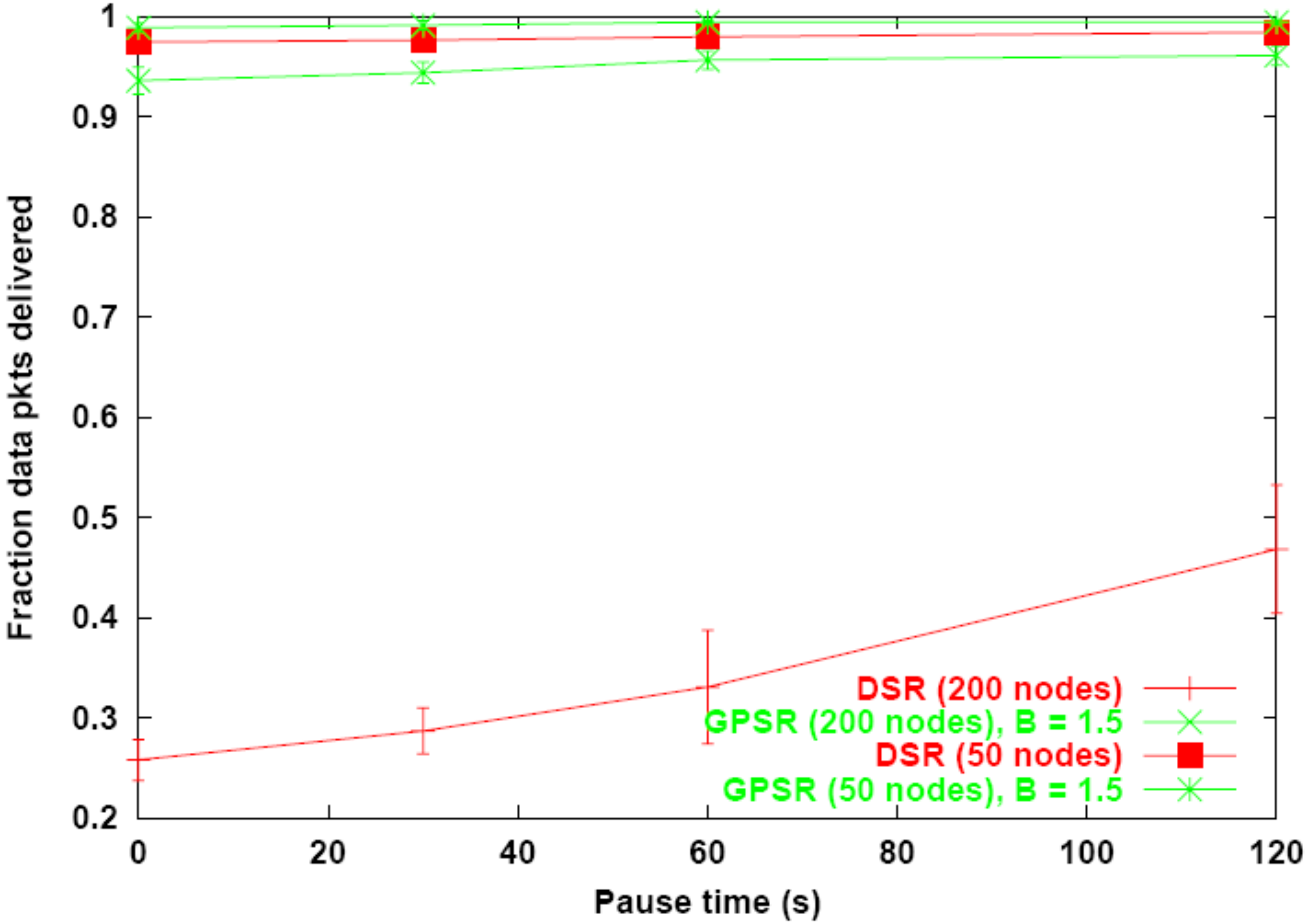
Evaluation: Simulations

- ns-2 with wireless extensions [Broch *et al.*, '98]; full 802.11 MAC, free space physical propagation
- Topologies:

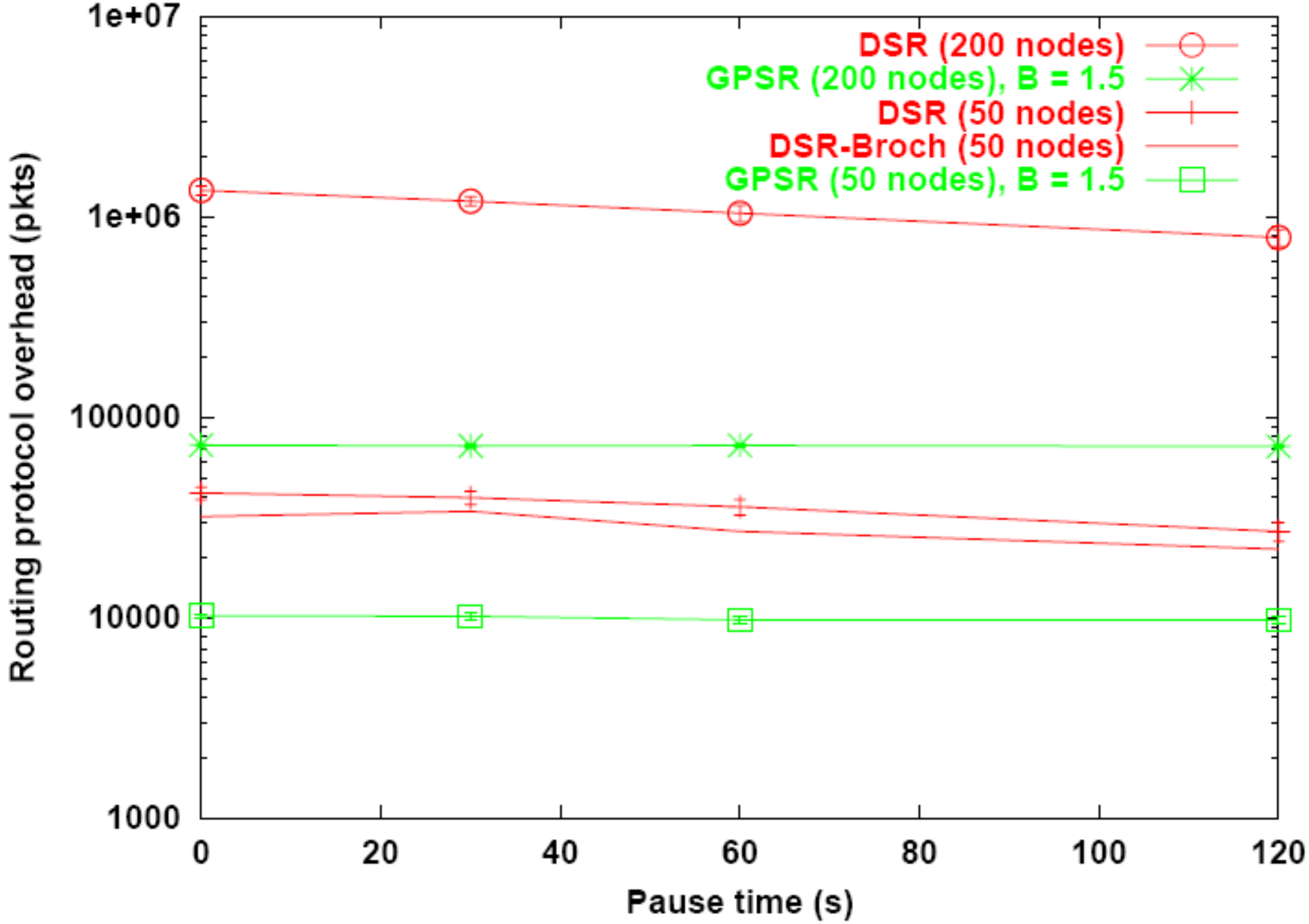
Nodes	Region	Density
50	1500 m x 300 m	1 node / 9000 m^2
200	3000 m x 600 m	1 node / 9000 m^2
50	1340 m x 1340 m	1 node / 35912 m^2

- 30 2-Kbps CBR flows; 64-byte data packets
- Random Waypoint Mobility in [1, 20 m/s]; Pause Time [0, 30, 60, 120s]; 1.5s GPSR beacons

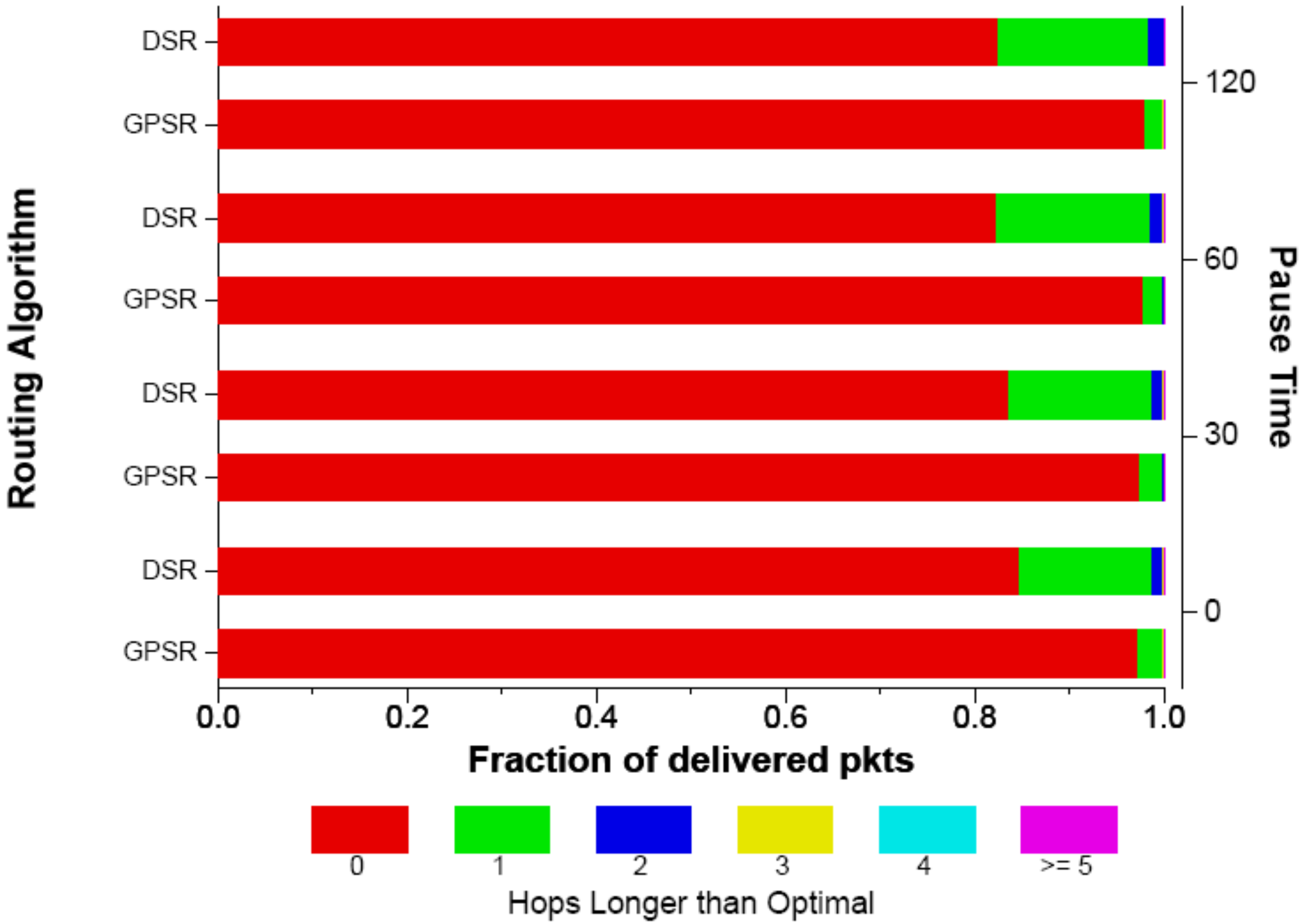
Packet Delivery Success Rate (50, 200; Dense)



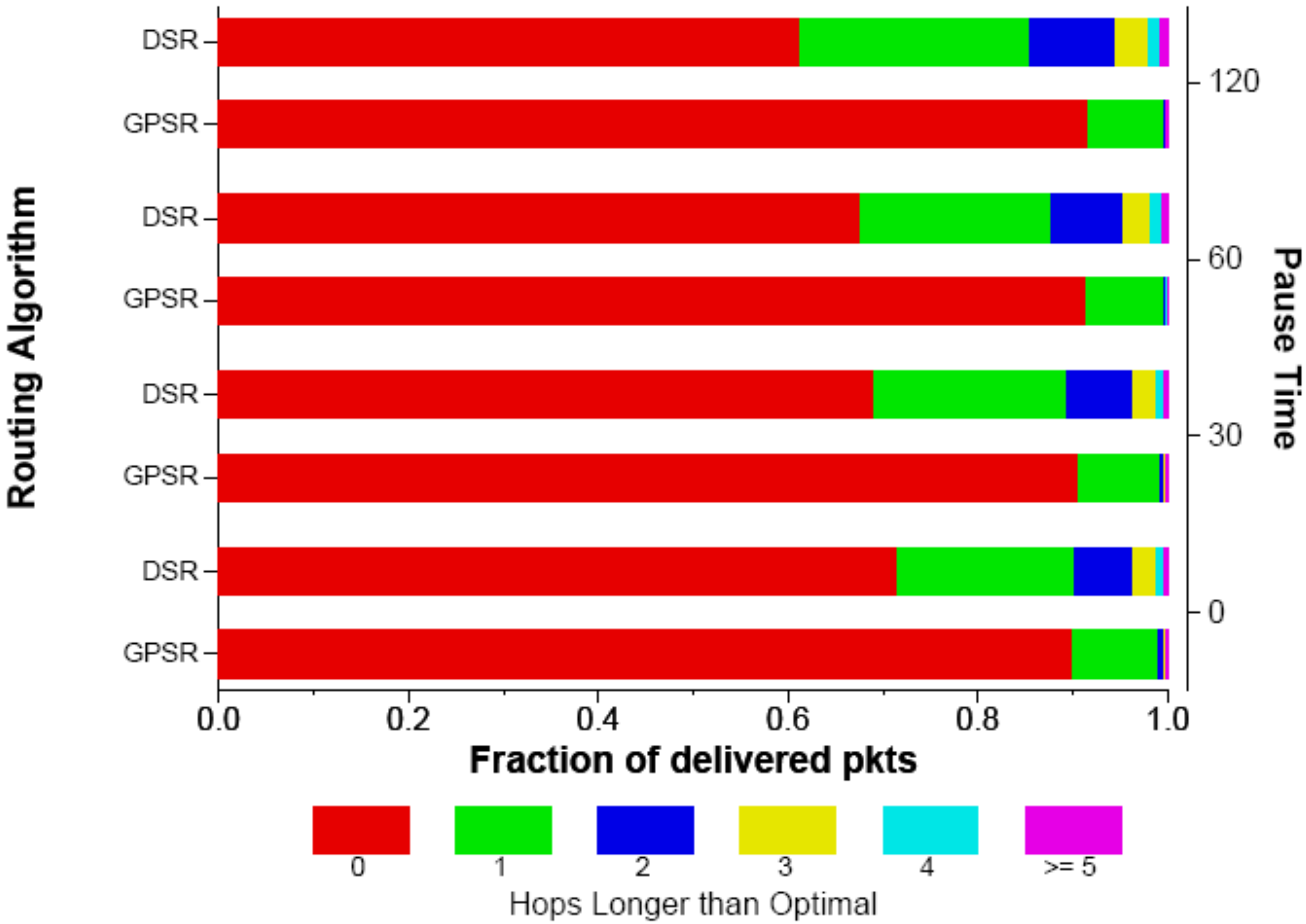
Routing Protocol Overhead (50, 200; Dense)



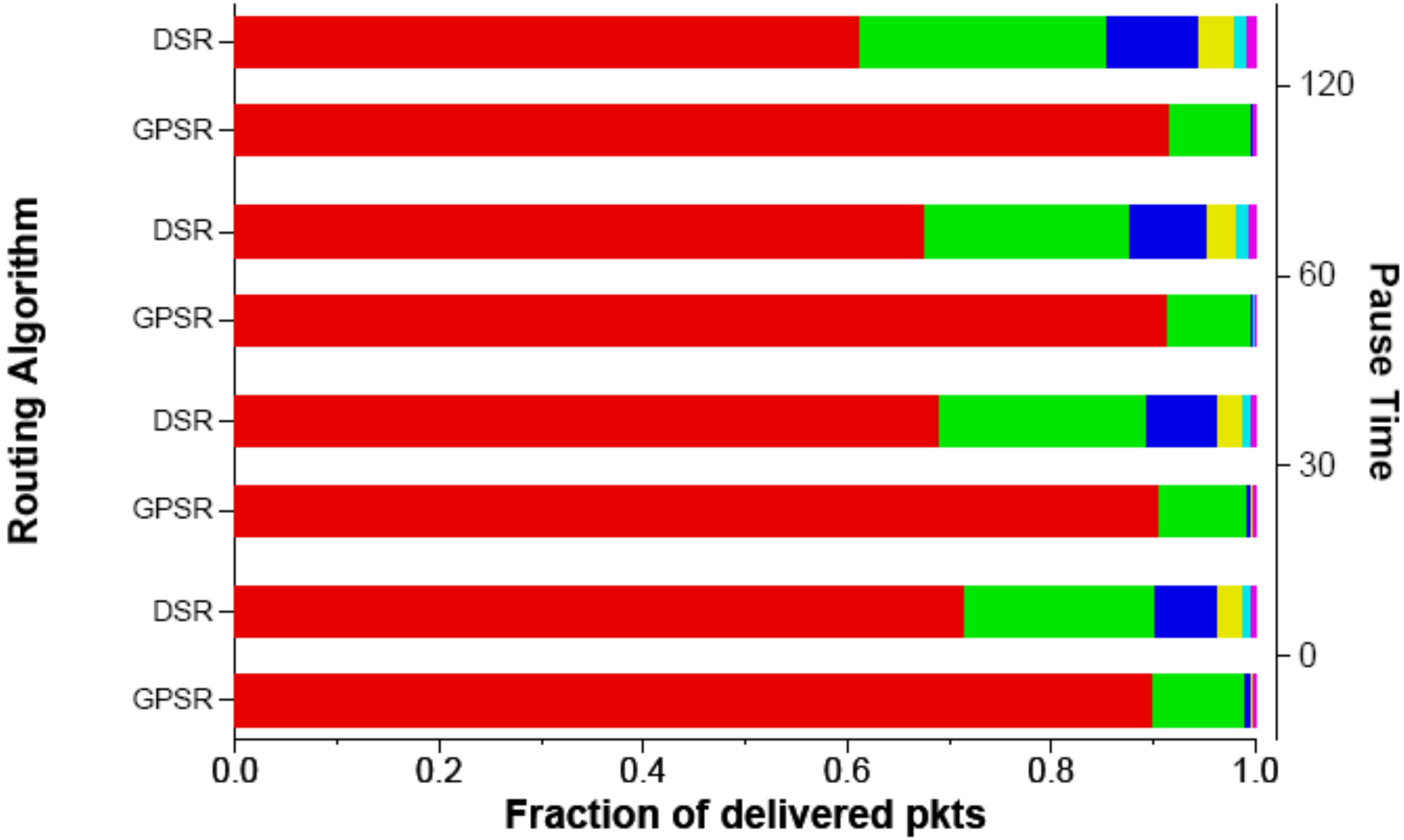
Path Length (50; Dense)



Path Length (200; Dense)

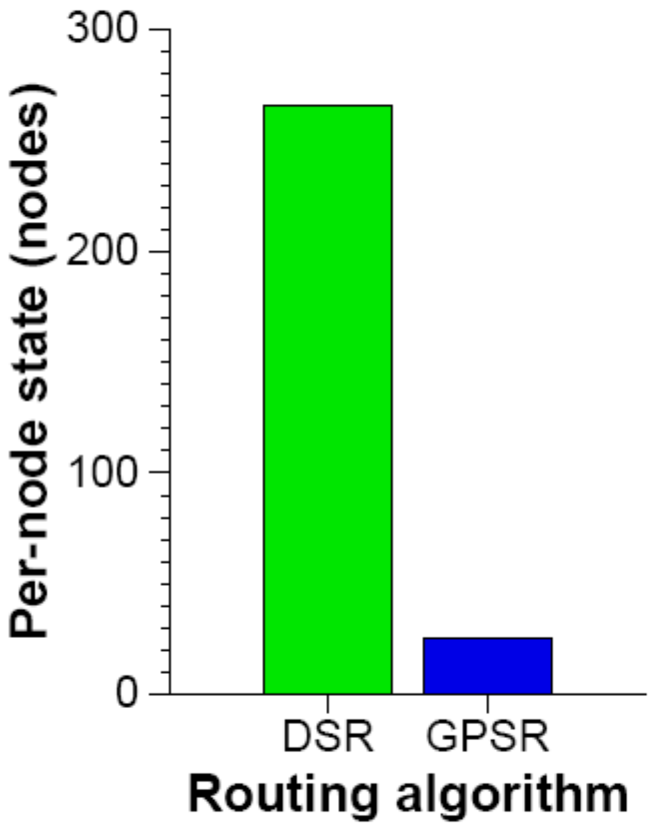


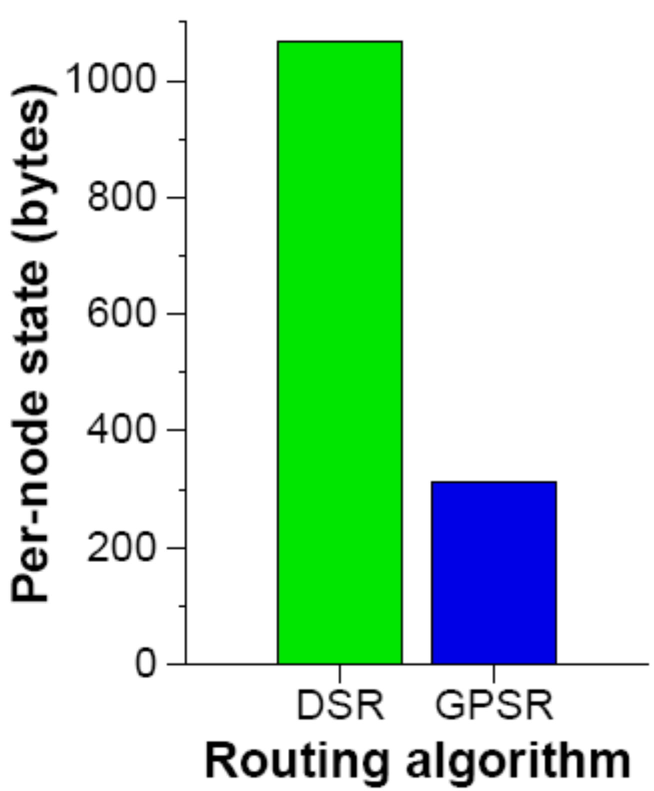
Path Length (200; Dense)



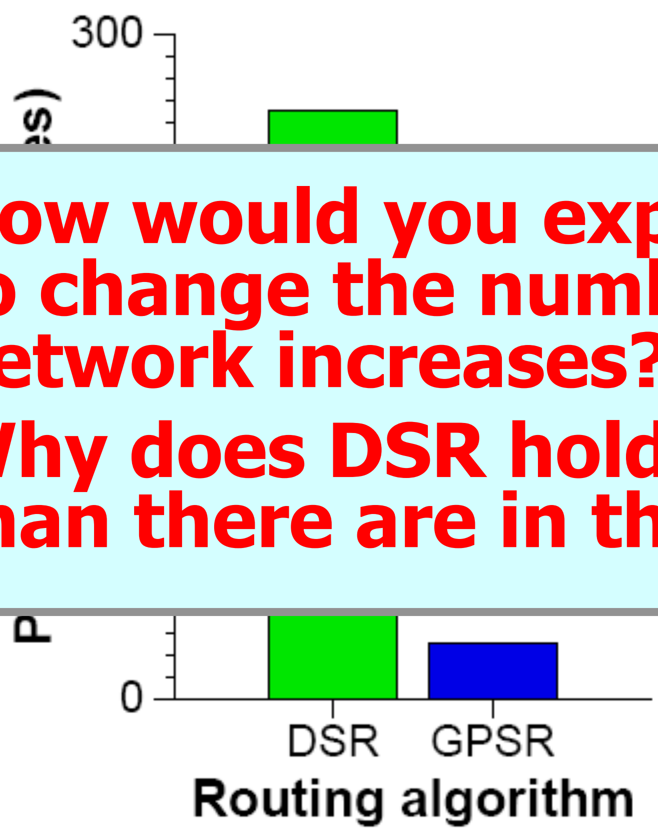
Why does DSR find shorter paths more of the time when mobility rate increases?

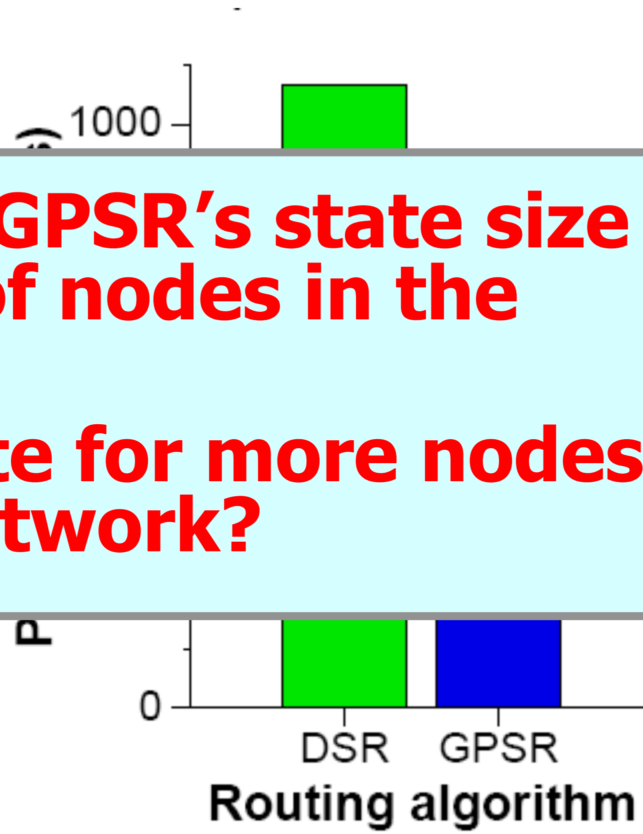
State Size (200; Dense)





State Size (200; Dense)





How would you expect GPSR's state size to change the number of nodes in the network increases?

Why does DSR hold state for more nodes than there are in the network?

Critical Thinking

- Based on the results thus far (indeed, all results in the paper), what do we know about the performance of GPSR's perimeter mode?
  - Would you expect it to be more or less reliable than greedy mode?
  - Would you expect use of perimeter mode to affect path length?

Critical Thinking

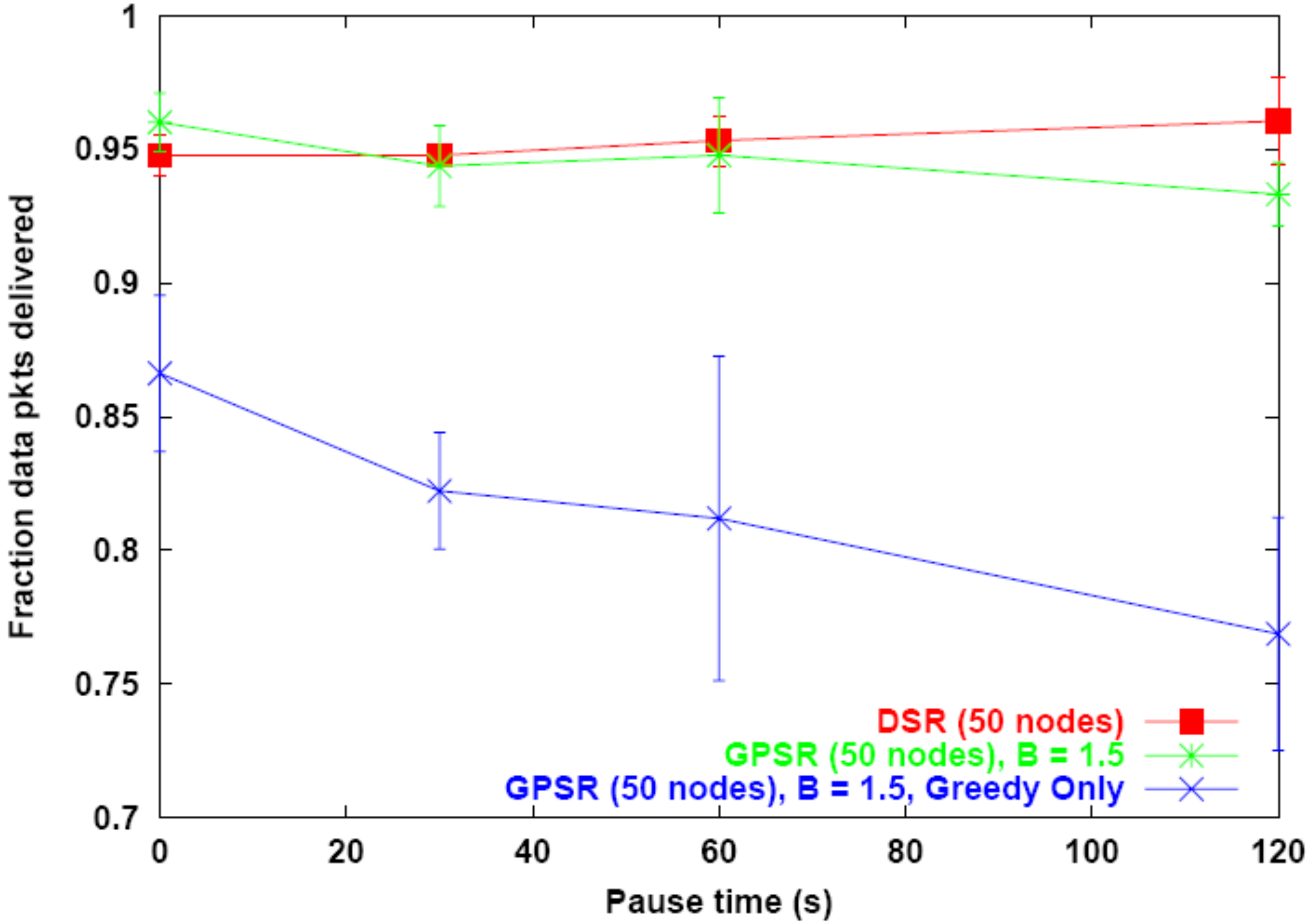
- Based on the results thus far (indeed, all

Evaluation in paper reveals nearly nothing about performance of perimeter mode!

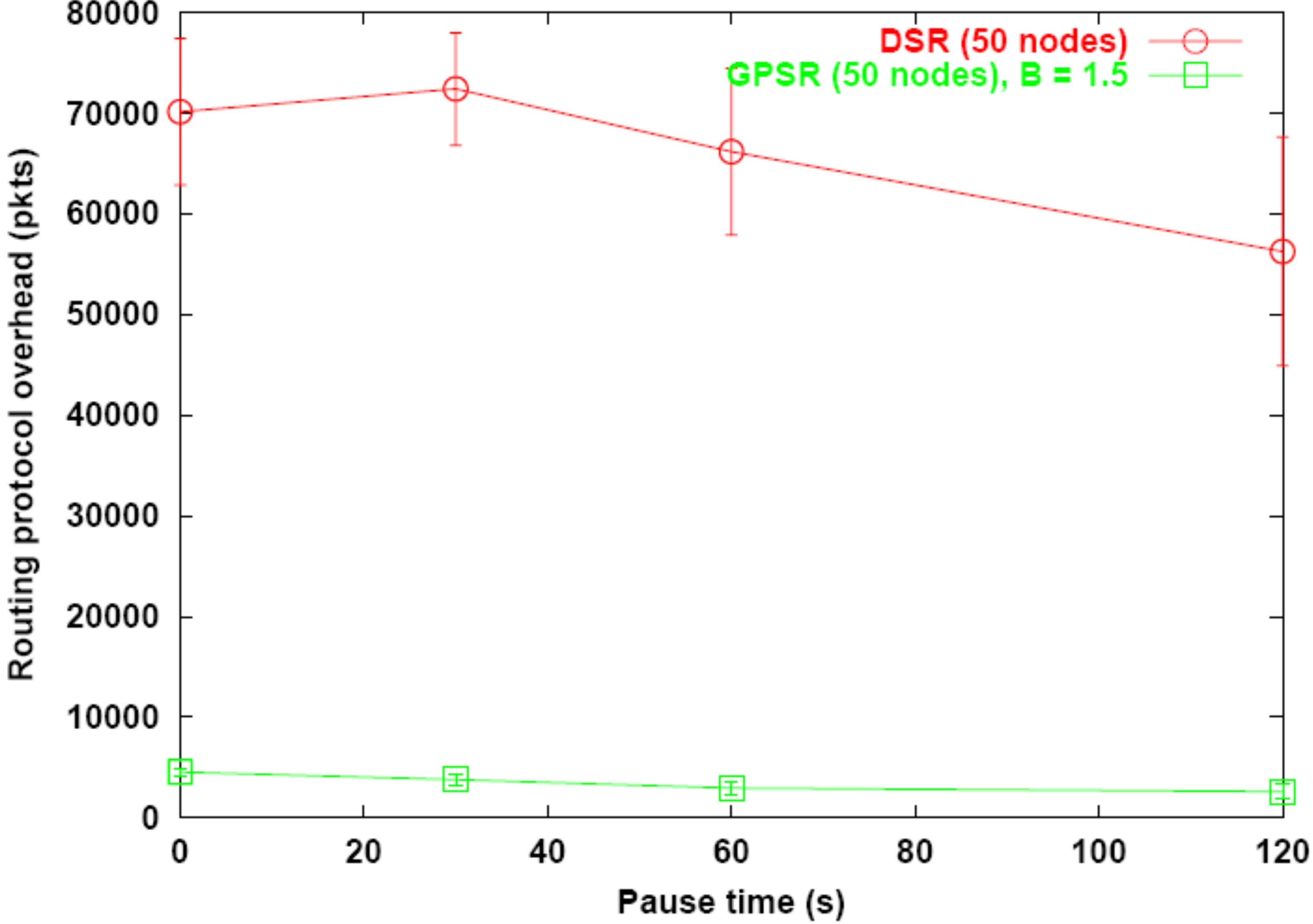
Why doesn't it?

reliable than greedy mode?
  - Would you expect use of perimeter mode to affect path length?

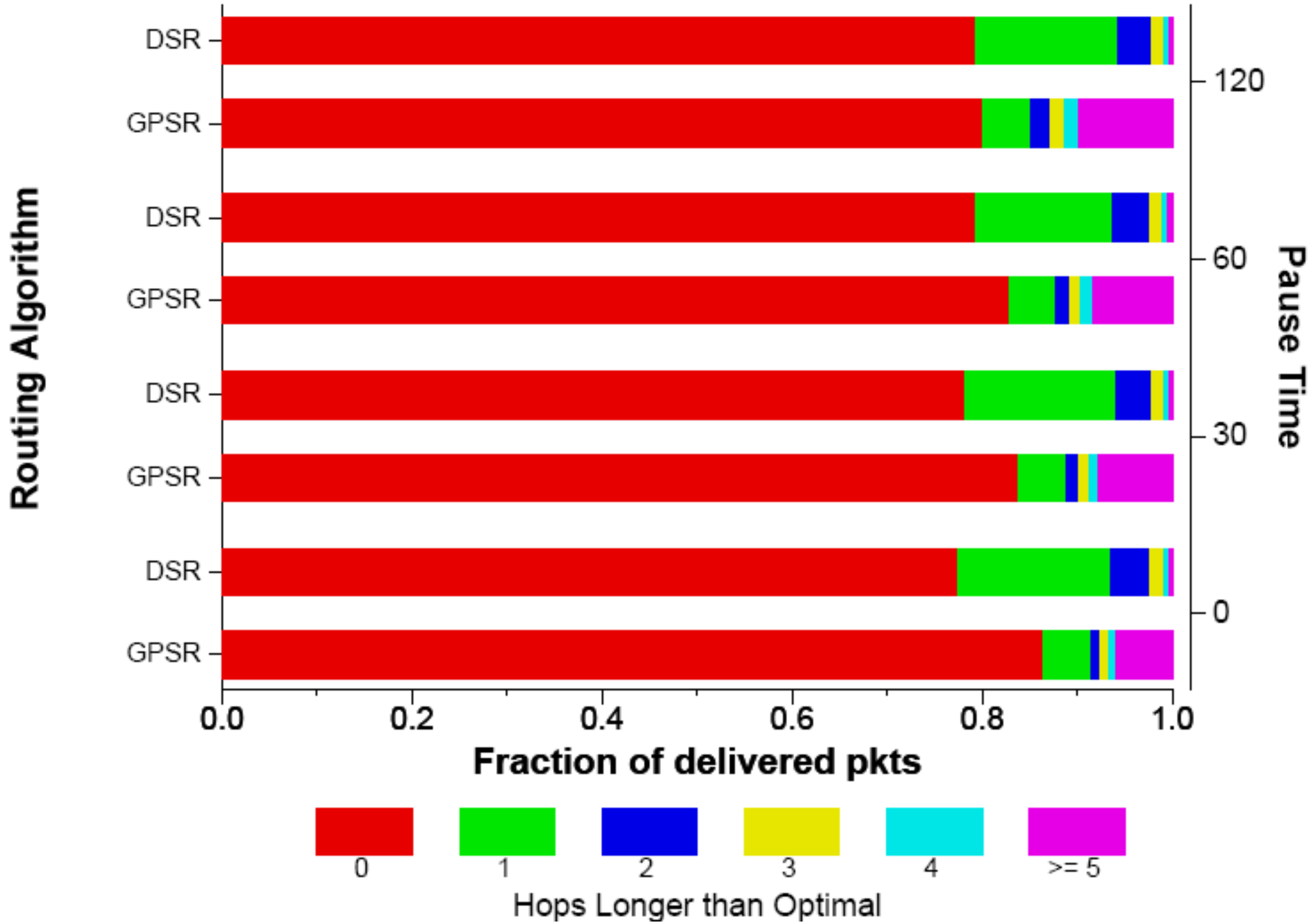
Packet Delivery Success Rate (50; Sparse)



Routing Protocol Overhead (50; Sparse)



Path Length (50; Sparse)



Outline

- Motivation
- Context
- Algorithm
  - Greedy forwarding
  - Graph planarization
  - Perimeter forwarding
- Evaluation in simulation
- Footnotes
  - Open questions
  - Foibles of simulation

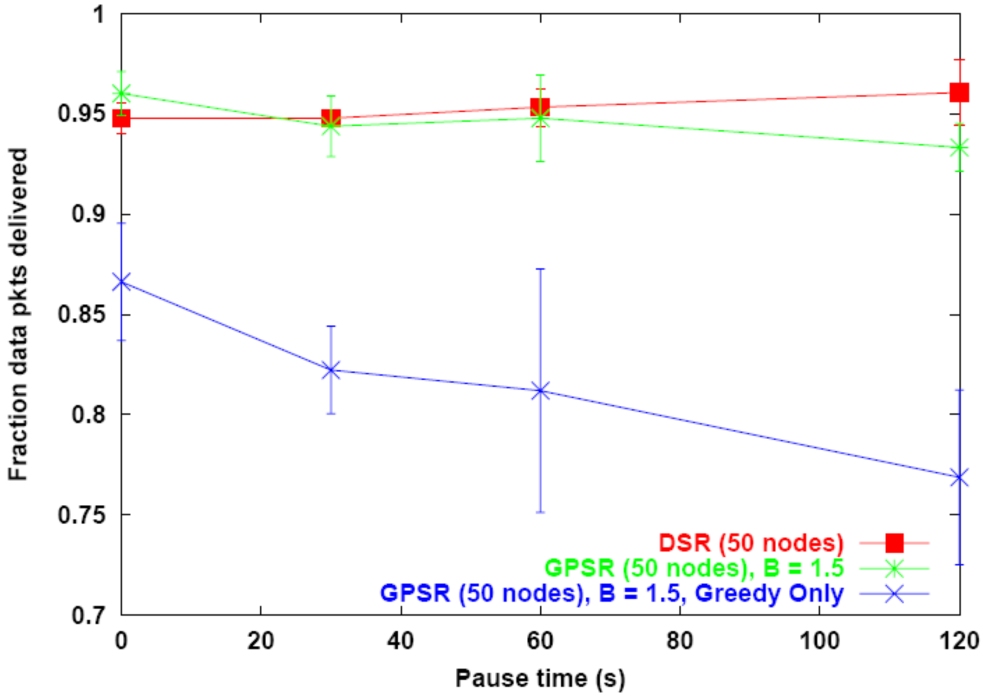
Open Questions

- How to route geographically in 3D?
  - Greedy mode?
  - Perimeter mode?
  - More in CLDP paper…
- Effect of radio-opaque obstacles?
  - More in CLDP paper…
- Effect of position errors?
  - More in CLDP paper… ☺
- “Better” planar graphs than GG, RNG?
  - See [Guibas et al., 2001]
- Name-to-location database, built atop geo routing?
  - See [GLS, Li et al., MobiCom 2000]

Critical Thinking: Why Not Single-Hop to a Base Station?

- High cost of one-hop coverage for all sensors; many base stations
- Transmit power grows as square of distance in free space, worse with obstacles
- Expensive radios not a panacea for single-hop communication
  - "Can you hear me now? How about now?"
  - "Wireless only works around the pool."

Foibles of Simulation



- **Greedy mode works more often as nodes move more rapidly?!**
- Why?

  (Hint: when does greedy forwarding work best?)

Recap:
Scalability via Geography with GPSR

Key scalability properties:

- Small state per router: O(D), not O(N) or O(L) as for shortest-path routing, where D = density (neighbors), N = total nodes, L = total links
- Low routing protocol overhead: each node merely single-hop broadcasts own position periodically
- Approximates shortest paths on dense networks
- Delivers more packets successfully on dynamic topologies than shortest-paths routing protocols