# Fault-Tolerant Middleware: Communication

## Reliable communication middelwares for distributed systems

Kyle E. Benson

Department of Computer Science
University of California, Irvine
Irvine, California 92697

kebenson@uci.edu

May 29, 2013

# Introduction

- Traditional routing
  - Unique address: IP, MAC, Peer ID, etc.
  - Source routing: next hop address, neighbor index
  - Local routing: distance-vector, link state, label-switching

# Introduction

- Traditional routing
  - Unique address: IP, MAC, Peer ID, etc.
  - Source routing: next hop address, neighbor index
  - Local routing: distance-vector, link state, label-switching
- Why location information?
  - Geocast: message all (or some) nodes in target region
  - Latency: request from closer server, route locally when possible
  - Congestion: confine route requests to smaller regions (MANETs)
  - Energy: closer nodes need less radio power to reach
  - Sensors: regional event detection, spatial querying
  - Planning: paths (robots), surveillance cameras (focus on area target will appear next)
  - Recovery: avoid problematic areas of the network

# Introduction

- Traditional routing
  - Unique address: IP, MAC, Peer ID, etc.
  - Source routing: next hop address, neighbor index
  - Local routing: distance-vector, link state, label-switching
- Why location information?
  - Geocast: message all (or some) nodes in target region
  - Latency: request from closer server, route locally when possible
  - Congestion: confine route requests to smaller regions (MANETs)
  - Energy: closer nodes need less radio power to reach
  - Sensors: regional event detection, spatial querying
  - Planning: paths (robots), surveillance cameras (focus on area target will appear next)
  - Recovery: avoid problematic areas of the network
    - our primary interest!

# Roadmap

# Roadmap

# Location Service

- Nodes have GPS
- But how to look up destination's location?
- Maintain global information

# Location Service

- Nodes have GPS
- But how to look up destination's location?
- Maintain global information
  easily outdated/inefficient

# Location Service

- Nodes have GPS
- But how to look up destination's location?
- Maintain global information easily outdated/inefficient
- Distribute load
  - In **?**, node updates *location servers* (LS) throughout network
  - Divide network into hierarchical grid
  - LS's in 3 external grids at each level
  - Lookup distance $<$ square LS co-resides in

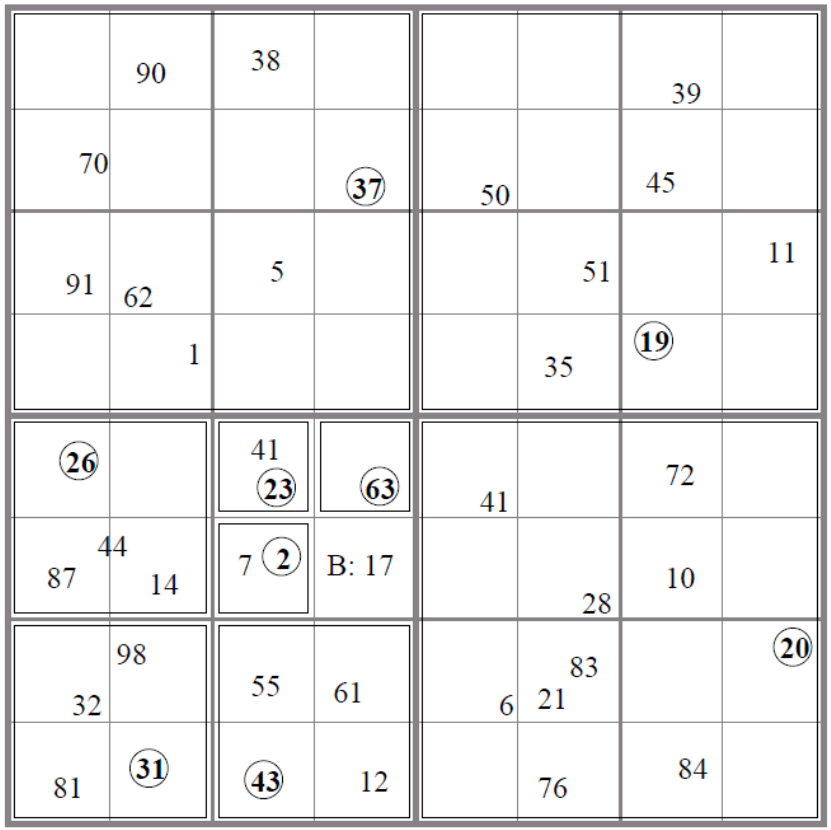


Figure: Hierarchical grid with 4 order-i squares in order-i+1 square.