ZigBee Mesh Networking

Chester Hamilton and Varun Sampath

Mission

- Network the Planes and the Ground Control Station (GCS)
  - Planes have AVR microcontrollers (ArduPilot)
  - GCS is an x86 PC (probably running Linux)
- Considerations
  - Range (Mesh Networking)
  - Performance
  - Power draw
  - Payload weight

Solution?

- XBee-PRO modules
  - 63mW draw
  - 3g weight
  - 250kbps throughput
  - 2 mile outdoor range
  - ZigBee-compliant mesh networking

ArduPilot

- Arduino-based board running LGPL'd autopilot software
- Has built-in telemetry via Serial
- Operates via pre-launch configured waypoints
- We need to dynamically alter waypoints (collision avoidance)

Coding time!

1. Communication between Arduino and a PC
   - Found an XBee API for Java
   - Relatively quick
2. Merging with ArduPilot code
   - Relatively not quick
   - Had to share the Serial bus with the GPS
   - Problem: GPS doesn't like to share
   - Problem 2 *(aside):* GPS doesn't like to work

Solutions

- Two solutions:
  - Multiplexer
  - Software Serial
    - Used the NewSoftSerial Arduino library
    - ~~Shared interrupt handler with RC transceiver~~
    - ~~Optimized the library to run in ArduPilot at 57600 baud~~
      - ~~% N → & (N – 1)~~
      - ~~inline function calls~~
      - ~~Remove unneeded conditionals~~
    - Used a free pin and new interrupt handler

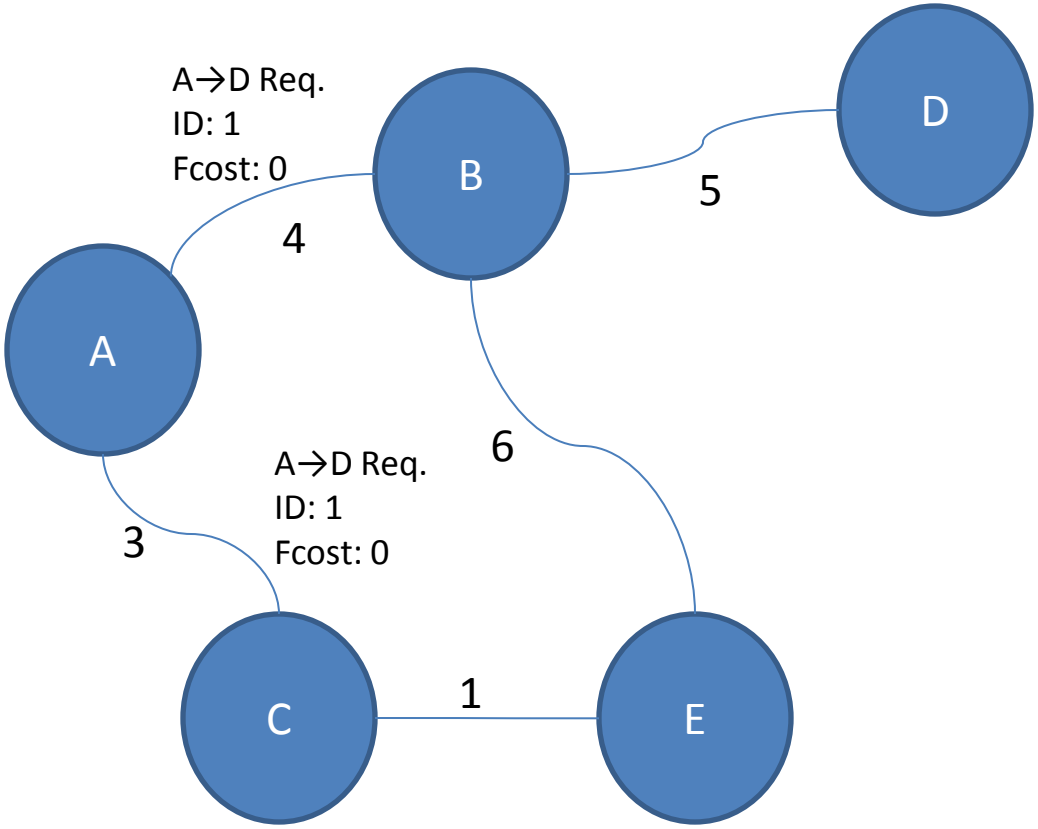
Research

- Performance of a ZigBee mesh network
- Effects of:
  - Moving nodes
  - WLAN interference (AU WiFi)
- Multi-hop performance in general

ZigBee Mesh Routing Algorithm

- Based off of AODV
- Nodes only know the routes they have used in the past
- Only maintain connections with neighbors
- Need to transmit somewhere new?
  - Path Discovery
- Some link disappears in the mesh?
  - Route maintenance

Path Discovery

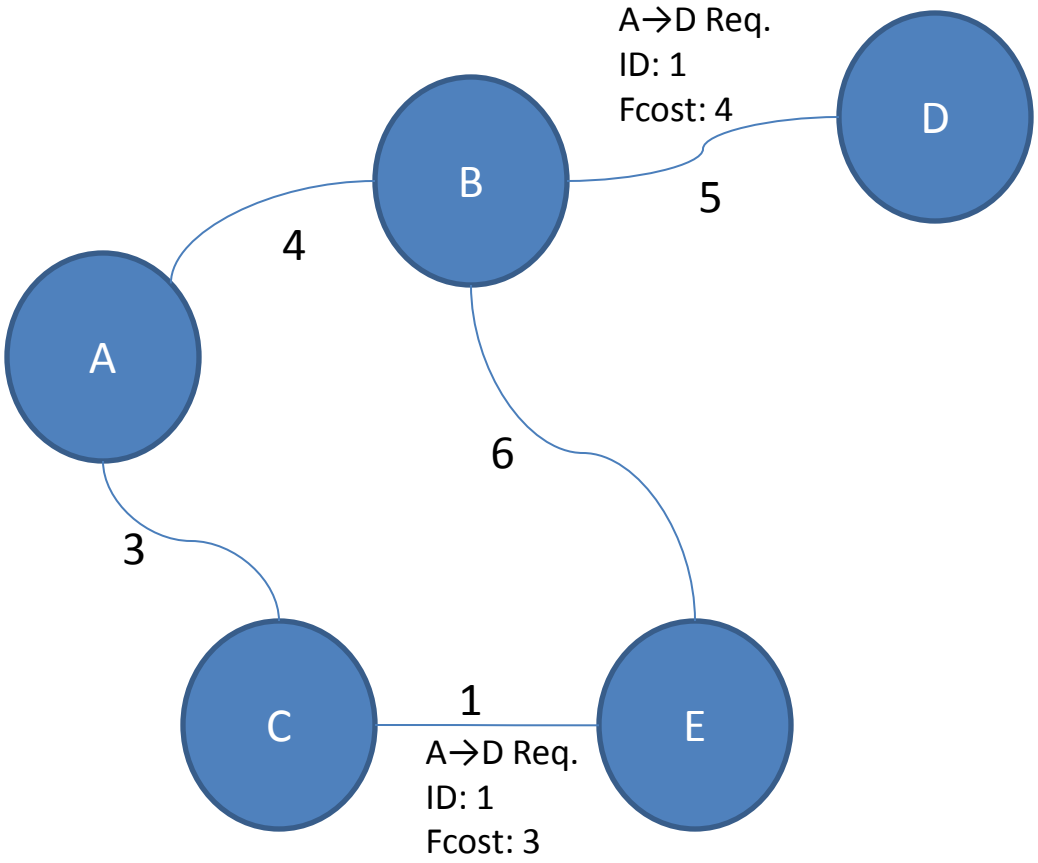


A wants to talk to D, but never met him before

1. A broadcasts a route request to its neighbors.

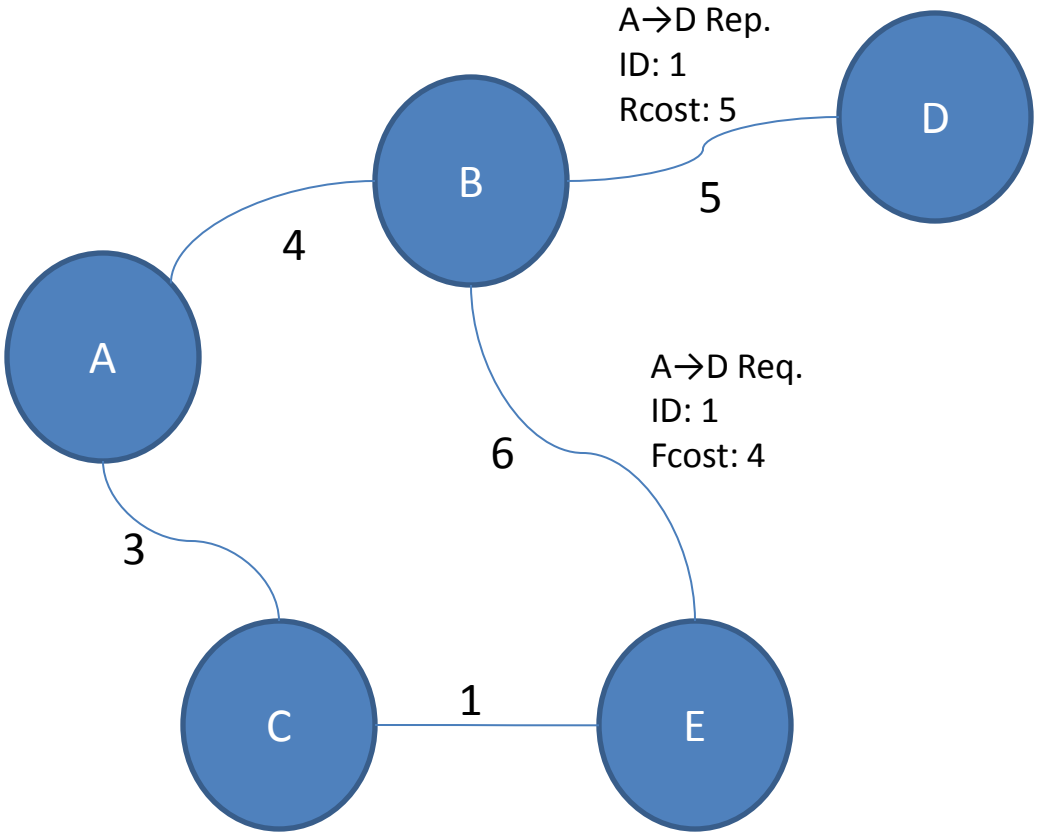
Note: Numbers beside links are link costs

Path Discovery



2. B and C are both not D, so they rebroadcast the route request. Each node creates routing table entries for both forward and reverse routes (not shown).

Path Discovery

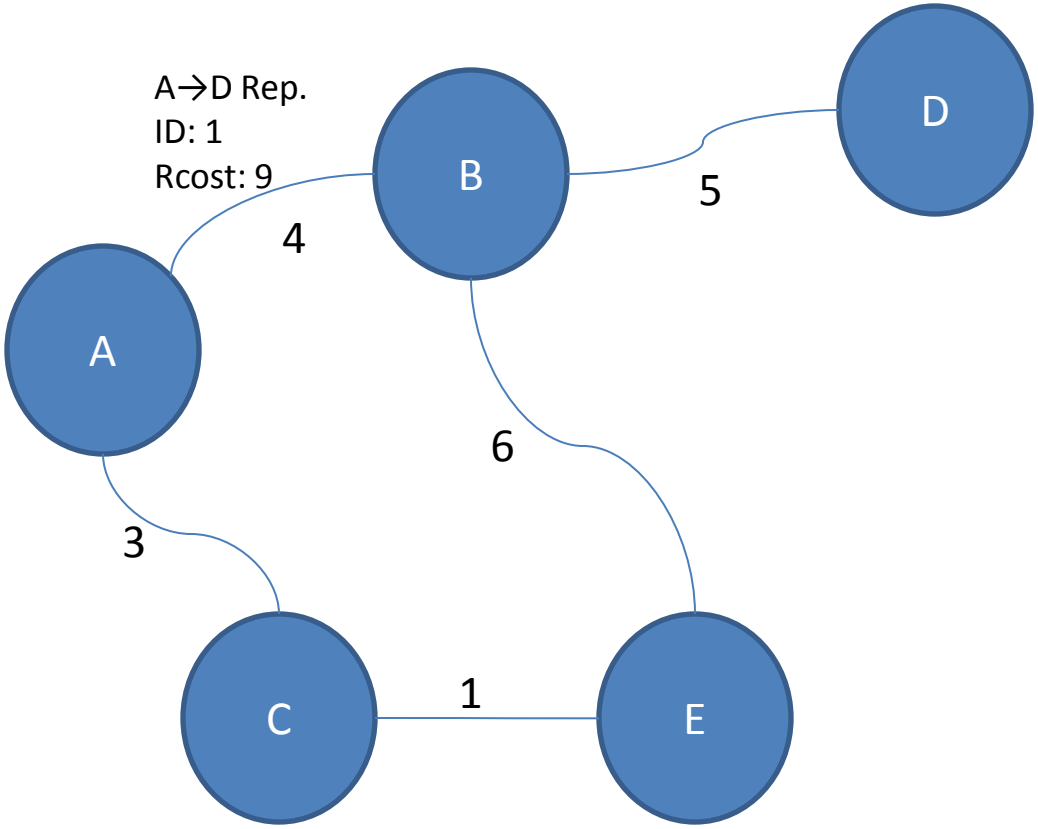


D→A: Cost 9 Next hop: B

3. D receives the route request and unicasts a route reply, setting the outgoing link cost as a residual cost. D's new routing table entry is shown. E continues to broadcast the request.

Path Discovery

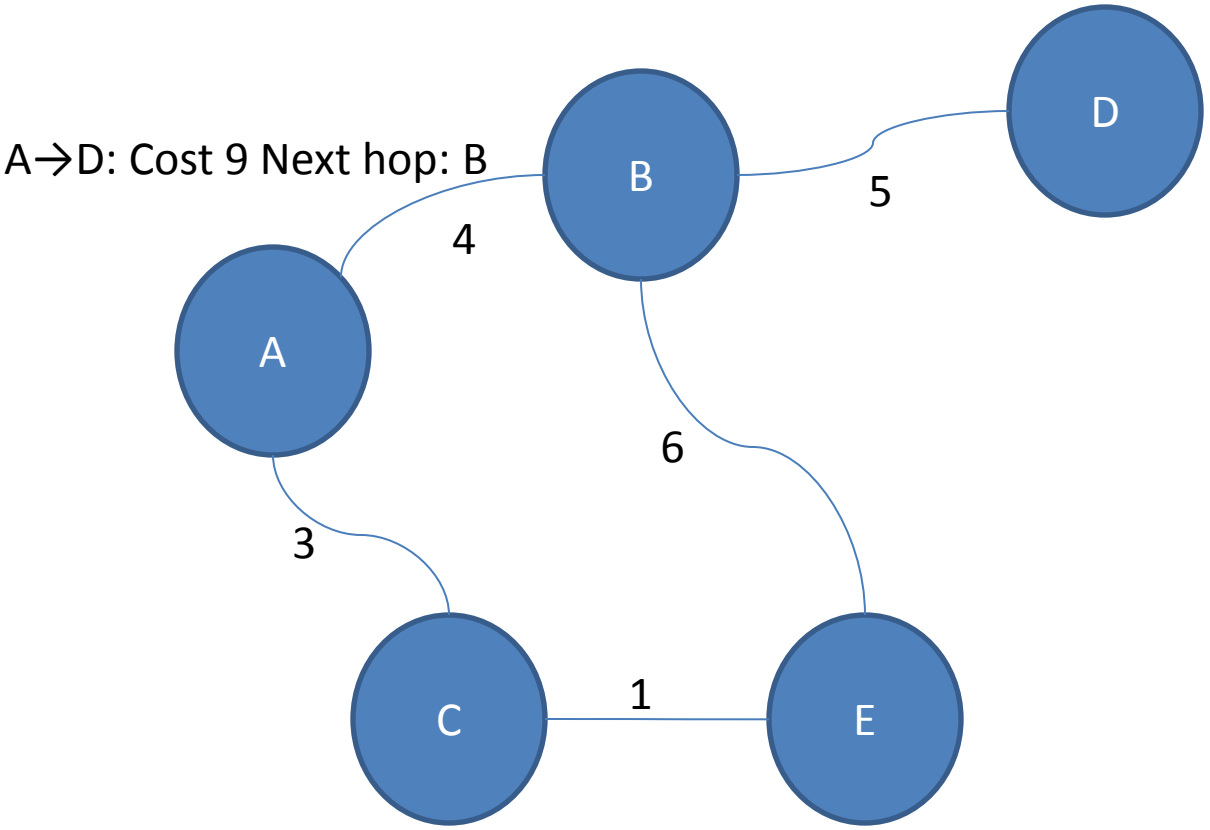
D→A: Cost 9 Next hop: B



4. E's route request is dropped by B since B has seen a lower cost. B unicasts the route reply to A, updating the residual cost.

Path Discovery

D→A: Cost 9 Next hop: B

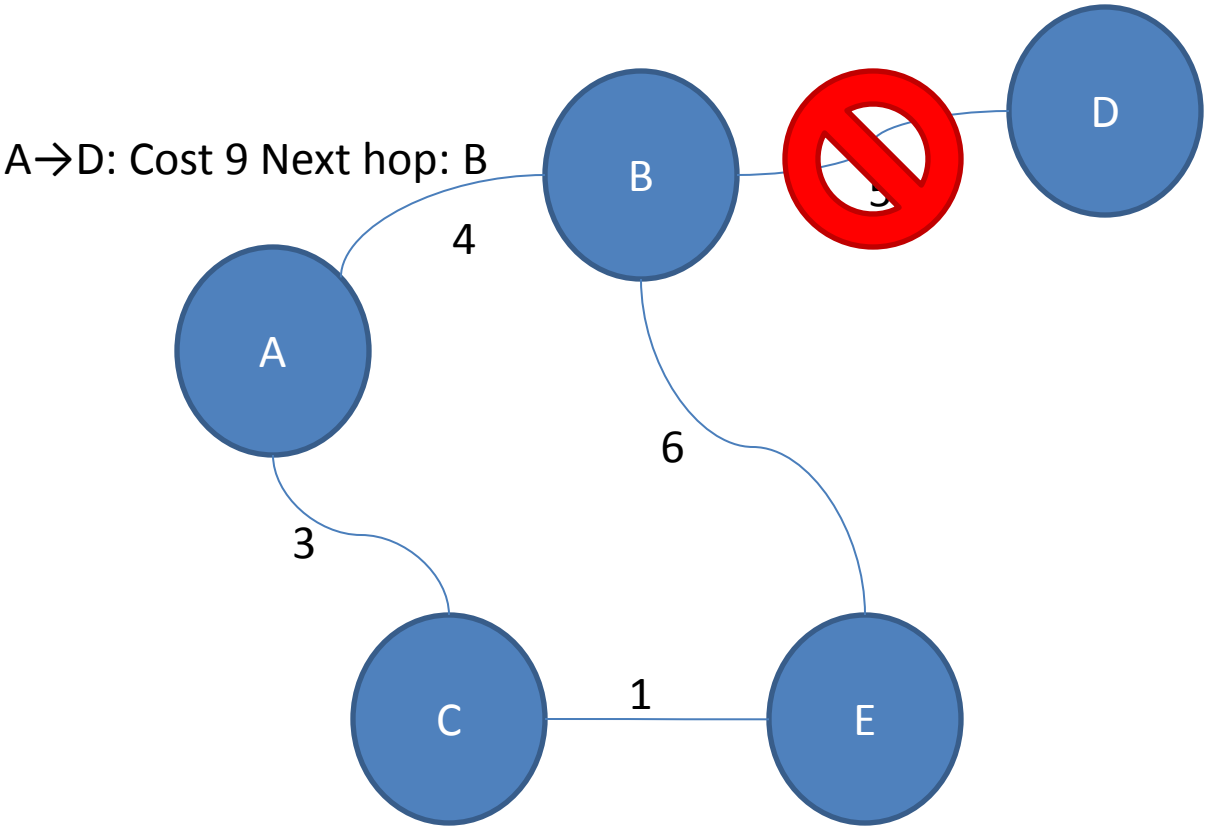


5. A receives the route reply, and updates its routing table. Path discovery is complete.

Route Maintenance

D→A: Cost 9 Next hop: B

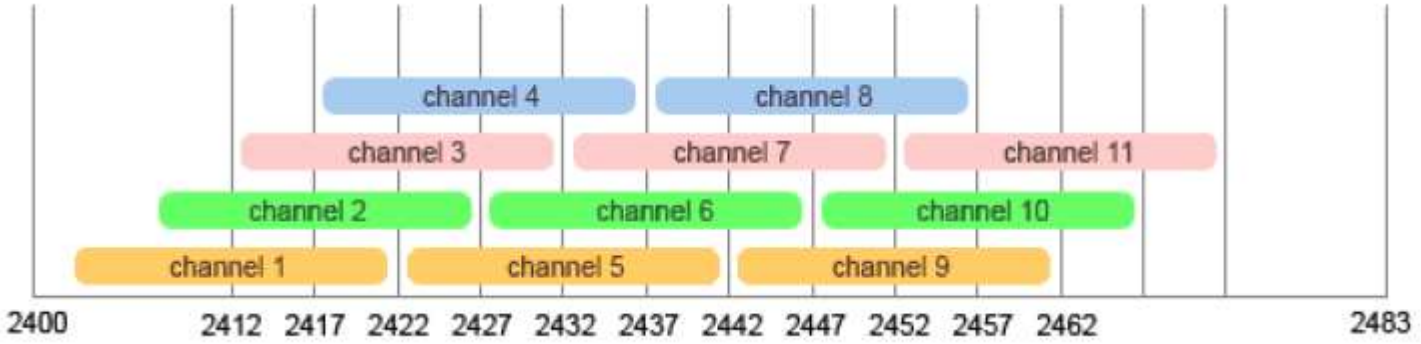
A→D: Cost 9 Next hop: B



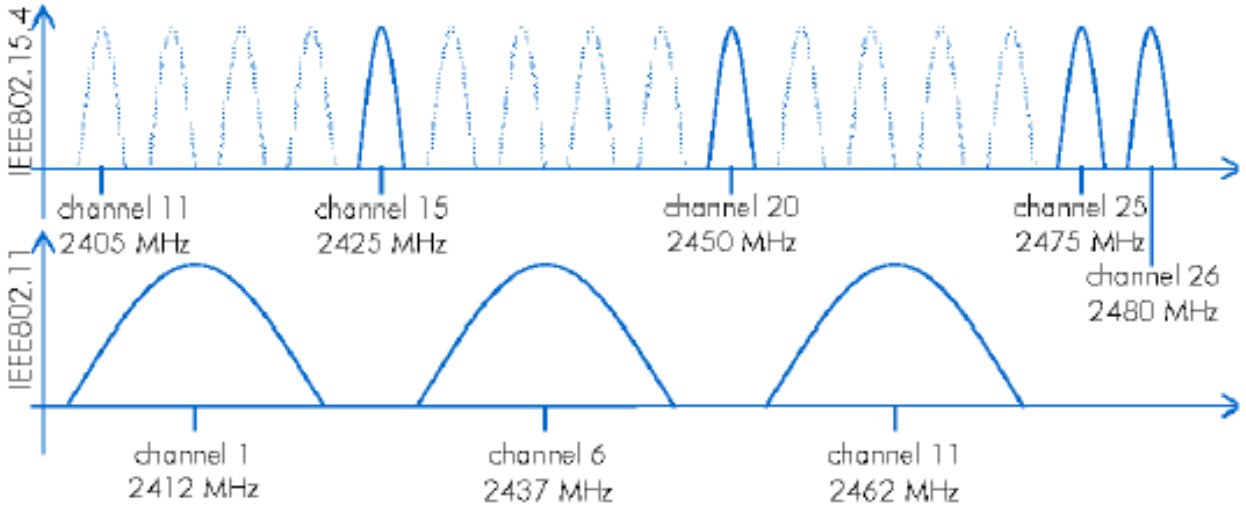
What if the link between B and D breaks, and A tries to send again?

- A sends a packet to D, first sending to B as its routing table suggests
- B notices the link failure, and tells A
- A deletes D's entry in the routing table

802.11 & 802.15.4 Overlap

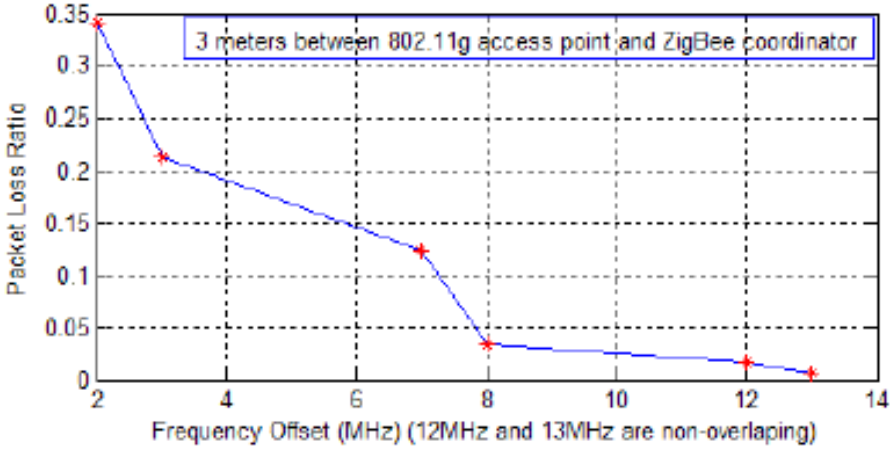


802.11 channel assignments



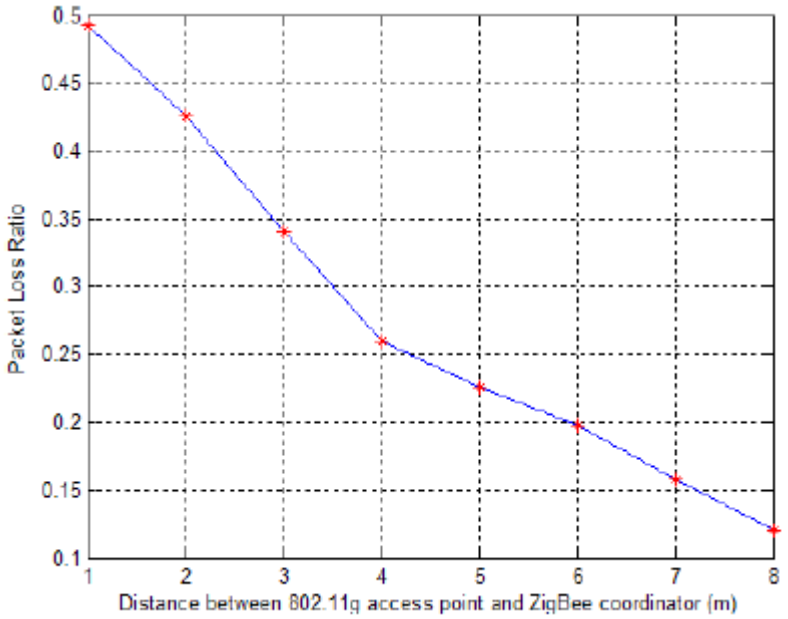
Available 802.15.4 channels in a typical crowded network space

802.11 & 802.15.4 Interference



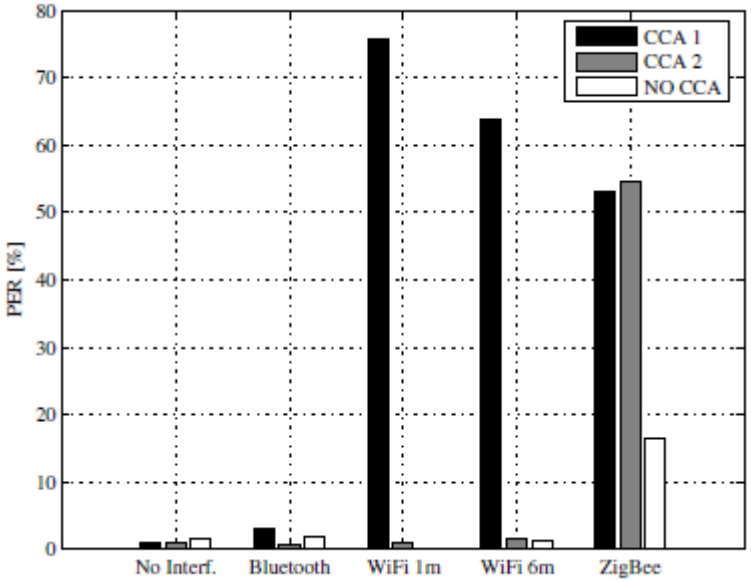
Packet Loss Ratio decreases rapidly as frequency offset is increased

802.11 & 802.15.4 Interference (cont.)



With 2 MHz frequency offset packet loss ratio decreases linearly as distance between AP and coordinator increases

CSMA/CA



ZigBee performance under different CCA modes with different interference sources

Hypotheses

- Moving nodes
  - Lots of link failures imply lots of path discovery operations
  - Should show an increase in latency
- WLAN Interference
  - ZigBee PRO dynamic channel selection should prevent significant overlap
  - Proximity to WLAN AP should decrease throughput, but less so for 802.11n networks