

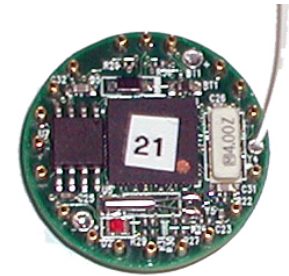
New and Improved Geographic Routing: CLDP

Brad Karp
UCL Computer Science



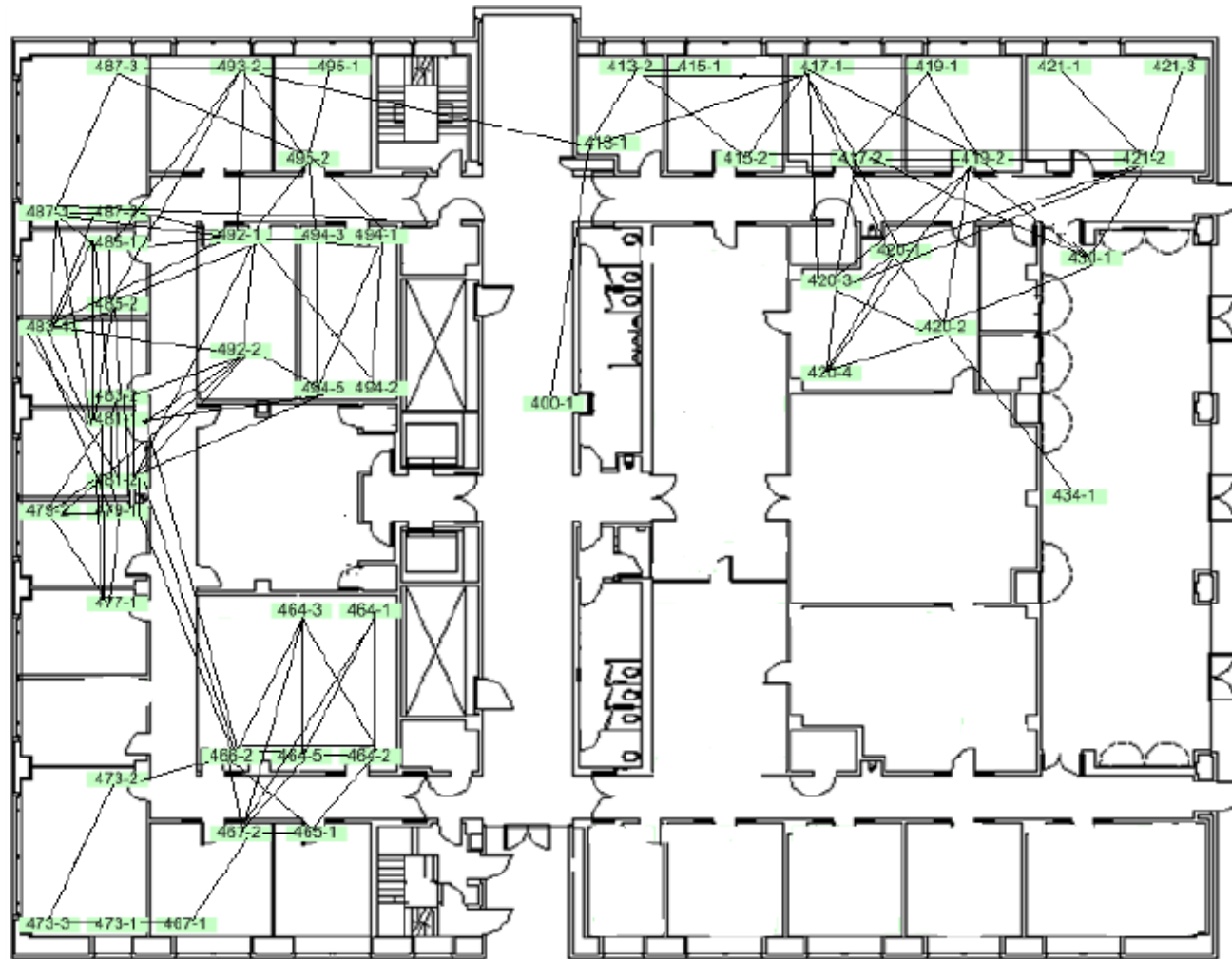
CS M038 / GZ06
21st January, 2011

GPSR: Making it Real

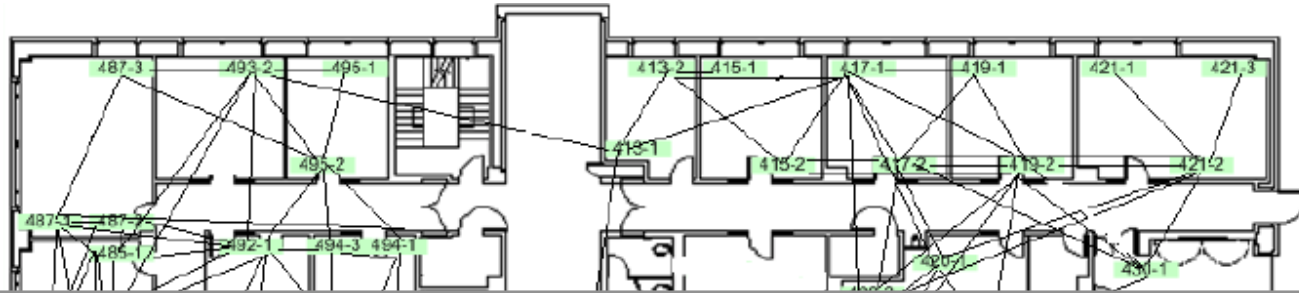


- We implemented **full GPSR for Berkeley mote sensors** [NSDI 2005]
 - 3750 lines of nesC code
 - also includes: simple link thresholding, ARQ
- Deployed on Mica 2 “dot” mote testbeds
 - 23-node, 50-node subsets of 100-node network in office building (Soda Hall; office walls; 433 MHz)
 - 40-node network in office building (Intel Research Berkeley; cubicles; 900 MHz)
- Delivery success workload: 50 packets between all node pairs, serially

50-Node Testbed, Soda Hall



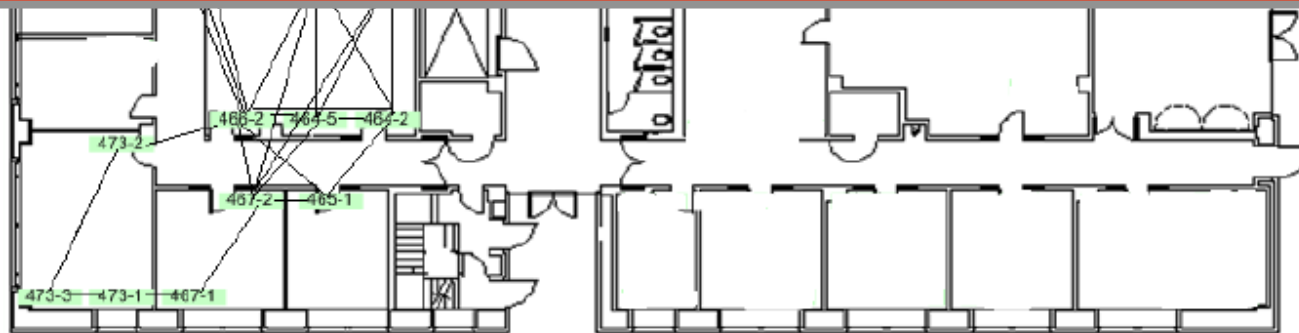
50-Node Testbed, Soda Hall



GAME OVER

Only 68.2% of node pairs connected!!

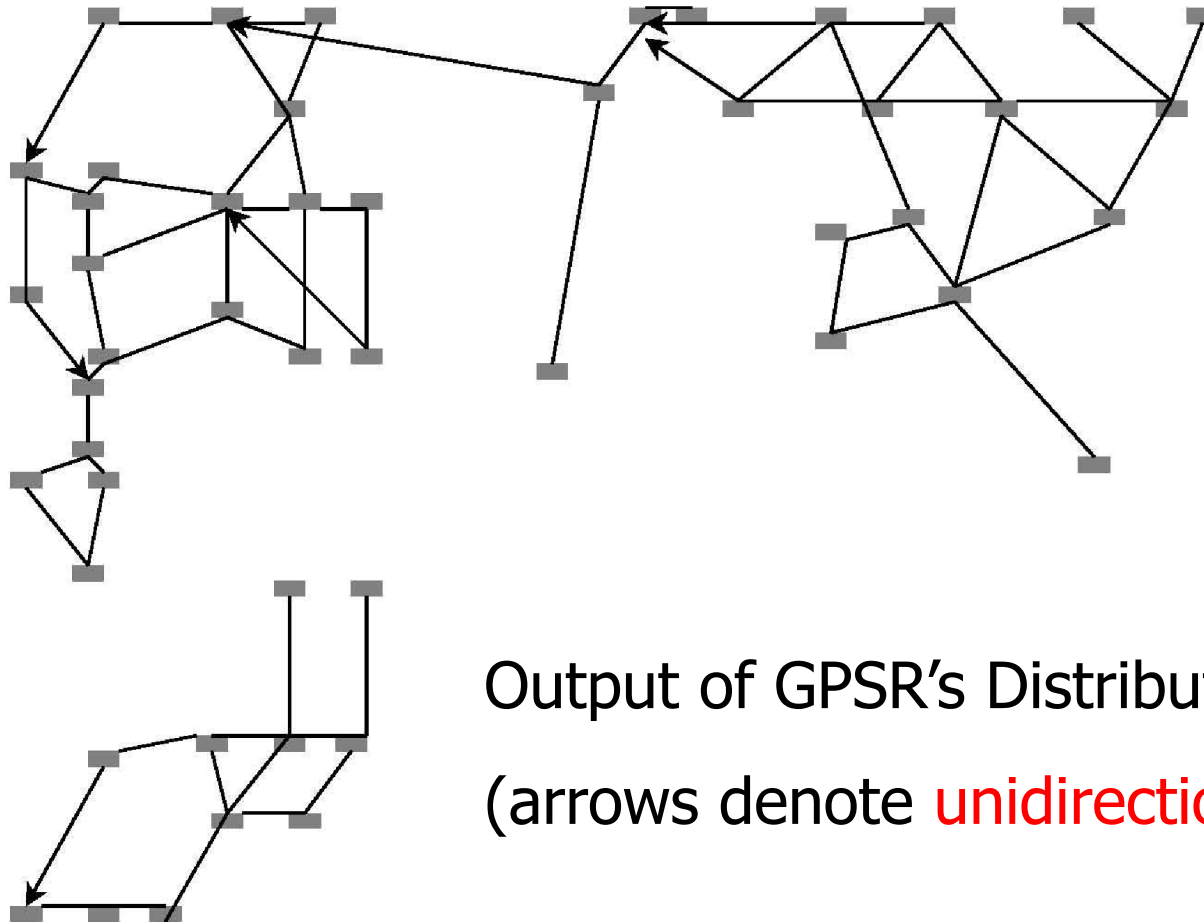
What's going on here?!



Outline

- Motivation
- **Analysis of GPSR Routing Pathologies**
- CLDP Algorithm
 - Assumptions and Goals
 - Basic, Serialized CLDP
 - Lazy Locking for Concurrent Probing
- Evaluation in Simulation and Deployment
- Geographic Routing Summary

Planar, but Partitioned



Output of GPSR's Distributed GG
(arrows denote **unidirectional links**)

Assumptions Redux

- 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

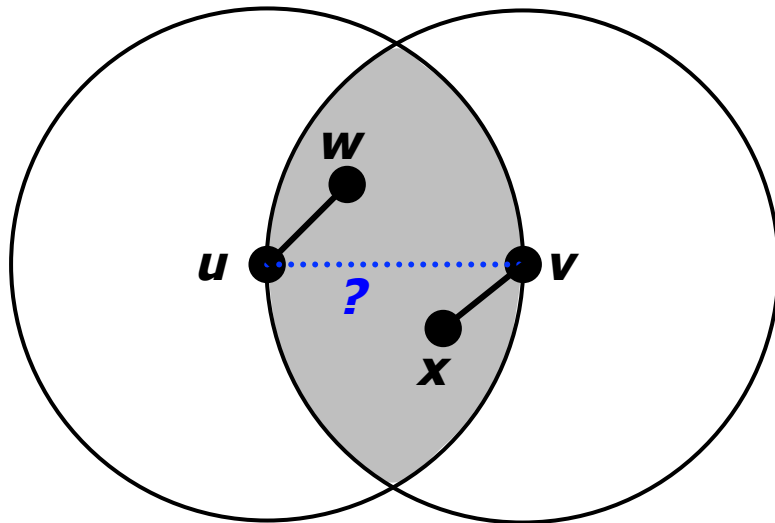
Assumptions Redux

- Bi-directional radio links (unidirectional links may be *blacklisted*)

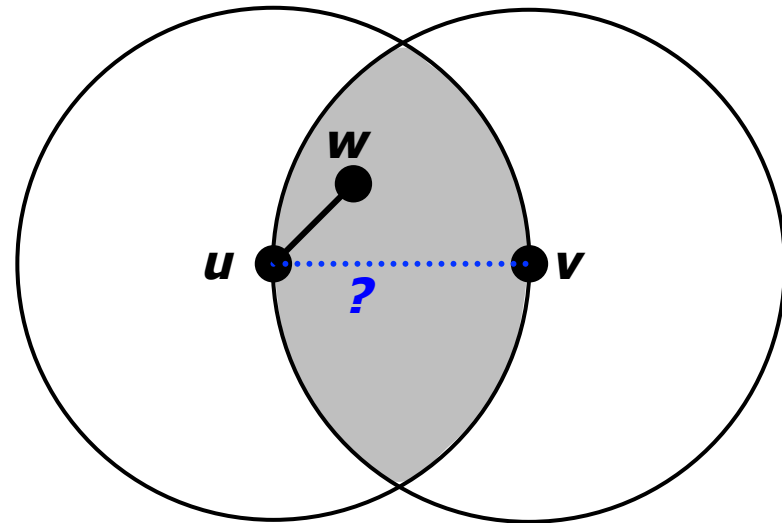
Absorption, reflection (multi-path), interference, antenna orientation differences, &c., lead to *non-unit graphs*.

Non-uniformity of radio ranges increasingly noted [Biswas, Morris 2003]; [Ganesan *et al.* 2002]; [Zhao, Govindan 2003]

Planarization Pathologies

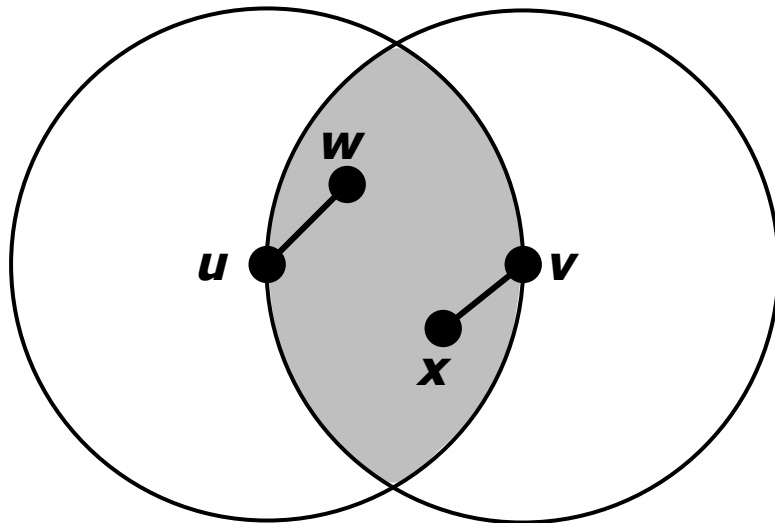


Partitioned RNG

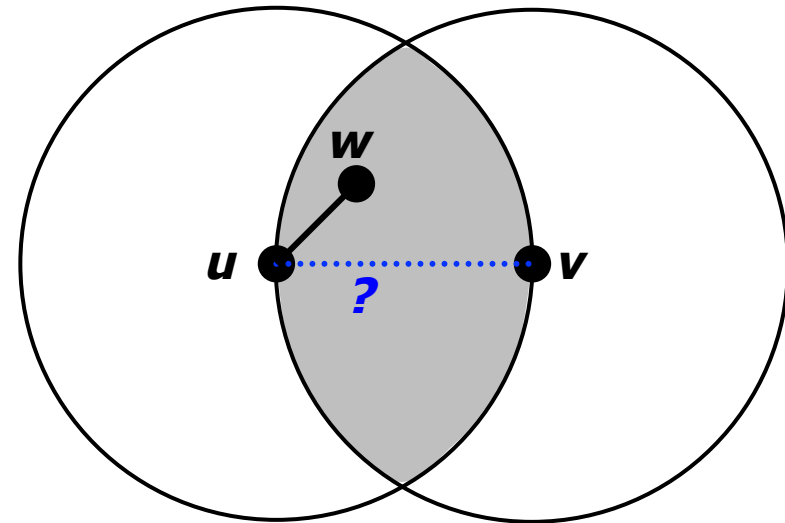


RNG w/Unidirectional Link

Planarization Pathologies

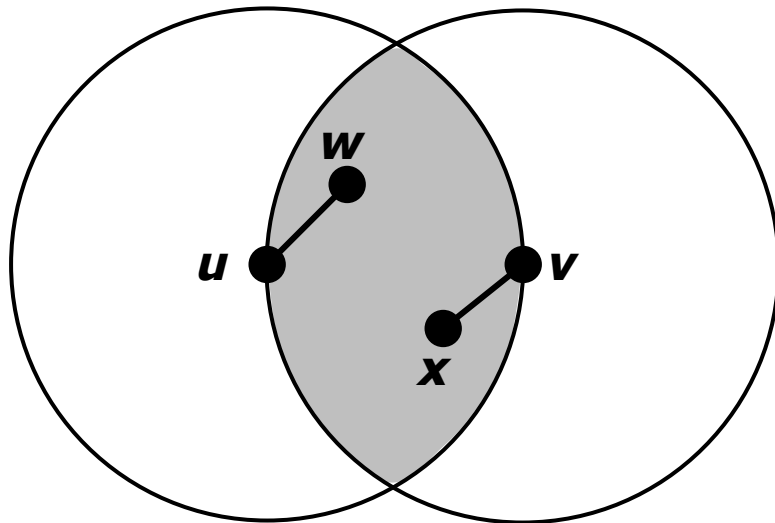


Partitioned RNG

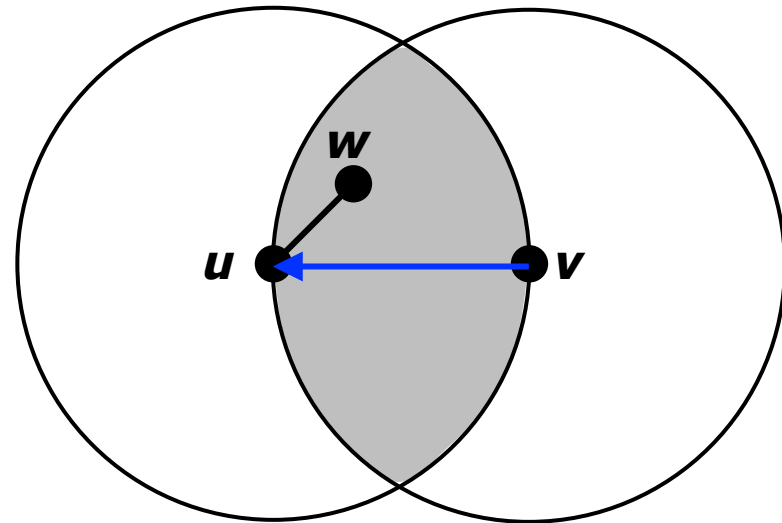


RNG w/Unidirectional Link

Planarization Pathologies



Partitioned RNG



RNG w/Unidirectional Link

Planarization Pathologies



Non-unit graphs produce:

- **partitioned RNG, GG**
- **asymmetric links in RNG, GG**

Localization errors produce:

- **non-planar RNG, GG**
- **asymmetric links in RNG, GG**

Outline

- Motivation
- Analysis of GPSR Routing Pathologies
- **CLDP Algorithm**
 - **Assumptions and Goals**
 - Basic, Serialized CLDP
 - Lazy Locking for Concurrent Probing
- Evaluation in Simulation and Deployment
- Geographic Routing Summary

Cross-Link Detection Protocol (CLDP): Assumptions and Goals

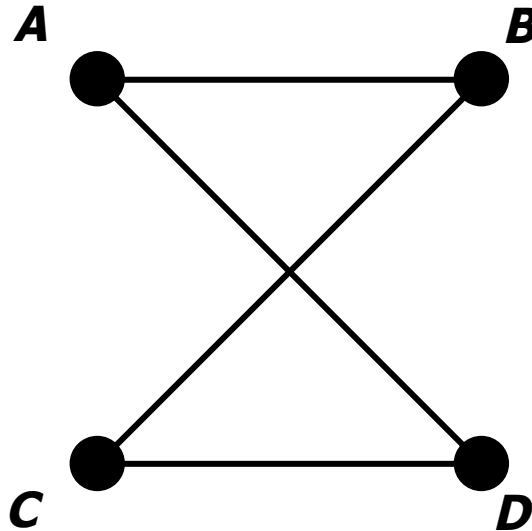
- Assumptions, revised:
 - nodes know their own positions in 2D coordinate system
 - connected graph
 - bidirectional links (*cf.* blacklisting)
 - **no assumption whatsoever** about structure of graph
- Seek a “planarization” algorithm that:
 - never partitions graph
 - always produces a **routable** graph; one on which GPSR routing never fails (**may contain crossings!**)

CLDP Sketch

Nodes explicitly probe their own links to detect crossings by other links

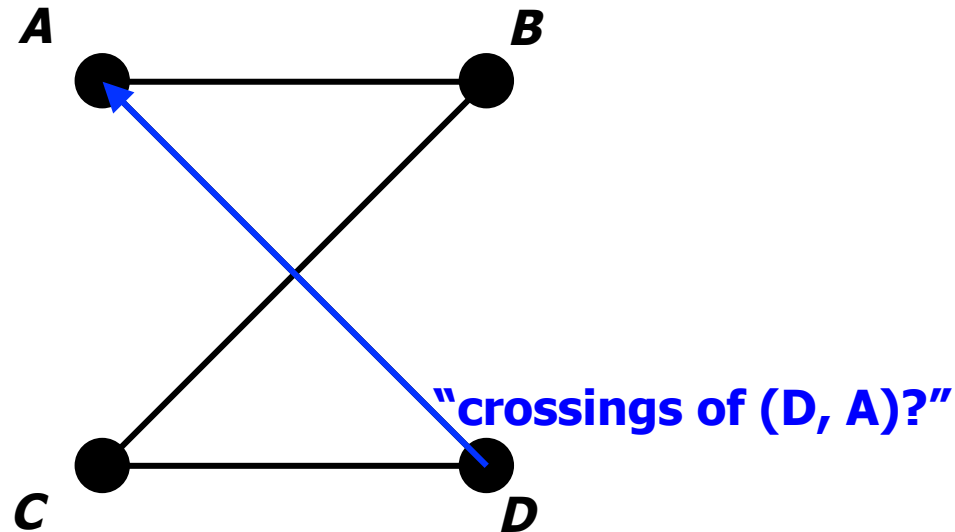
- probe packet **follows right-hand rule**; carries locations of candidate link endpoints
- probe packet **records first crossing link** it encounters en route
- one of two crossing links **"eliminated"** when probe returns to originator
 - originator may mark candidate link *unroutable* OR
 - request remote crossing link be marked **unroutable**
- probe packets **only traverse routable links**

CLDP: A Simple Example



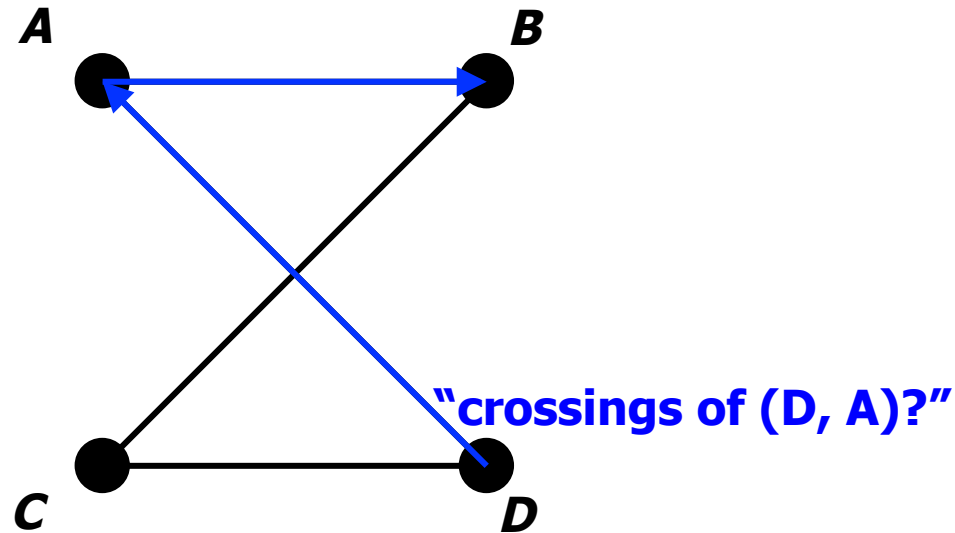
- All links initially marked “routable”
- Detected crossings result in transitions to “unroutable” (by D, or by B or C)

CLDP: A Simple Example



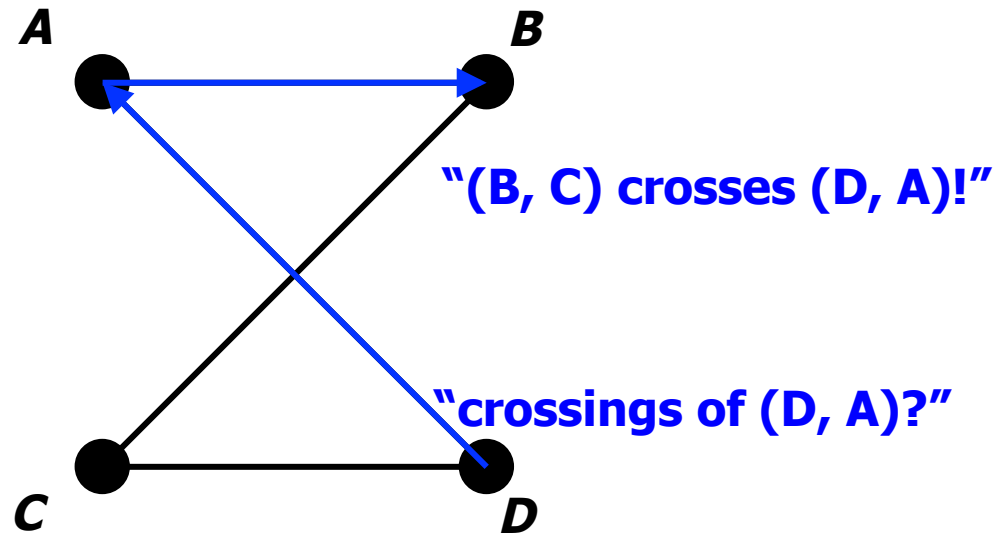
- All links initially marked "routable"
- Detected crossings result in transitions to "unroutable" (by D, or by B or C)

CLDP: A Simple Example



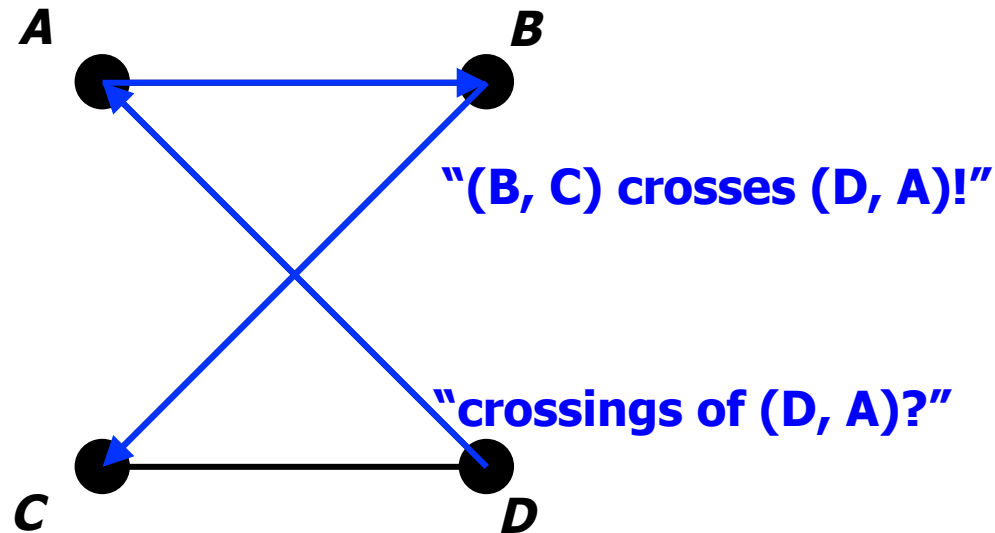
- All links initially marked "routable"
- Detected crossings result in transitions to "unroutable" (by D, or by B or C)

CLDP: A Simple Example



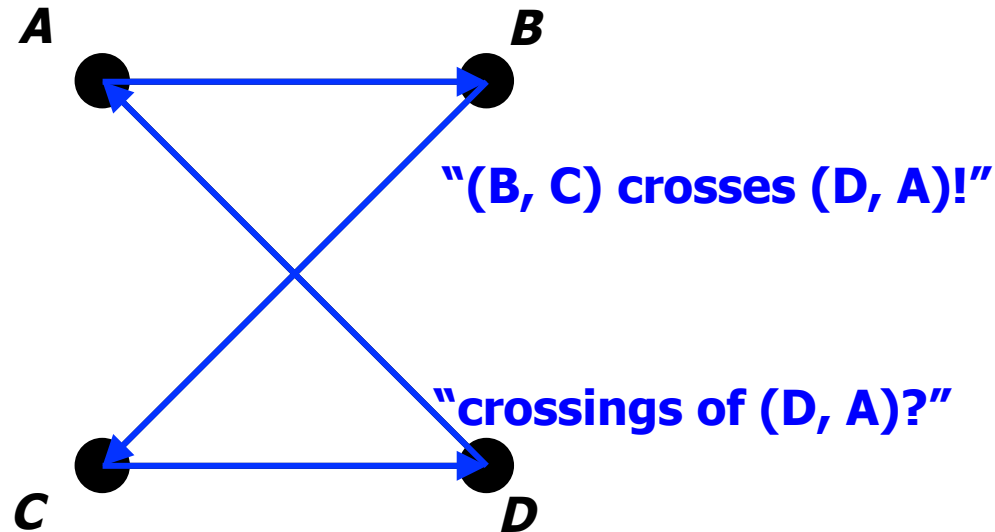
- All links initially marked "routable"
- Detected crossings result in transitions to "unroutable" (by D, or by B or C)

CLDP: A Simple Example



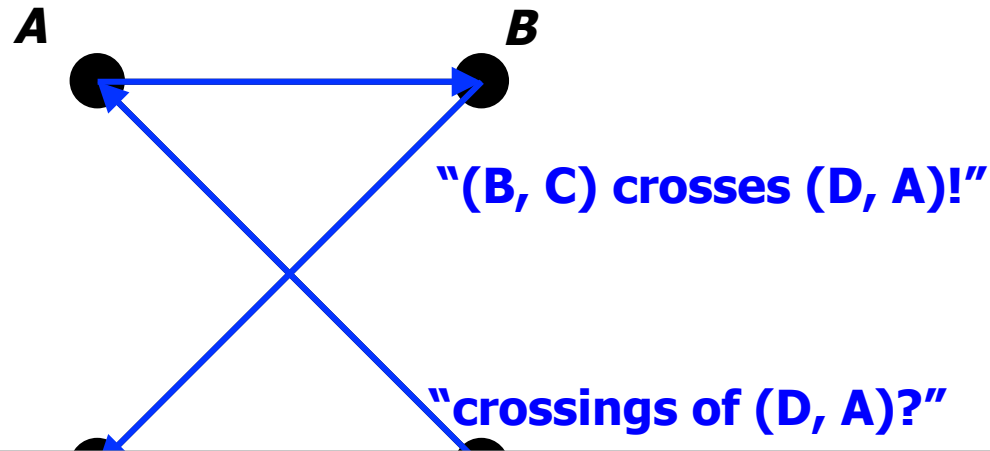
- All links initially marked "routable"
- Detected crossings result in transitions to "unroutable" (by D, or by B or C)

CLDP: A Simple Example



- All links initially marked "routable"
- Detected crossings result in transitions to "unroutable" (by D, or by B or C)

CLDP: A Simple Example

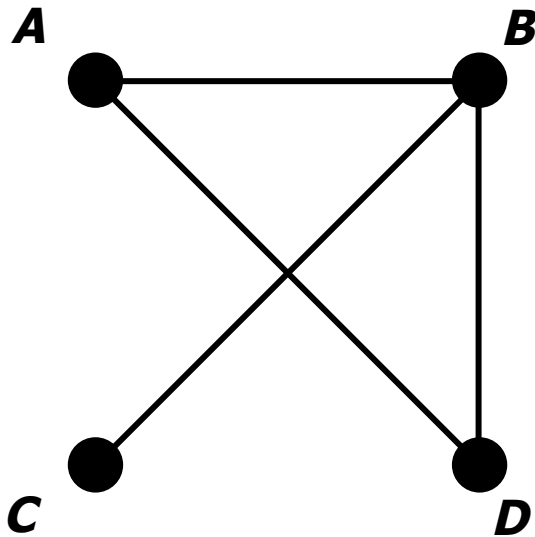


In a dense wireless network, most perimeters short (3 hops); **most probes traverse short paths**

- Detected crossings result in transitions to "unroutable" (by D, or by B or C)

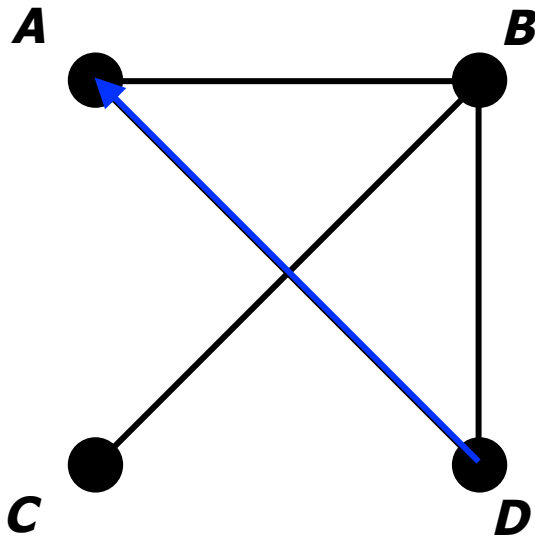
CLDP and Cul-de-sacs

- Cul-de-sacs give rise to links that cannot be eliminated without partitioning graph
- Not all {edges, crossings} can be eliminated!



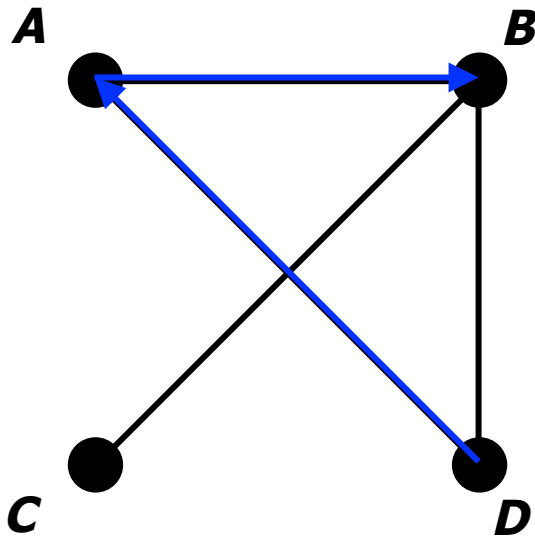
CLDP and Cul-de-sacs

- Cul-de-sacs give rise to links that cannot be eliminated without partitioning graph
- Not all {edges, crossings} can be eliminated!



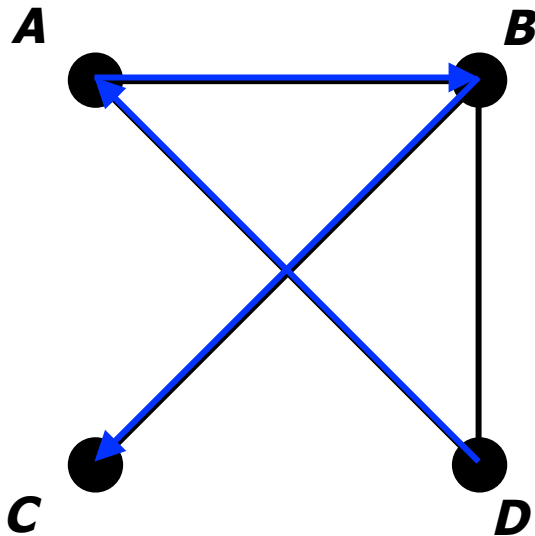
CLDP and Cul-de-sacs

- Cul-de-sacs give rise to links that cannot be eliminated without partitioning graph
- Not all {edges, crossings} can be eliminated!



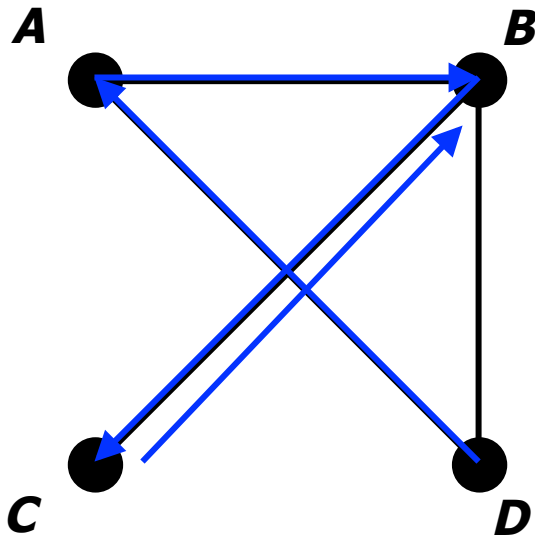
CLDP and Cul-de-sacs

- Cul-de-sacs give rise to links that cannot be eliminated without partitioning graph
- Not all {edges, crossings} can be eliminated!



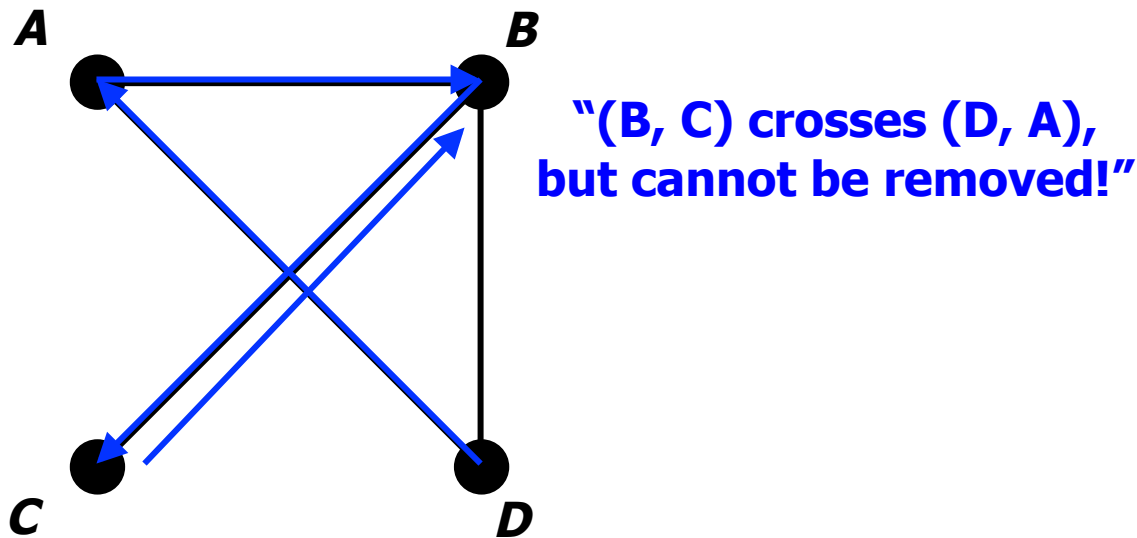
CLDP and Cul-de-sacs

- Cul-de-sacs give rise to links that cannot be eliminated without partitioning graph
- Not all {edges, crossings} can be eliminated!



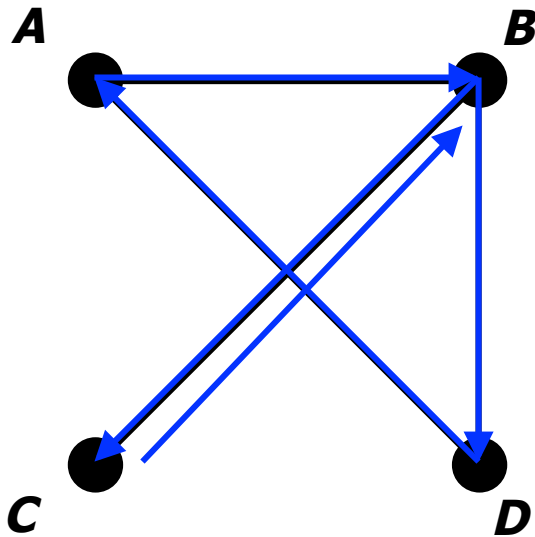
CLDP and Cul-de-sacs

- Cul-de-sacs give rise to links that cannot be eliminated without partitioning graph
- Not all {edges, crossings} can be eliminated!



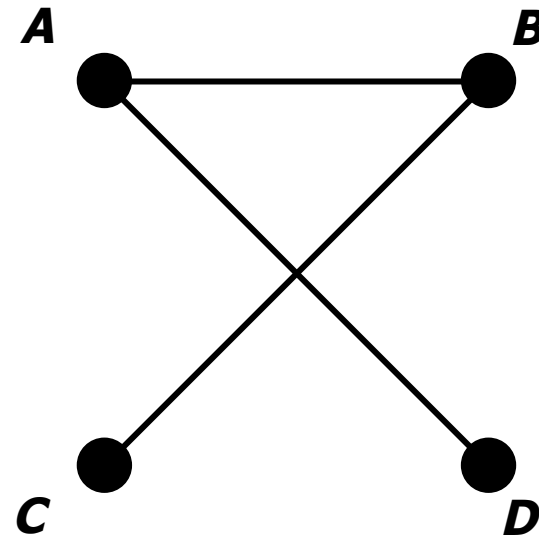
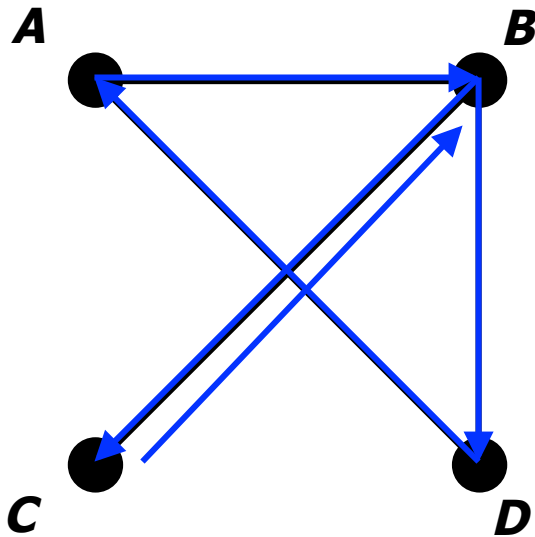
CLDP and Cul-de-sacs

- Cul-de-sacs give rise to links that cannot be eliminated without partitioning graph
- Not all {edges, crossings} can be eliminated!



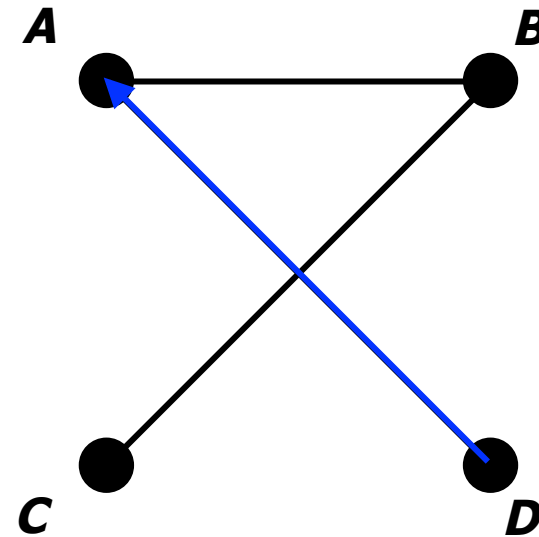
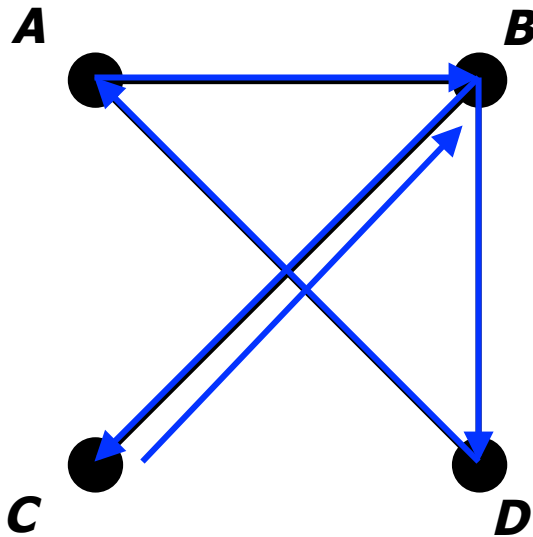
CLDP and Cul-de-sacs

- Cul-de-sacs give rise to links that cannot be eliminated without partitioning graph
- Not all {edges, crossings} can be eliminated!



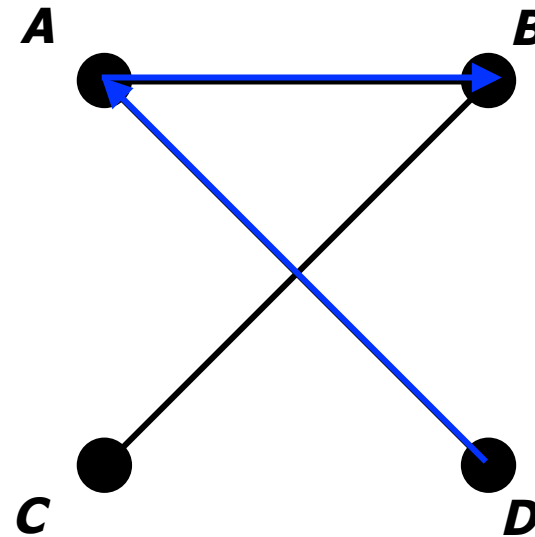
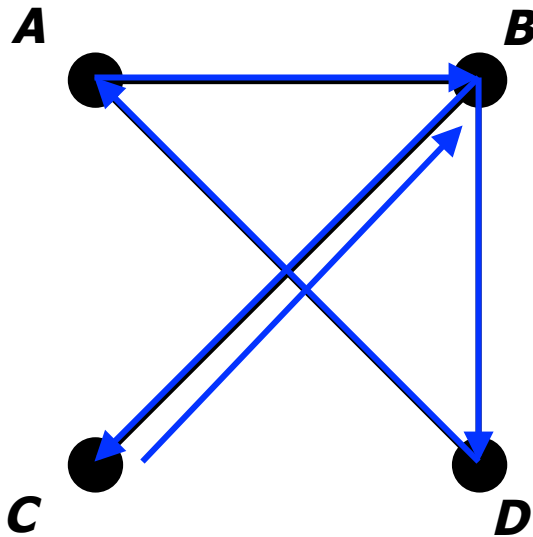
CLDP and Cul-de-sacs

- Cul-de-sacs give rise to links that cannot be eliminated without partitioning graph
- Not all {edges, crossings} can be eliminated!



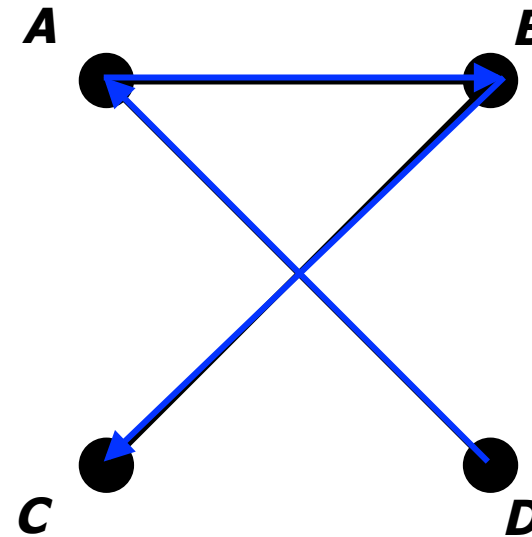
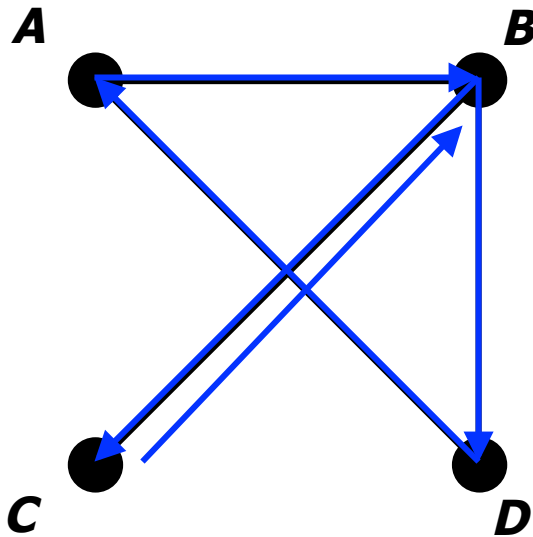
CLDP and Cul-de-sacs

- Cul-de-sacs give rise to links that cannot be eliminated without partitioning graph
- Not all {edges, crossings} can be eliminated!



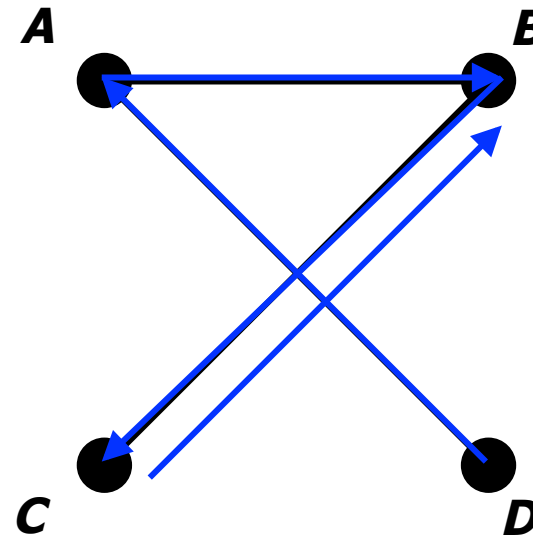
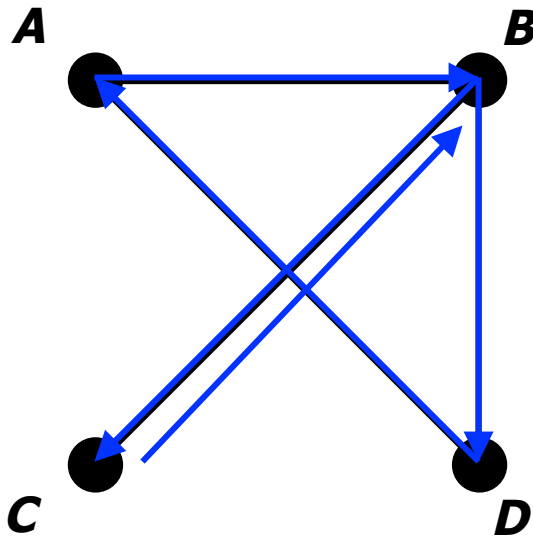
CLDP and Cul-de-sacs

- Cul-de-sacs give rise to links that cannot be eliminated without partitioning graph
- Not all {edges, crossings} can be eliminated!



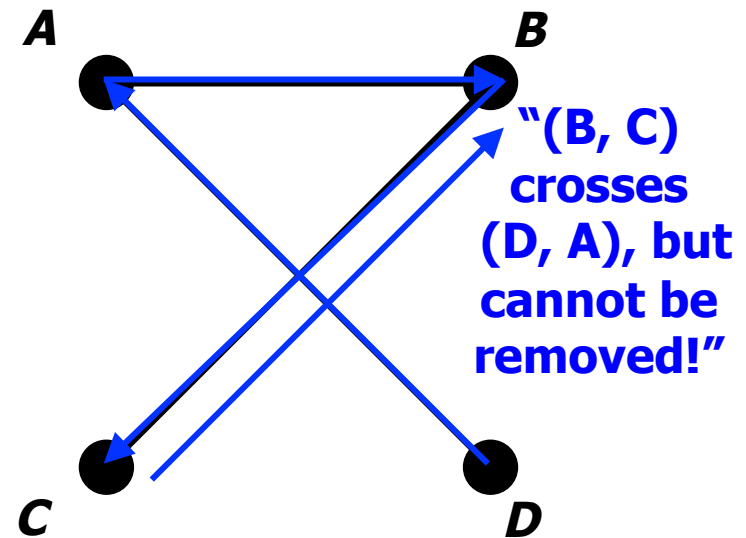
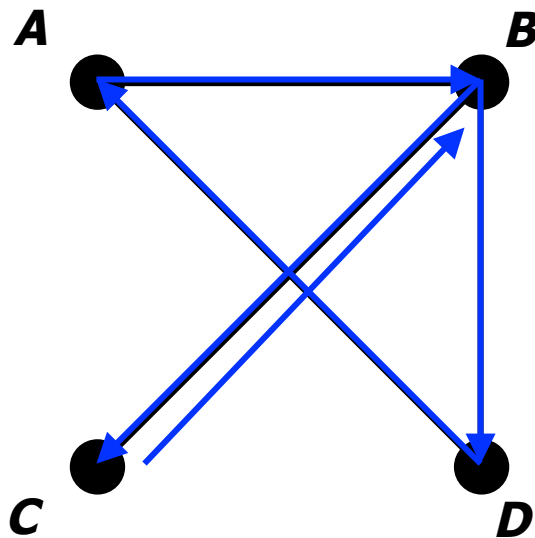
CLDP and Cul-de-sacs

- Cul-de-sacs give rise to links that cannot be eliminated without partitioning graph
- Not all {edges, crossings} can be eliminated!



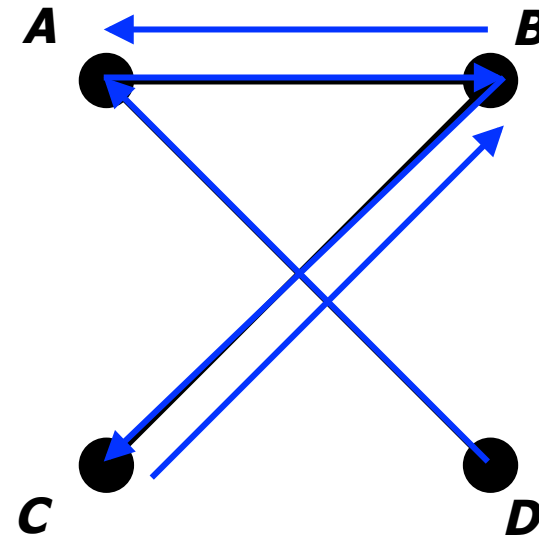
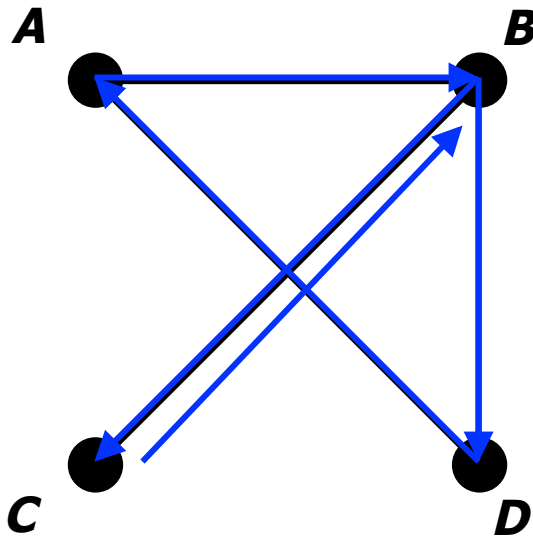
CLDP and Cul-de-sacs

- Cul-de-sacs give rise to links that cannot be eliminated without partitioning graph
- Not all {edges, crossings} can be eliminated!



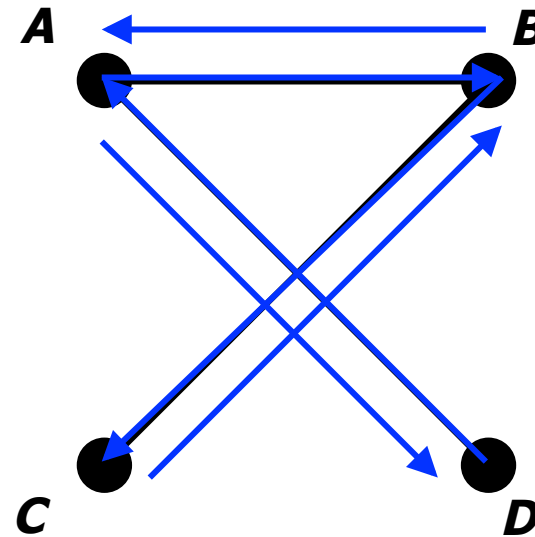
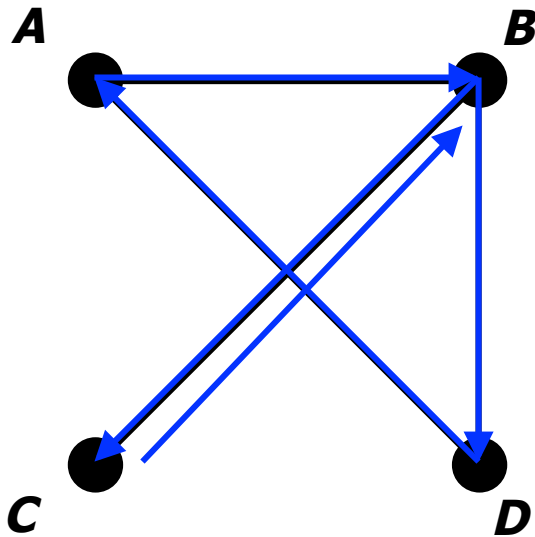
CLDP and Cul-de-sacs

- Cul-de-sacs give rise to links that cannot be eliminated without partitioning graph
- Not all {edges, crossings} can be eliminated!



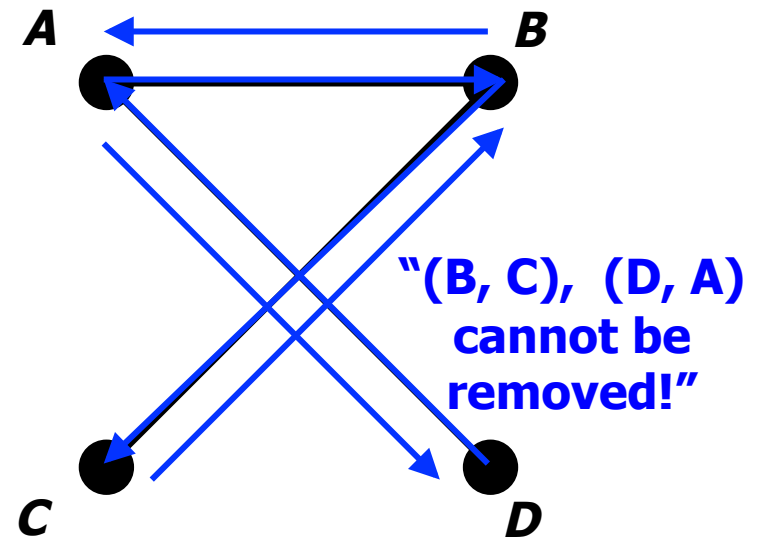
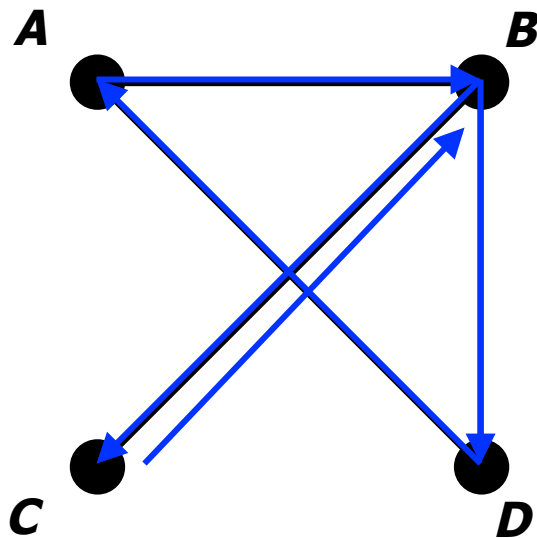
CLDP and Cul-de-sacs

- Cul-de-sacs give rise to links that cannot be eliminated without partitioning graph
- Not all {edges, crossings} can be eliminated!



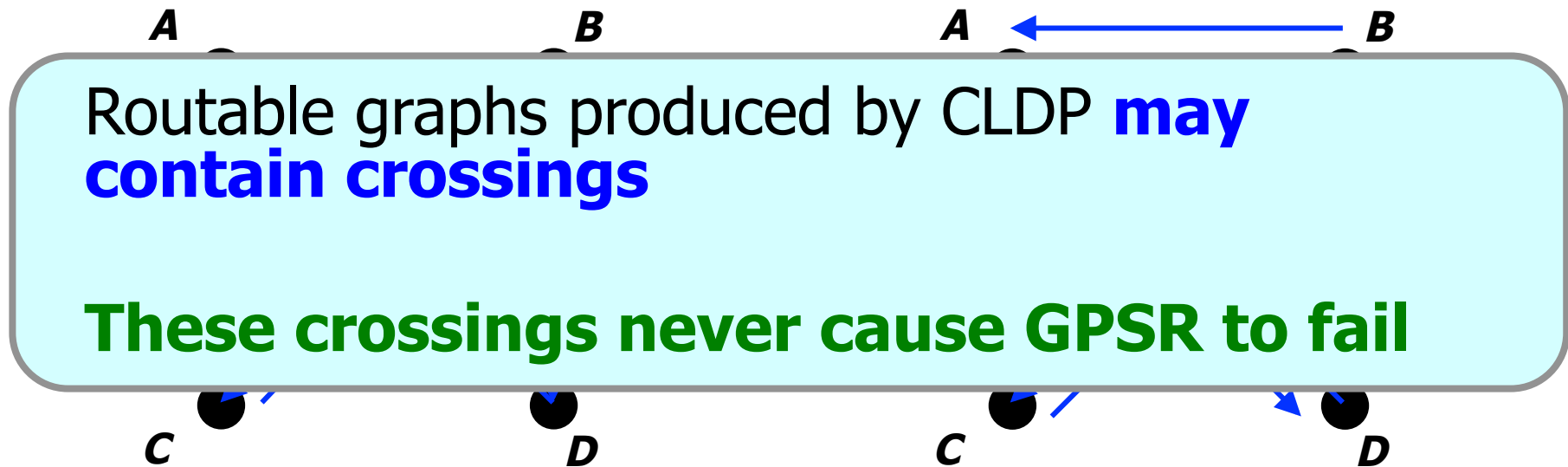
CLDP and Cul-de-sacs

- Cul-de-sacs give rise to links that cannot be eliminated without partitioning graph
- Not all {edges, crossings} can be eliminated!



CLDP and Cul-de-sacs

- Cul-de-sacs give rise to links that cannot be eliminated without partitioning graph
- Not all {edges, crossings} can be eliminated!



Summary: CLDP Protocol

- If link L probed, crossing link L' found:
 - both L and L' removable: **remove L**
 - L removable, L' not removable: **remove L**
 - L not removable, L' removable: **remove L'**
 - neither L nor L' removable: **remove no link**
- **Links reprobbed periodically**, to maintain correct graph under network dynamics
- **Locking protocol** supports concurrent probes

Summary: CLDP Protocol

- If link L probed, crossing link L' found:
 - both L and L' removable: **remove L**

Given any **static, connected graph**, CLDP **always produces a graph on which GPSR succeeds, for all node pairs**

GPSR+CLDP: provably correct geographic routing for any network, wired or wireless

- **Locking protocol** supports concurrent probes

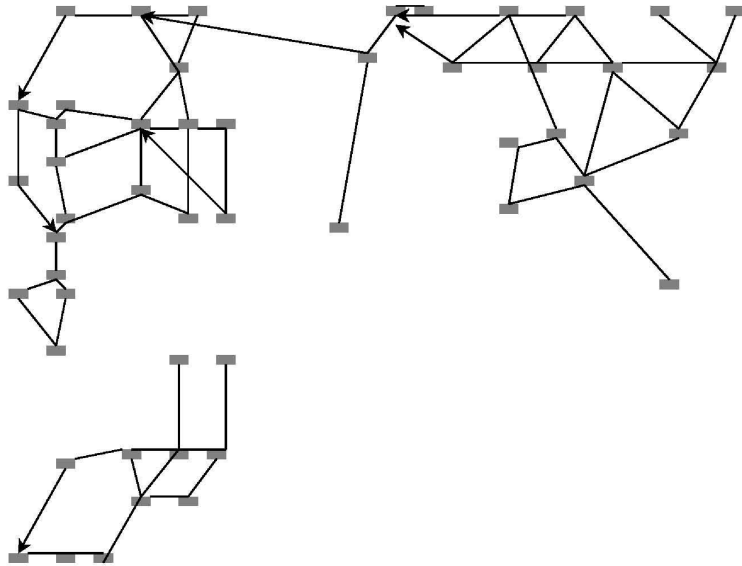
Concurrent Probes: Lazy Locking

- Phase 1: **Probe**
 - If link not crossed, done
 - If link not removable, done
- Phase 2: **Commit (lock)**
 - Locked links drop probes
 - Only one commit can cross a locked link
 - Upon return to originator...
- Phase 3: **Unlock**
 - Mark link unroutable

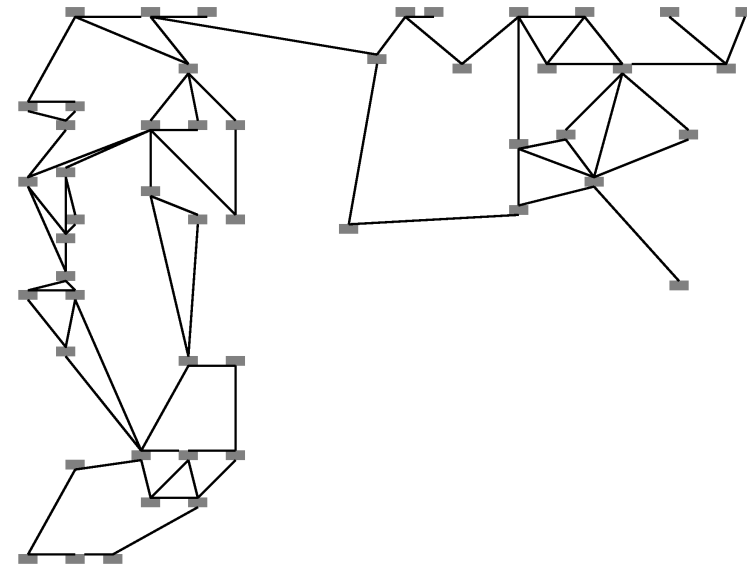
Outline

- Motivation
- Analysis of GPSR Routing Pathologies
- CLDP Algorithm
 - Assumptions and Goals
 - Basic, Serialized CLDP
 - Lazy Locking for Concurrent Probing
- **Evaluation in Simulation and Deployment**
- Geographic Routing Summary

Meanwhile, Back in Soda Hall...



GG



CLDP

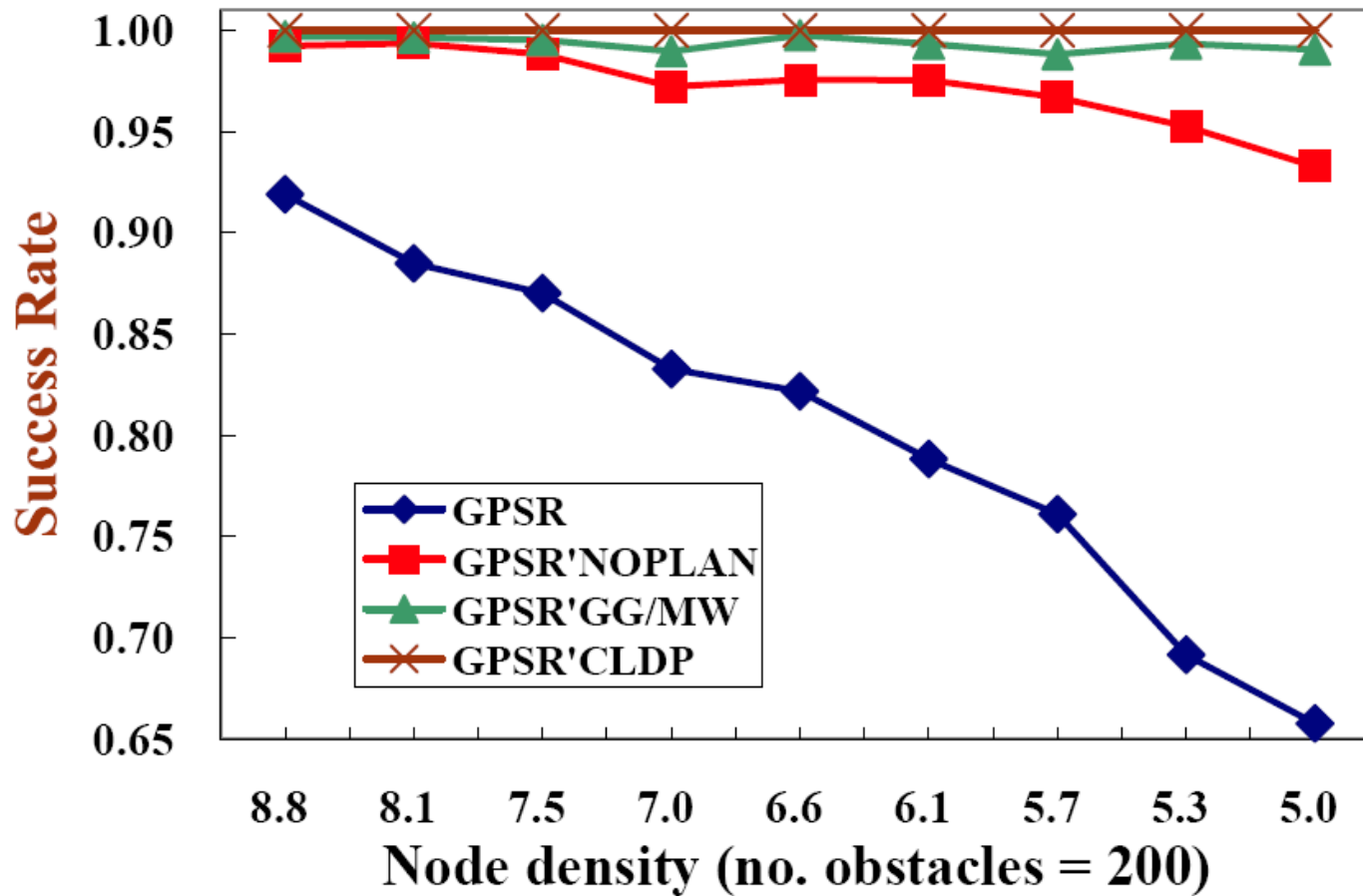
Meanwhile, Back in Soda Hall...



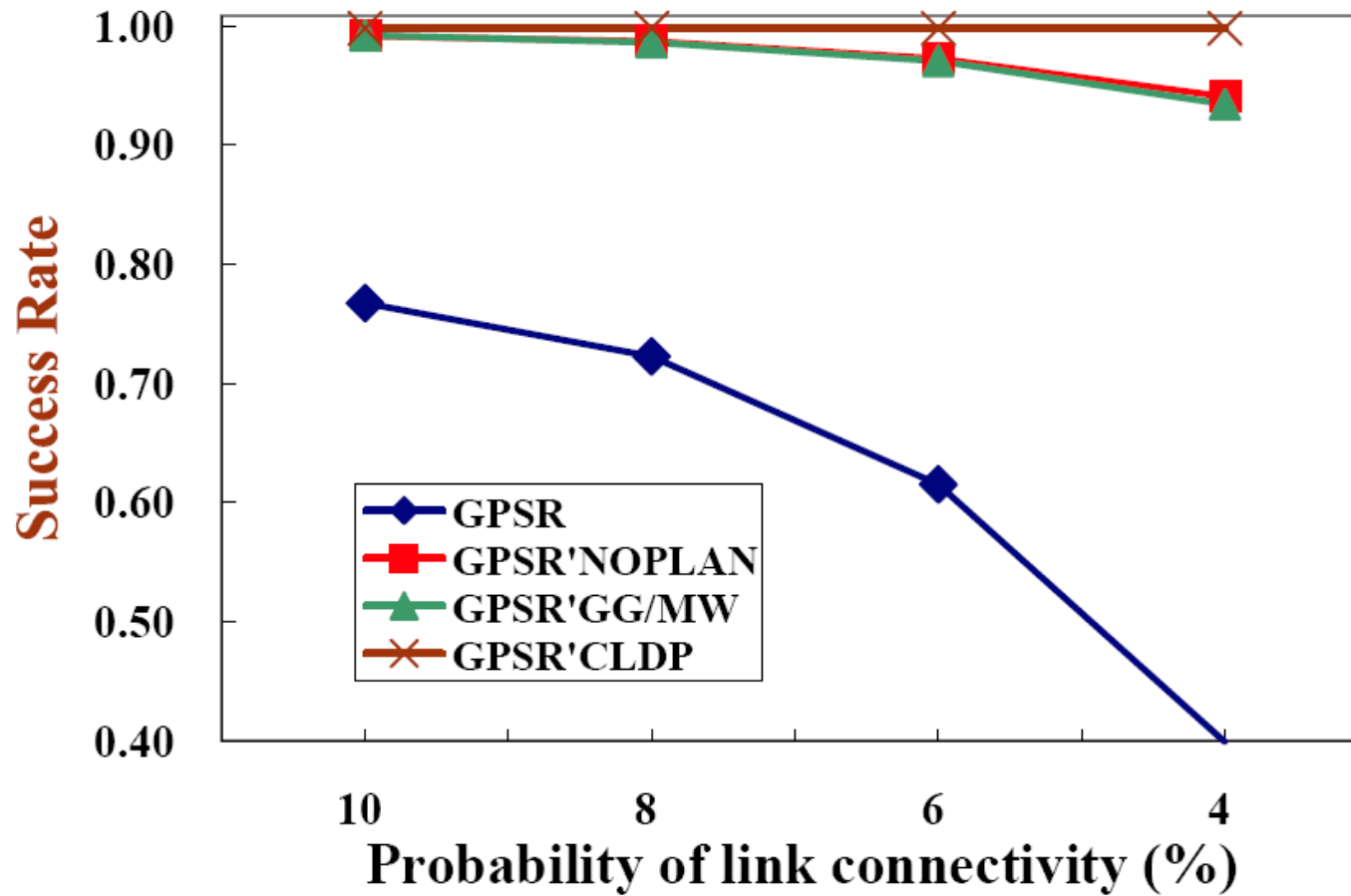
CLDP mote implementation: ~750 lines nesC
Deployed on same three mote testbeds

Broader evaluation in simulation (Bernoulli graphs, wireless networks with obstacles, &c.)

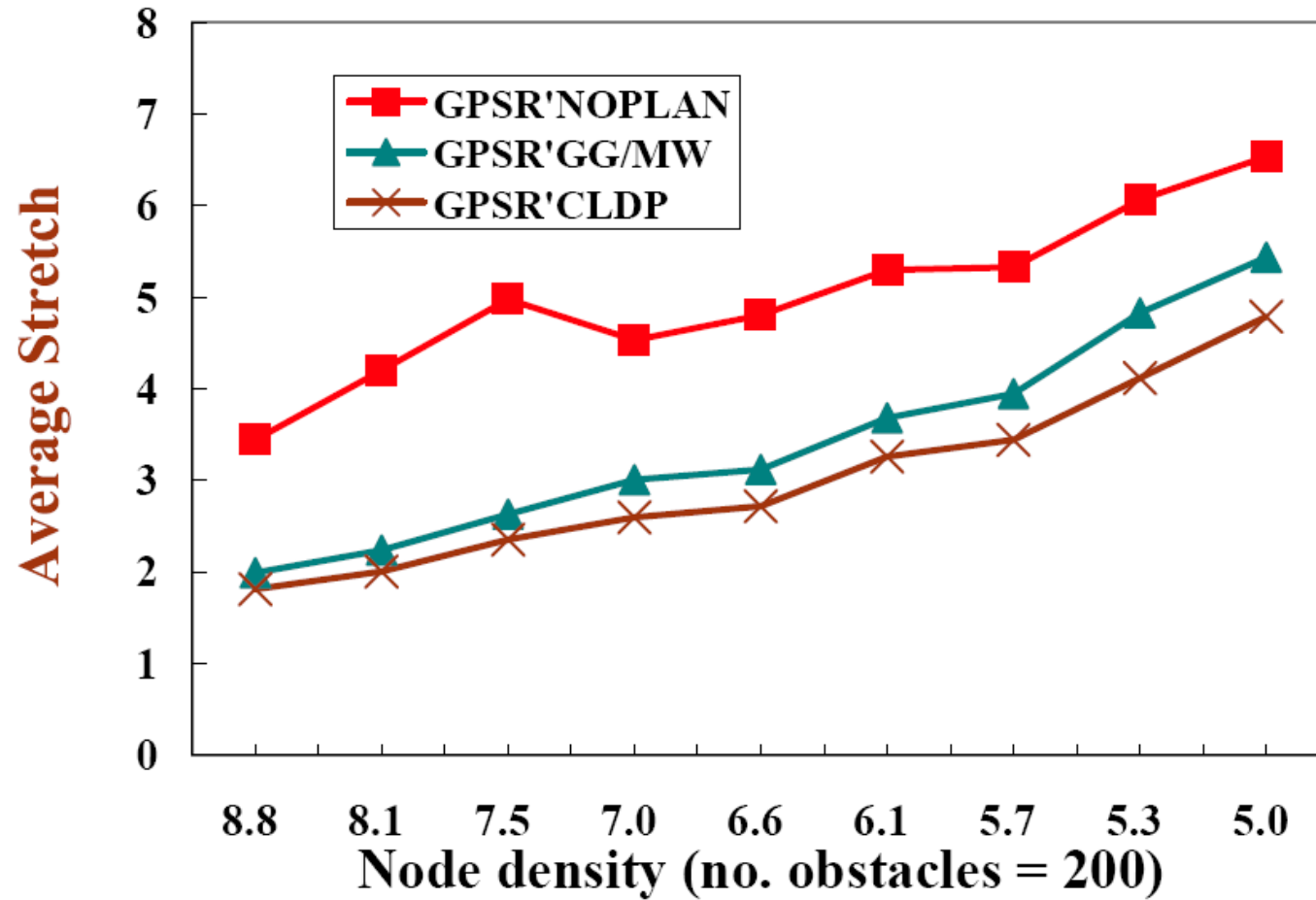
CLDP: Packet Delivery Success Rate (200 Nodes; 200 Obstacles)



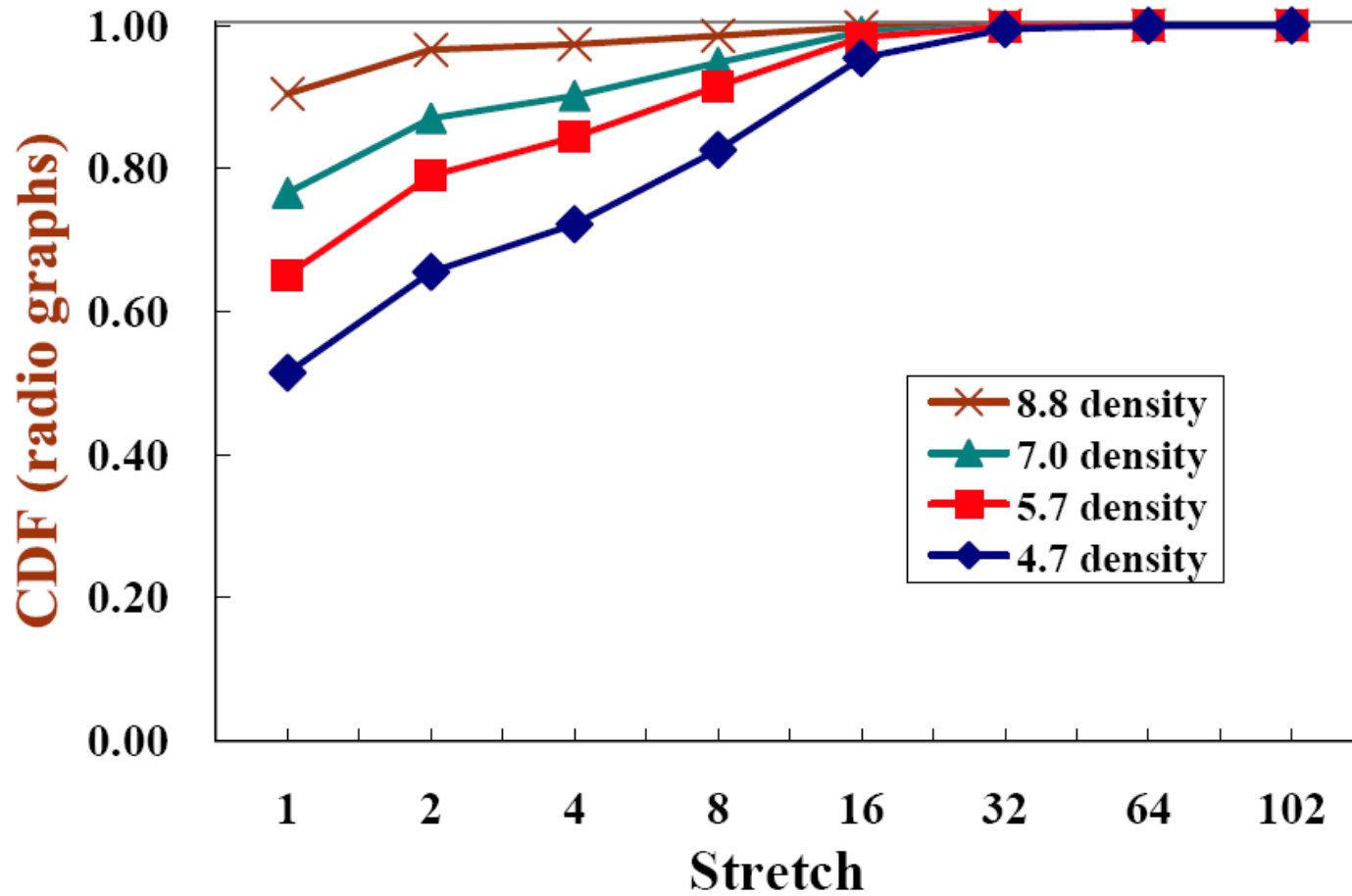
CLDP: Packet Delivery Success Rate (200 Nodes)



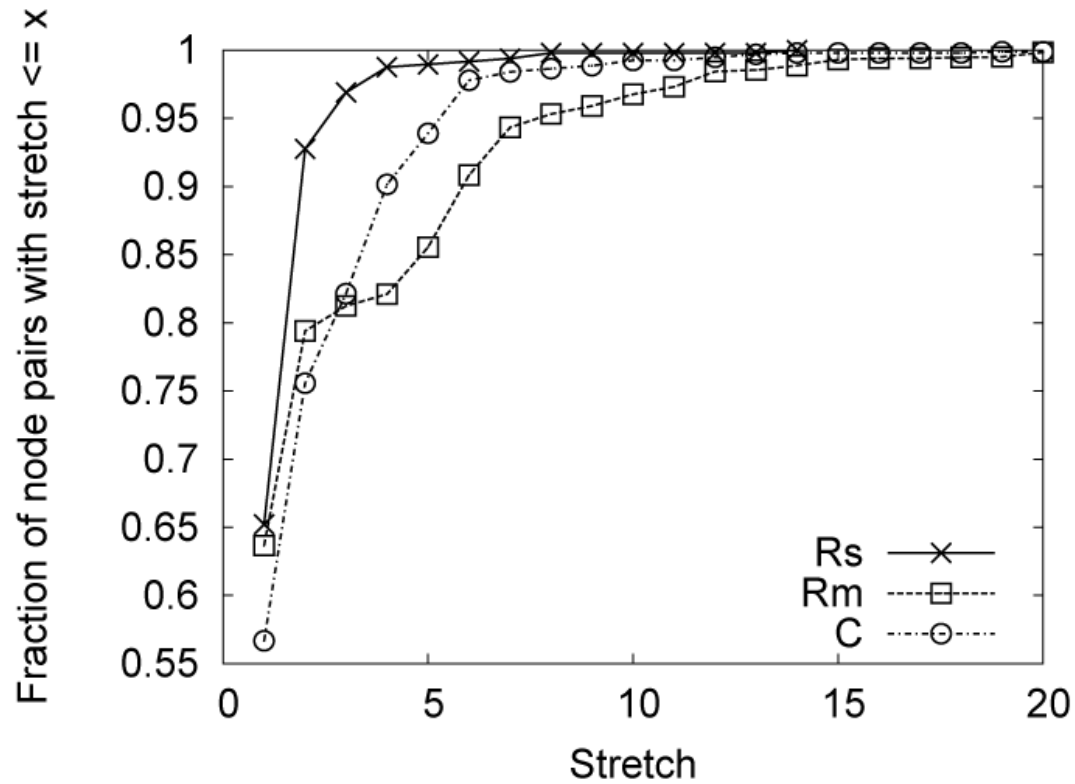
CLDP: Path Stretch (200 Nodes; 200 Obstacles)



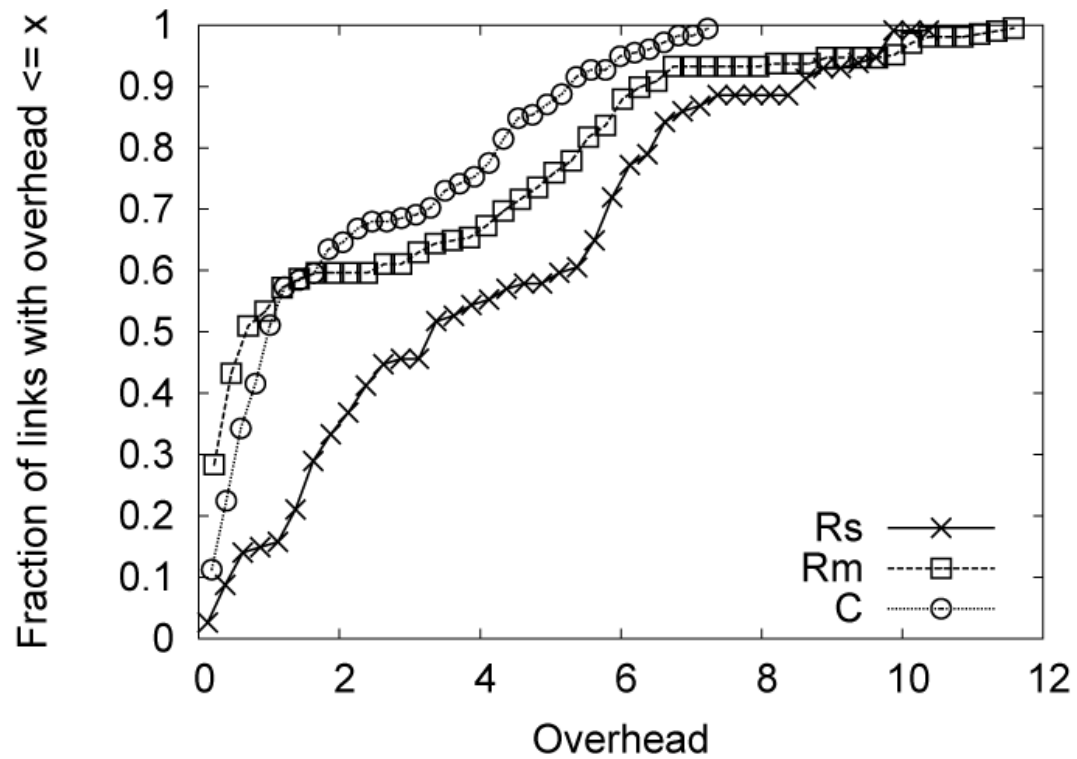
CLDP: Stretch Distribution (200 Nodes; 200 Obstacles)



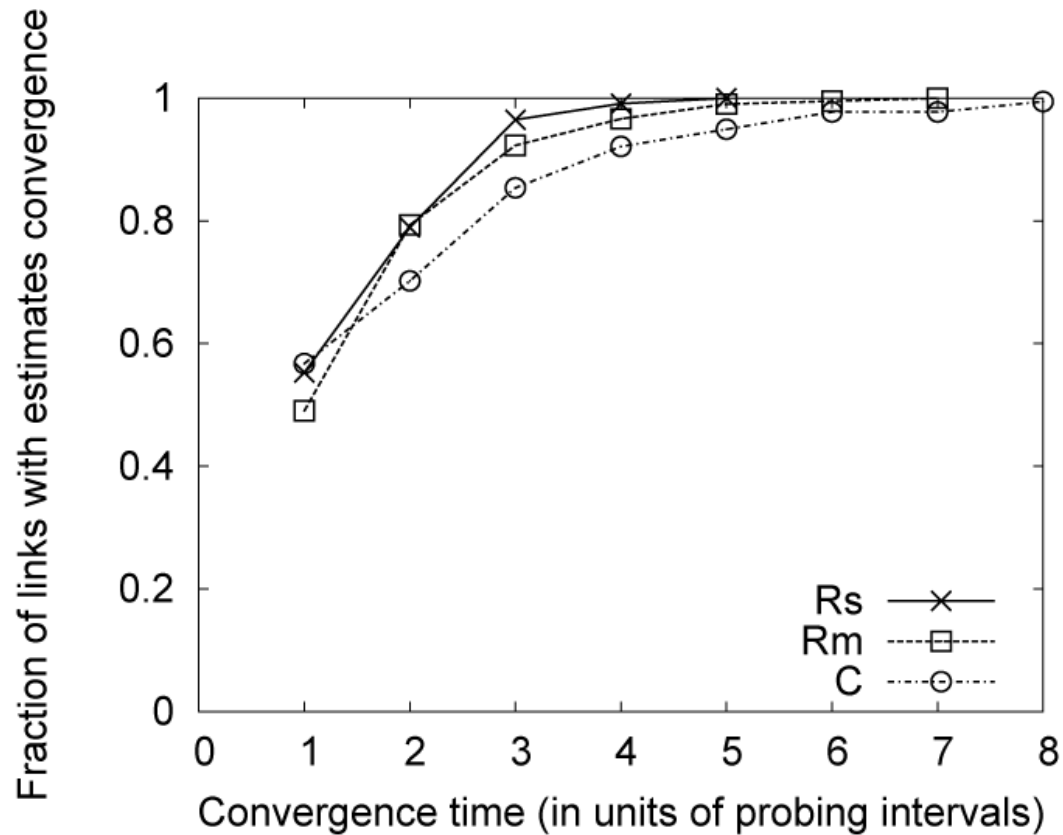
CLDP: Stretch (Mote Testbeds; CDF)



CLDP: Per-Link Overhead (Mote Testbeds; 15 s; CDF)



CLDP: Convergence Time (Mote Testbeds; CDF)



Geographic Routing: A History

- Greedy routing with flooding [Finn, 1987]
- GPSR [MobiCom 2000]
- Planar Graph Pathologies [DIMACS 2001]
- Restricted Delaunay Graph for shorter paths [Gao, Guibas, *et al.*, 2001]
- GHT [HotNets 2002, WSNA 2002, MONET 2003]
- GEM: Query-by-name on trees [Newsome and Song, 2003]
- DIM: Range queries on GPSR routing [Li *et al.*, 2003]
- NoGeo: GPSR's greedy routing on synthetic coordinates [Rao *et al.*, 2003]
- CLDP [NSDI 2005]
- GLIDER: hierarchy/greedy hybrid [Fang, Gao, *et al.*, 2005]

An Aside: Geographic Routing for the Wired Internet?

- GPSR+CLDP route correctly on all connected graphs
- What might they offer the Internet?
 - tiny forwarding tables at core routers
 - but greedy forwarding seems at odds with
 - Intra-domain traffic engineering
 - Inter-domain policy routing
- First sketch of the possibilities [HotNets 2004]

Conclusion

Resource constraints, failures, and scale of deployment make design of sensor network systems hard

Tidy, unrealistic assumptions about radio behavior (e.g., circular radio range) cause algorithm failures in practice

Any-to-any routing, with GPSR and CLDP

- $O(\text{density})$ state per node, correct on all networks

Geographic routing demonstrates difference between paper designs and building real systems!