TUHH
Technische Universität Hamburg-Harburg

TELEMATIK

BGR: Blind Geographic Routing for Sensor Networks

Matthias Witt and Volker Turau

Content

- Motivation
- General Framework
- The BGR Algorithm
- Simulation Results
- Conclusion

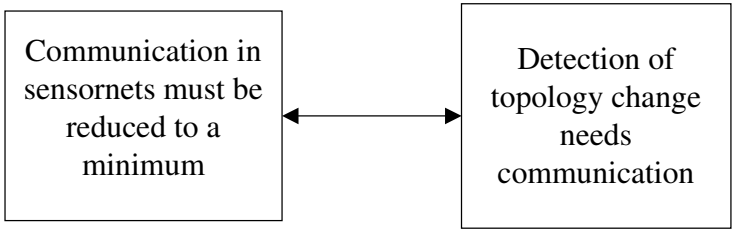
Content

- Motivation
- General Framework
- The BGR Algorithm
- Simulation Results
- Conclusion

Geographic Routing

- Nodes know their geographic position
- Packets are routed to a location
- Greedy mode when applicable
- Example: GPSR (neighbor tables, beacon messages)

Motivation



→ No beacon messages

→ No routing tables

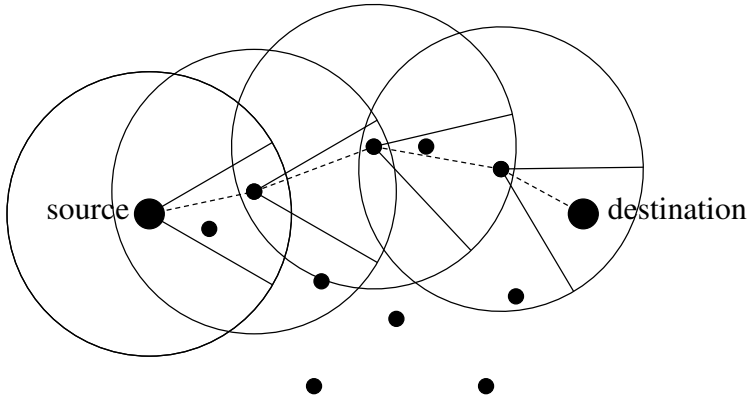
Motivation

- No beacon messages, no routing tables: Nodes have no information at all?
- Nodes know their *geographic position*
- Blind Geographic Routing forwards packets using only location information

Content

- Motivation
- General Framework
- The BGR Algorithm
- Simulation Results
- Conclusion

Example

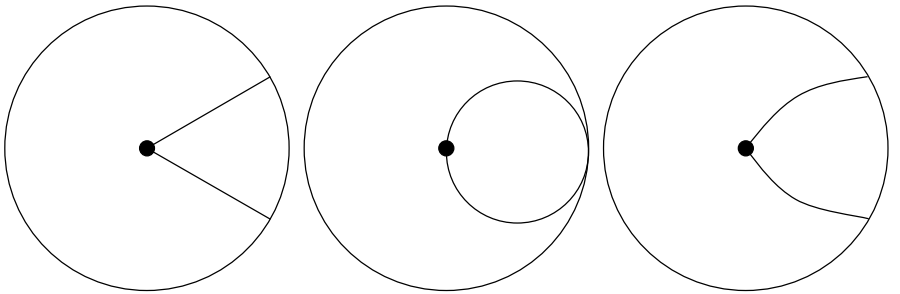


The Forwarding Area

The forwarding area should

- be sufficiently large (and contain at least one node)
- contain many nodes that make big progress toward the target (compared to the forwarder)
- contain few nodes close to the forwarder

Different Forwarding Areas



Timer Function

The timer function should

- select a good forwarder (e. g., based on the progress)
- differentiate the length of times at the different nodes in order to avoid simultaneous or almost simultaneous selection

Content

- Motivation
- General Framework
- The BGR Algorithm
- Simulation Results
- Conclusion

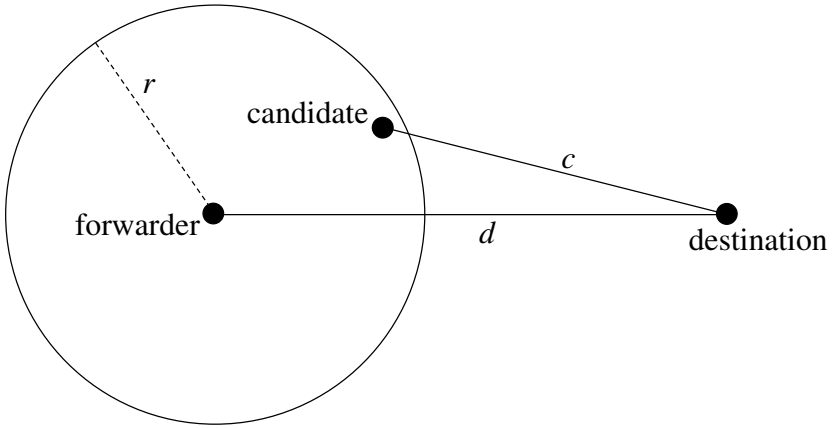
BGR

Main extensions:

- Recovery strategy
- Avoidance of Simultaneous Forwarding

Timer Function in BGR

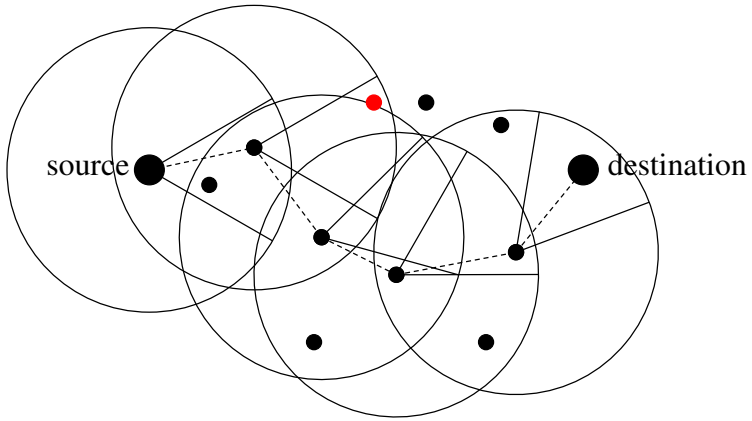
$$t = Max\_Delay \cdot (1 - ((d - c) / r))$$



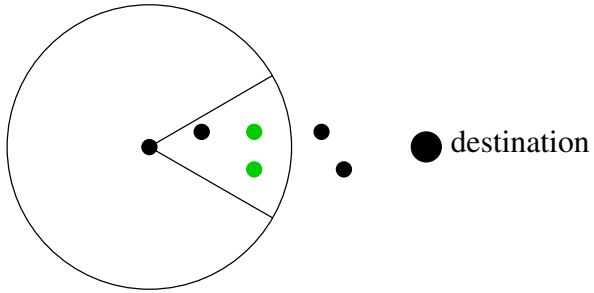
Recovery

- Forwarder sets recovery timer to *Max_Delay*
- On expiration, forwarding area is turned by 60 degrees; next try
- After 3 tries: packet dropped

Example



Avoidance of Simultaneous Forwarding



- Problem: Two hops forward the packet (almost) simultaneously
- All nodes cancel their timers
- Solution: Compare number of hops

Content

- Motivation
- General Framework
- The BGR Algorithm
- Simulation Results
- Conclusion

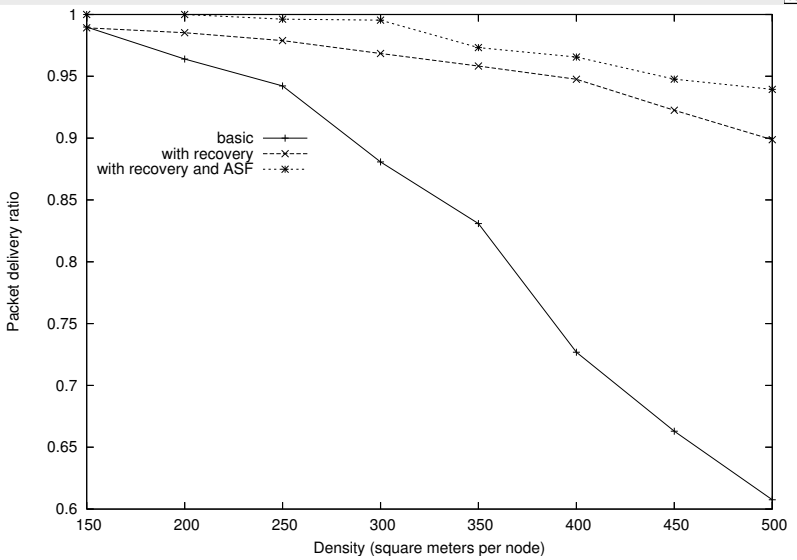
Simulation Scenario

- BGR implemented in *ns-2*
- 100 nodes randomly placed
- Node density was varied
- Every node sends to every other node in intervals of 10 s
- Average values of 20 topologies

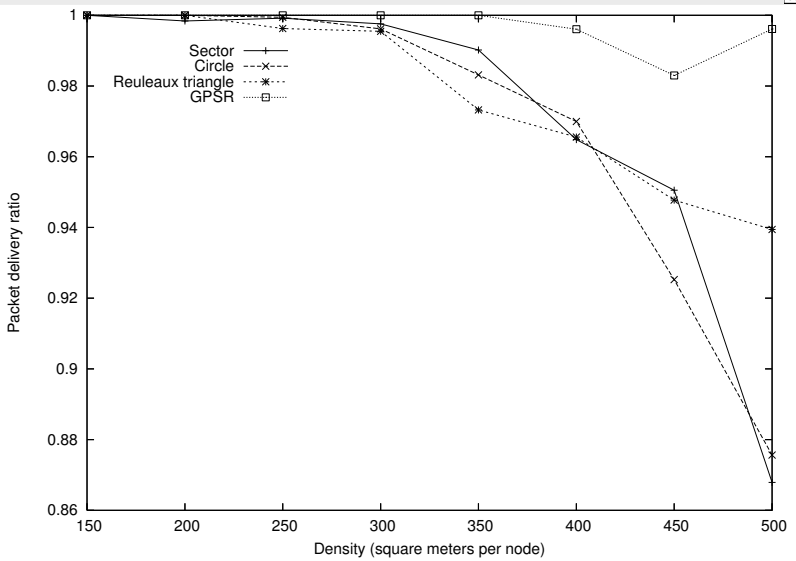
Simulation Parameters

MAC Layer	IEEE 802.11
Number of Nodes	100
Transmission Range	40 m
Simulation Time	10^5 s
Max_Delay	0.5 s

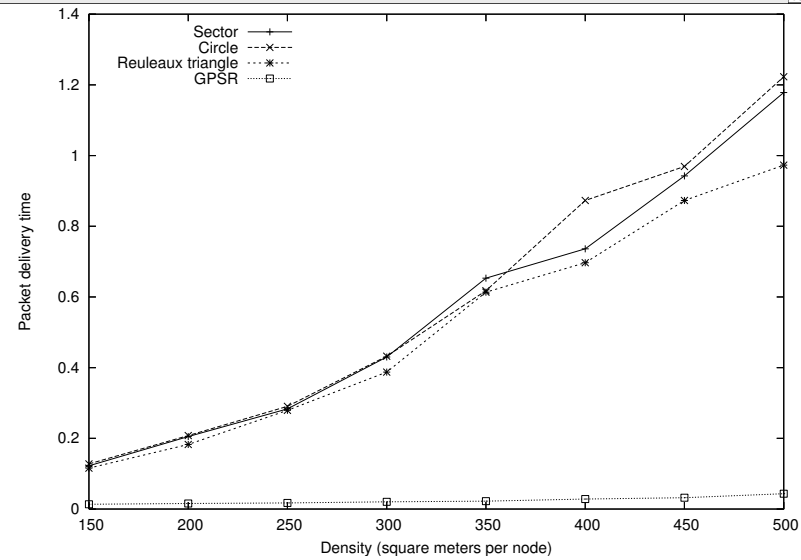
Delivery Ratio for Reuleaux Triangle



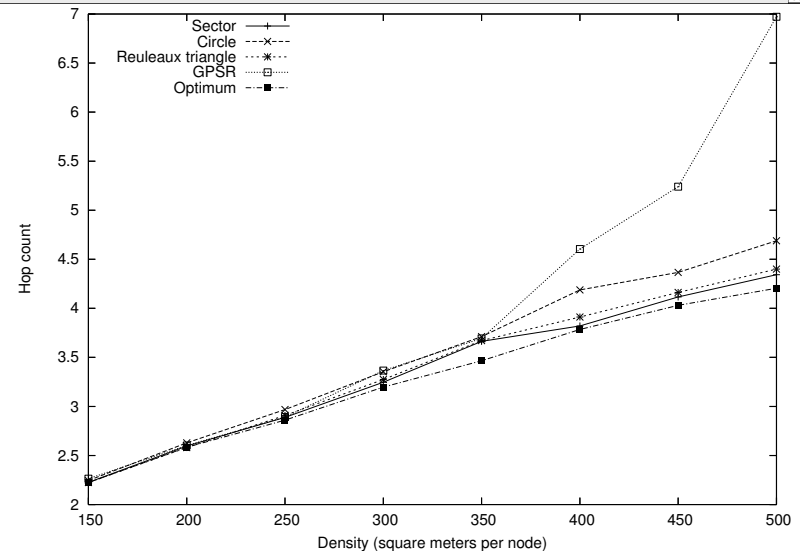
Delivery Ratio



Delivery Time



Hop Count



Content

- Motivation
- General Framework
- The BGR Algorithm
- Simulation Results
- Conclusion

Conclusion

- BGR reliable when node density high
- High adaptivity with regard to topology changes (e. g., node failures or movements)
- Very few packets
- Main contributions: recovery, ASF