Faster and Better:
The Promise of Dynamic Spectrum Access

Seth Gilbert
National University of Singapore

Dynamic Spectrum Access

What is dynamic spectrum access?

What’s in a name?

- Dynamic spectrum access
- Dynamic spectrum sharing
- Dynamic spectrum management
- Cognitive radio
- Frequency agile radios
- Spectrum agile radios
- Frequency hopping radios
- Multi-channel networks

What is dynamic spectrum access?

Most wireless networks operate on a *single* pre-assigned radio channel.

- Examples in practice:
  - Base stations are assigned fixed channels.
  - Sensor networks are pre-tuned to a particular radio frequency.
- Examples in theory:
  - "Assume all nodes in the system are sharing a single multiple access channel…"

What is dynamic spectrum access?

Key observations:

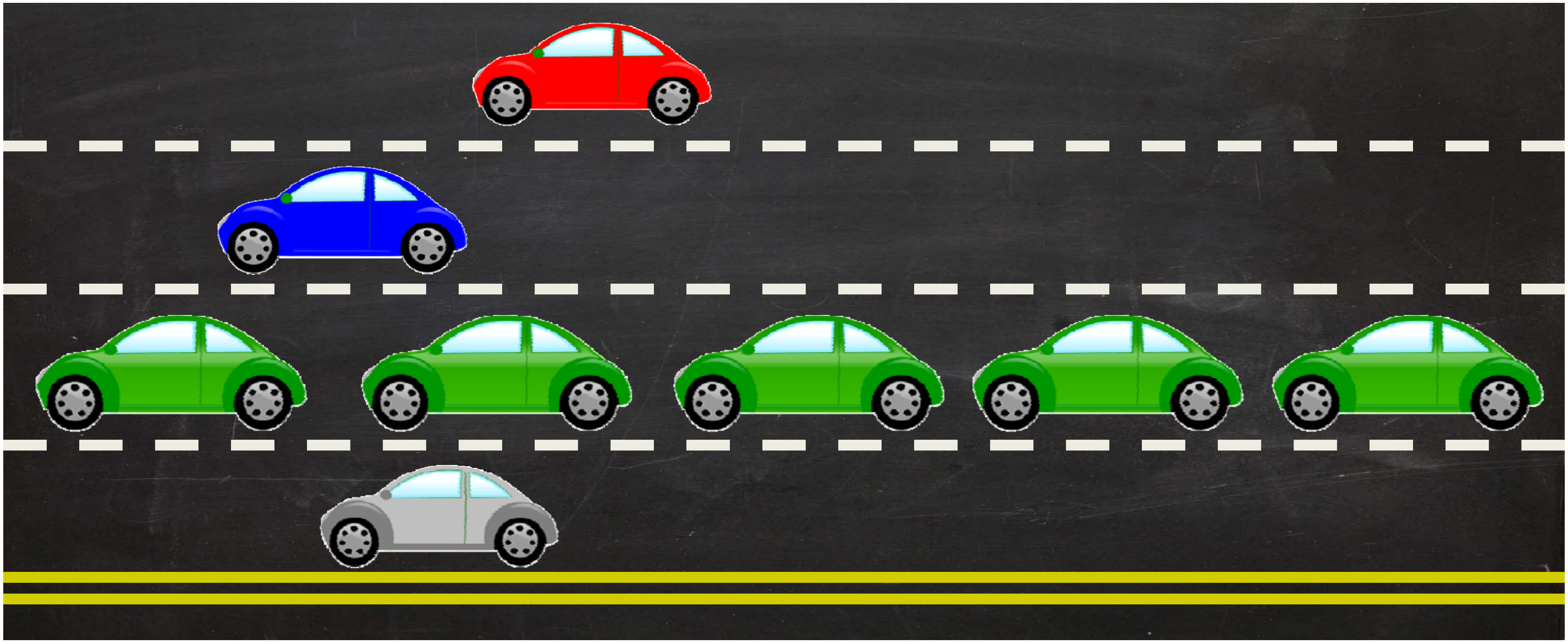
- Most radios can access several different channels:
  - 802.11b: 11 channels (3 orthogonal)
  - 802.11a: 13 (orthogonal) channels
  - Frequency hopping (e.g., Bluetooth)

- Dynamic spectrum usage can be more efficient:
  - Distributed communication over multiple channels
  - More efficient use of a limited resource

What is dynamic spectrum access?

Analogy: Driving on the highway

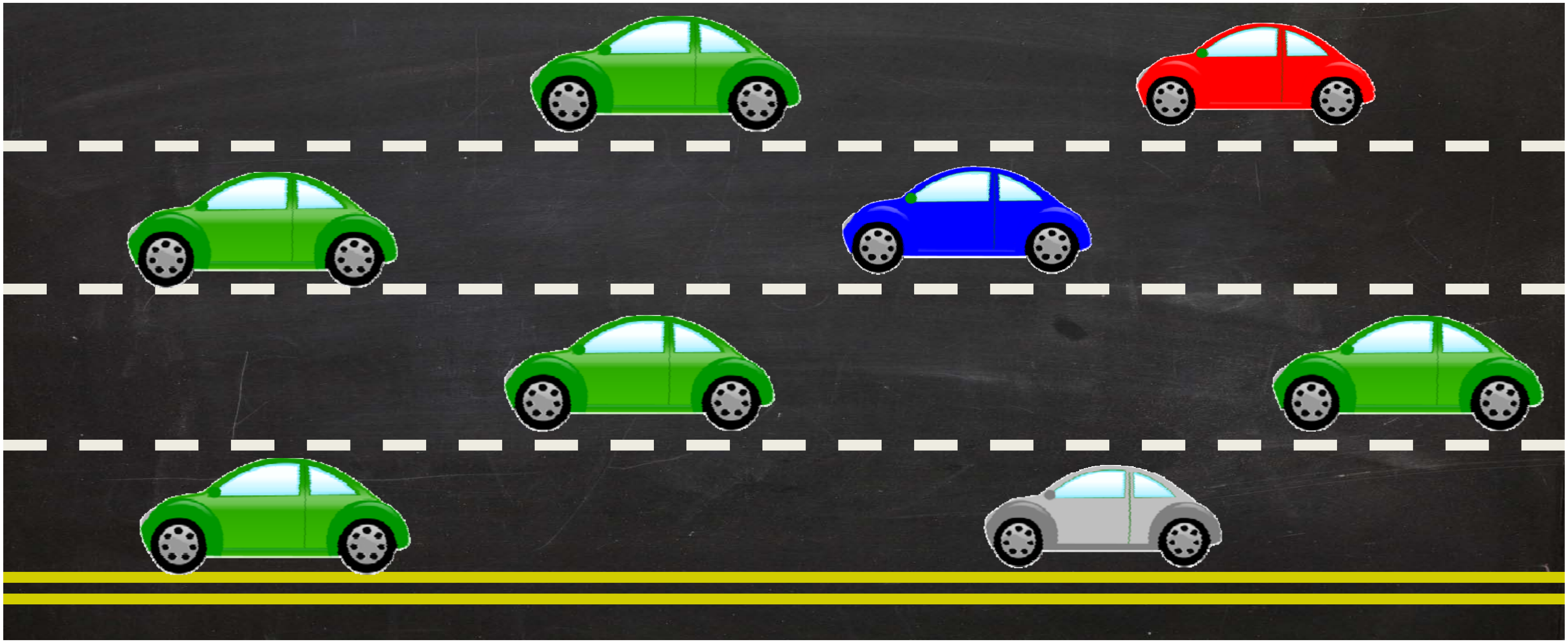
- Every new car is assigned a fixed lane. To change lanes, take your car to a mechanic.



What is dynamic spectrum access?

Analogy: Driving on the highway

- More efficient if driver’s can shift lanes at will.



What is dynamic spectrum access?

Caveats

- Every driver must follow the rules to avoid collisions.
- Some central planning may still help!

What is dynamic spectrum access?

Benefits:

- More efficient use of spectrum

Challenges:

- Minimize cost of changing channels
- Coordination
  - who uses which channels when
  - synchronization
  - overhead for coordination

Dynamic Spectrum Access

Outline

- What is dynamic spectrum access?
- Faster: Can we use DSA to solve problems faster?
  - Standard technologies: 802.11
  - New technologies: Software defined radios
- Better…
  - Can we use DSA to solve problems more reliably, more securely, more efficiently?

Faster...

- 802.11
  - Existing implementation
  - Model
  - Open questions
- SDR / GNU Radio
  - Existing implementation
  - Model
  - Open questions

Dynamic Spectrum Access:

802.11

DSA Implementation: 802.11

Can we use DSA techniques in existing networks?

- Basic 802.11(ab) wireless networks
  - 2.4GHz range: 3 orthogonal channels
  - 5GHz range: 13 orthogonal channels

- Goal:
  - Mobile network of devices supporting pair-wise flows
  - Multi-hop mesh networking applications

DSA Implementation: 802.11

SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks

by Bahl, Chandra, and Dunagan (*MobiCom 2004*)

DSA Implementation: 802.11

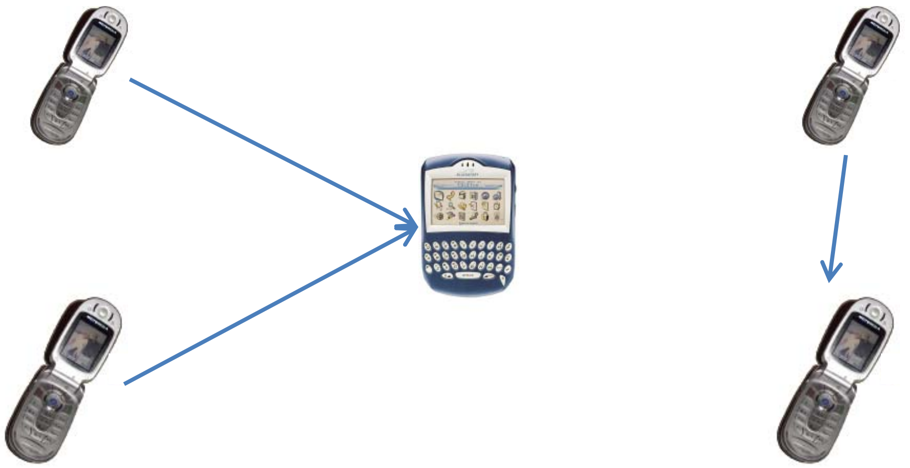
A few interesting details…

- Timing:
  - 10ms communication slots (35 packets/slot @ 54Mbps)
  - Clock synchronization within 5ms

- Overhead achieved:
  - Changing channels: 80µs

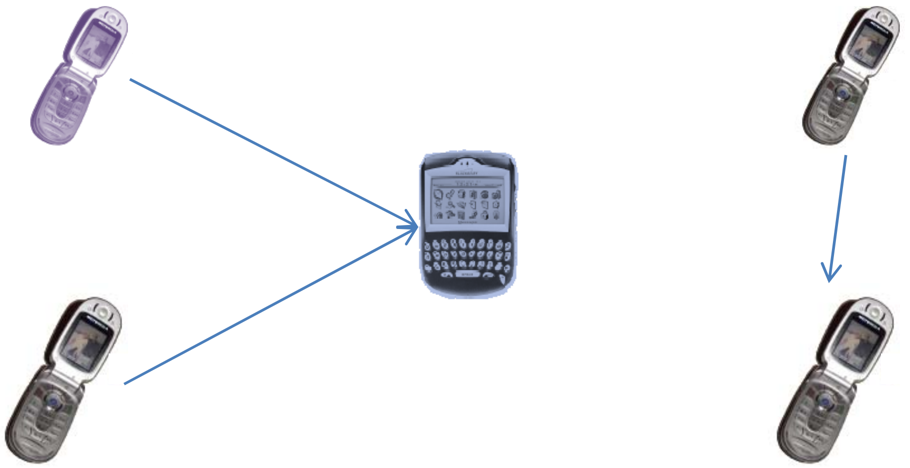
DSA Implementation: 802.11

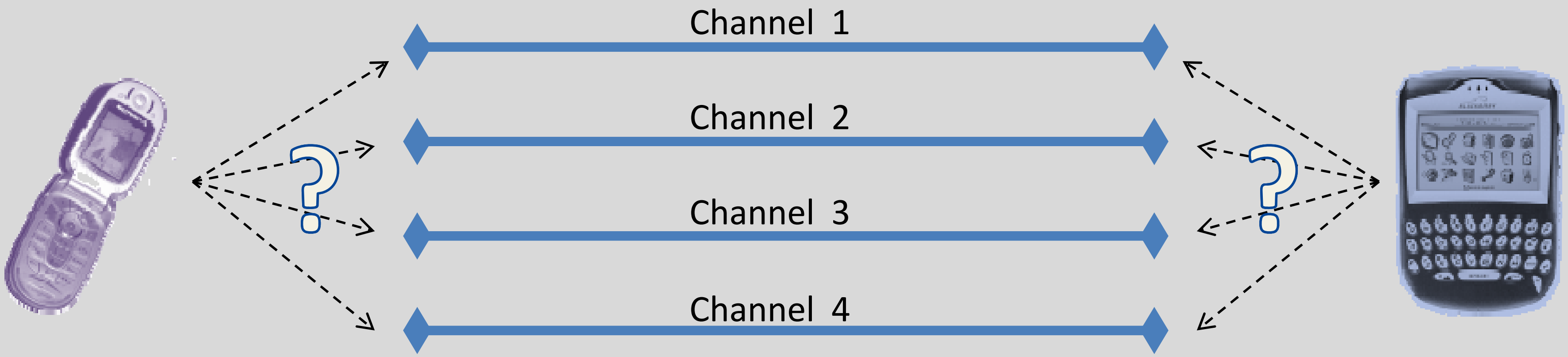
Challenge: How to coordinate?



DSA Implementation: 802.11

Challenge: How to coordinate?





Channel 1

Channel 2

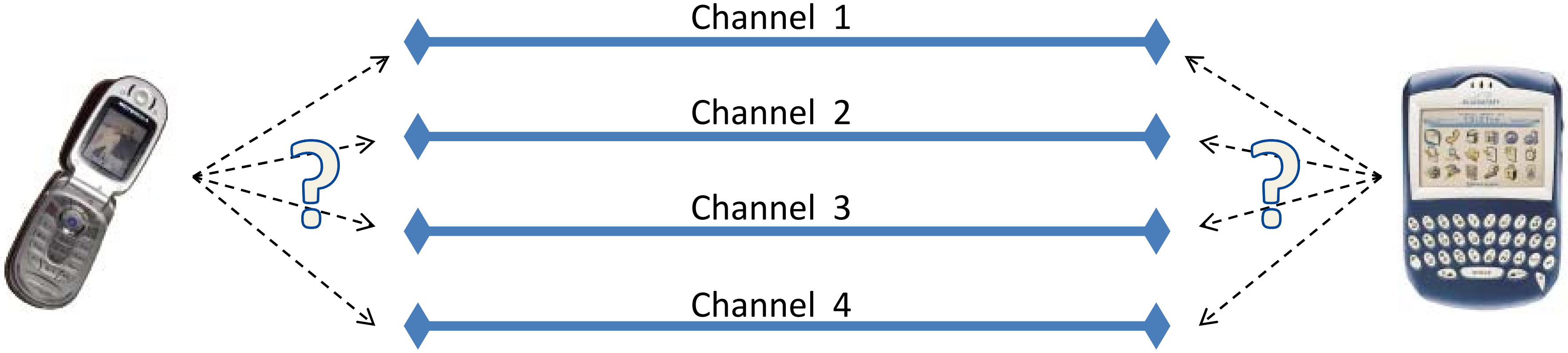
Channel 3

Channel 4

DSA Implementation: 802.11

Challenge: How to coordinate?

- Idea 1: Random frequency hopping
  - Each pair synchronizes for one slot every 130ms.
  - For long flows, very bad performance!



DSA Implementation: 802.11

Challenge: How to coordinate?

- Idea 2: Adapt schedule to maximize synchronization
  - Align schedule of devices supporting a flow.
  - Un-align devices with no pair-wise flows.
  - Devices choose schedules independently.

- Basic idea: 4 interleaved schedules
  - If many packets were received during the last iteration, leave the schedule unchanged.
  - If too many other processes were scheduled for the same channel in the same slot, desynchronize.
  - Otherwise, choose a new schedule that synchronizes with the maximum number of nodes that have queued packets.

DSA Implementation: 802.11

Results

- Analysis
  - Every 530ms, devices overlap their schedules
  - This leads to exchange of schedule information, etc.

- Simulations
  - Big improvement in throughput, compared to single channel 802.11.
  - Not quite linear in # channels, due to randomization.

- Open question
  - Can you get linear improvement in throughput?

DSA Implementation: 802.11

Hardware implementation (Le, Rhee 2010)

Compared three different 802.11 DSA implementations

- AMCP
- MMAC
- SSCH

Conclusions:

- Sub-optimal performance at low load
- Good performance at medium/high load
- Less effective in multi-hop scenarios

DSA Theory: 802.11

Basic Model

- C independent channels
    - Access one channel per round
    - C is small
    - Standard multiple-access channel
        - Collisions, etc.
        - SINR
        - Dual-graph
    - Small overhead for changing channels

- Coarse-grained time synchronization

DSA Theory: 802.11

Cooperative Spectrum Access

- Devices share the spectrum
  - Everyone follows the rules.
  - No malicious users.
  - No interference.

- One application using the spectrum
  - No competing applications.
  - No competing users.

DSA Theory: 802.11

Problem: Partial Information Exchange

Holzer, Pignolet, Smula, Wattenhofer

- Setting:
  - Single-hop network
- Results:
  - Time: $O(k)$ (for some values of k)
  - Number of channels: $O(n^{\varepsilon})$
- Faster:
  - Beats the lower bound of $\Omega(k + \log n)$

DSA Theory: 802.11

Problem: Multi-hop Broadcast / Aggregation

Dolev, Gilbert, Khabbazian, Newport (unpublished)

- Setting:
  - Multi-hop network, diameter D

- Results:
  - Broadcast time: $O((D + \log n)(\log C + \log n / C))$
  - Aggregation?
  - Number of channels: C

- Faster:
  - Beats single-channel results: $O(D \log n)$

DSA Theory: 802.11

Problem: Synchronization

- Setting:
  - Multi-hop network
- Goal:
  - Synchronize all the devices in the network.
- Speed-up:
  - Increased parallelism?
  - Less contention?

DSA Theory: 802.11

Problem: Neighbor Discovery

- Setting:
    - Multi-hop network
- Goal:
    - Find nearby devices.
- Speed-up:
    - Increased parallelism?
    - Less contention?

DSA Theory: 802.11

Problem: Structuring Networks

- Setting:
  - Multi-hop network

- Questions:
  - Leader election
  - Wake-up
  - Independent Set
  - Connected Dominating Set

- Goal:
  - Beat single-channel results: $O(\log^2 n)$

DSA Theory: 802.11

Why can we go faster?

- Throughput:
  - Using C channels, we can send C messages per round.
  - Expected speed-up: C
- Contention resolution:
  - Using C channels, we can select 1 more rapidly.
  - Expected speed-up: $\log(n)$

Dynamic Spectrum Access:
Software Defined Radios

DSA Implementation: Software Defined Radio

Software Defined Radios

- USRP Radios (Universal Software Radio Peripheral)
  - Software reconfigurable radio
  - Supports operation in many bands (e.g., 50MHz-2.9GHz)

- GNU Radio:
  - Implement most radio functionality in software
  - Enable easy experiments with DSA

DSA Implementation: 802.11

Papyrus: A Software Platform for Distributed Dynamic Spectrum Sharing Using SDRs

by Yang, Zhang, Hou, Zhao, Zheng
(*Computer Communication Review 2011*)

Supporting Demanding Wireless Applications with Frequency-agile Radios

by Yang, Hou, Cao, Zhao, Zheng (*NSDI 2010*)

DSA Implementation: Software Defined Radio

Papyrus Platform

- Available spectrum:
  - Devices use 1MHz frequency band.
  - Divided into (up to) 512 sub-carriers.
  - OFDMA modulation

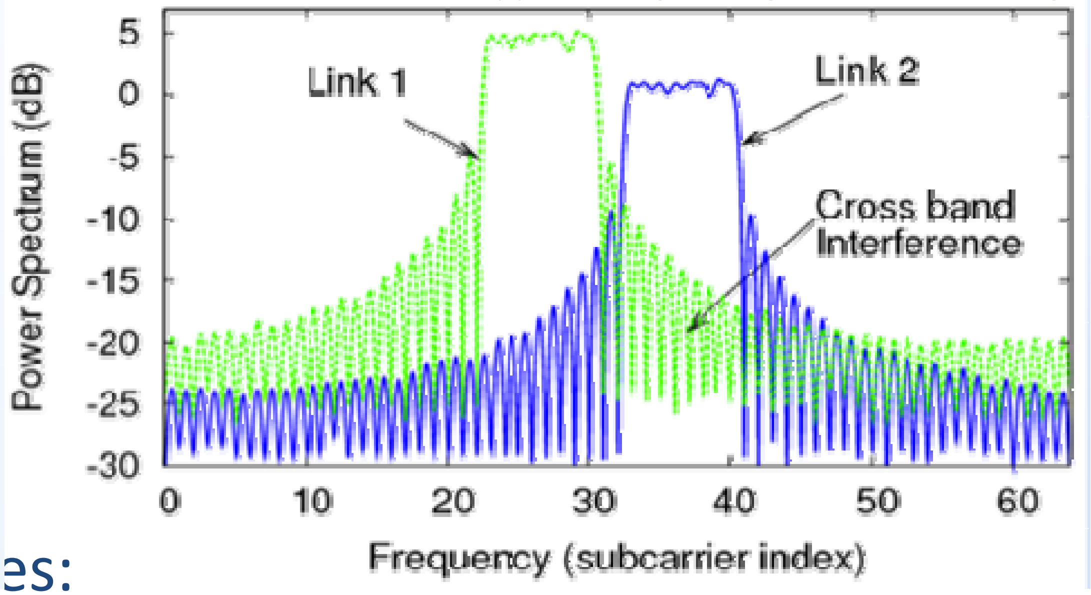
- Papyrus API:
  - SetFreq: set central carrier frequency
  - SetSpectrumUsage: choose any set of sub-carriers
  - SendPacket / ReceivePacket

DSA Implementation: Software Defined Radio

Sensing Vacant / Occupied Frequencies

- Power spectrum density map:

Papyrus, Yang, Zhang, Hou, Zhao, Zheng



- Techniques:
  - Threshold energy
  - Feature detection
  - Edge detection (used in Papyrus)

DSA Implementation: Software Defined Radio

Jello MAC Layer

- Designed for media/streaming applications
  - Pairwise communication
  - Maintain sessions

- Key challenges
  - Coordinate frequency selection
  - Efficiently allocate spectrum
  - Minimize disruption

DSA Implementation: Software Defined Radio

Key Aspect: Bandwidth Allocation

- Requests for bandwidth
  - Arrive on-line
  - Costly to re-allocate---non-constant costs.
  - Can be sub-divided---at a loss (guard bands)

- Solution
  - Classic heuristic: best fit
  - Sub-divide requests when necessary
  - Defragment (concurrently)

DSA Implementation: Software Defined Radio

Results

- Experimental deployment
  - Supports high quality media streaming
  - Low disruption rates
  - Better performance than static spectrum allocation

- Issues
  - Overhead (re-SYNC) due to external disruption (2-3%)
  - Spectrum sensing errors (5-10%)
  - USRP radio large processing delay

DSA Theory: Software Defined Radio

Basic Model

- C channels
  - Access subset of channels in every round
    - Variant 1: any subset
    - Variant 2: any subset within a contiguous range
  - C is (relatively) large
  - Standard multiple-access channel
    - Collisions, etc.
    - SINR
    - Dual-graph
  - Some channel interference?

DSA Theory: Software Defined Radio

Semi-Cooperative Spectrum Access

- Devices share the spectrum
  - Everyone follows the rules.
  - No malicious users.
  - No interference.

- Devices sense and avoid interference
  - Different applications can share the spectrum.
  - Scanning reliably for free spectrum is important!

For more on avoiding primary users, see:
White Space Networking with Wi-Fi like Connectivity
by Bahl, Chandra, Moscibroda, Murty, and Welsh

DSA Theory: Software Defined Radio

Problem: Channel Coordination

- Setting:
  - *Input*: requests (i.e., applications or streams) for some subset of devices to communicate.
  - *Output*: set of channels for each request to use.

- Aspects:
  - *Agreement*: sets of processes should all output the same channel subset.
  - *Non-interference*
  - *Efficiency*

DSA Theory: Software Defined Radio

Problem: Bandwidth Allocation / Re-allocation

- Setting:
  - Single-hop network
  - Centralized defragmentation

- Results:
  - Optimal on-line re-allocation where the cost of re-allocation is unknown.

- Open:
  - Distributed, multi-channel re-allocation protocol

DSA Theory: Software Defined Radio

Problem: Leader Election, Synchronization

- Very fast algorithms: O(1)?
  - Distribute devices over channels
  - Choose winner on smallest channel (via scanning)

- Building blocks:
  - Structuring algorithms
  - Information exchange
  - Replicated state machine algorithms

DSA Theory: Software Defined Radio

Problem: Channel Coordination

- Group Renaming:
  - Assign each group a name (i.e., channel)

- Speed-up:
  - Fast contention resolution (via multi-channel)
  - Cheap signaling (via spectrum scanning)
    - E.g., spell the chosen channel in binary: broadcast on a channel if 1, silent on a channel if 0
    - Use error-correcting codes to tolerate overlap
    - Fast check for agreement…

Summary

Dynamic Spectrum Access: Faster

- Two basic variants
    - 802.11 networks
    - SDR networks

- Preliminary systems work experimenting with increasing speed / spectrum efficiency.

- Lots of algorithmic open questions

Better...

- More robust?
  - Tolerate disruption
- More secure?
  - Tolerate malicious users
- More energy efficient?

Robustness

Interference

- Some channels are disrupted by interference
  - Not too many…
  - Enough to cause problems.

- Causes of disruption
  - Bad channel conditions
  - Other applications

- For now: consider 802.11-style DSA
  - One channel per round

Robustness

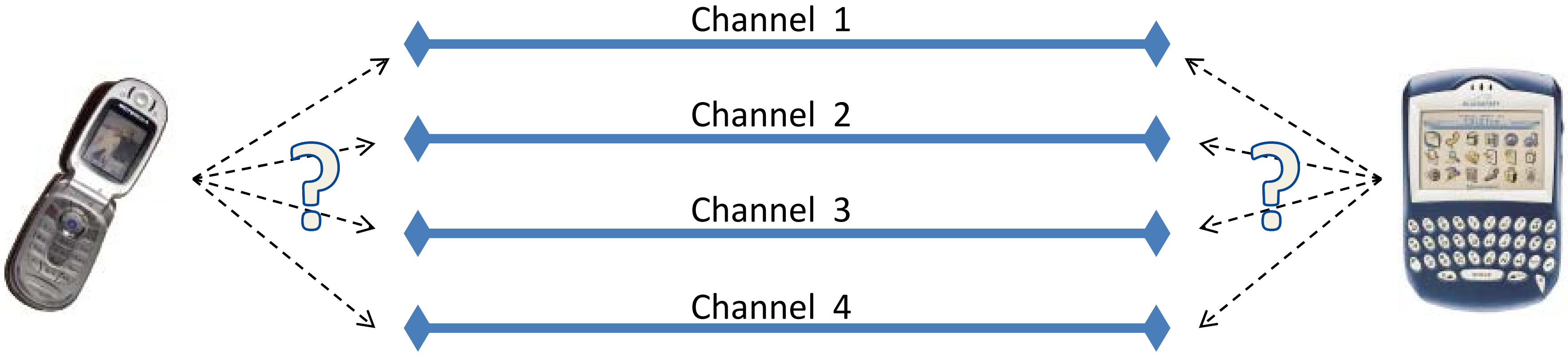
Wireless Channel Models (aside)

- Simple models
  - Deterministic
  - Discrete
- Physical (SINR) model
  - Deterministic
  - Continuous
- Dual-Graph model
  - Nondeterministic
  - Discrete

Robustness

Example: Neighbor Discover

- How to find each other?



Robustness

Speed Dating Despite Jammers

Meier, Pignolet, Schmid, and Wattenhofer

- Jammers disrupt t channels
  - Unknown number of disrupted channels

- Optimal strategy
  - If t is known: choose a random channel in $[1..2t]$
  - Otherwise:
    - Randomly choose a value of t' in $[1, 2, 4, \ldots, C]$
    - Choose a random channel in $[1..2t]$
  - Running time: $O(t \log^2 C)$

Robustness

Example: Synchronization

– Devices arrive in an ad hoc manner
  - Unsynchronized clocks => different round numbering

– Goal: agree on a shared round numbering
  - All participants identify each round in the same way

Robustness

Example: Leader Election / Synchronization

- Devices arrive in an ad hoc manner

- Goal:
  - Choose exactly one device to be the leader.
  - Leader specifies shared round number

Robustness

The Wireless Synchronization Problem

Dolev, Gilbert, Guerraoui, Kuhn, Newport

- Jammers disrupt *t* channels
  - Assume *t* < C/2

g

- Basic idea:
  - In every round, choose a channel at random.
  - Broadcast/listen according to specified distribution.
- Timestamps: how many rounds have you been trying.
  - If you ever receive a message from a process with a bigger timestamp, abort. Otherwise, become leader.

Robustness

The Wireless Synchronization Problem

Dolev, Gilbert, Guerraoui, Kuhn, Newport

- Jammers disrupt t channels
  - Assume $t < C/2$

g

- Basic idea:
  - In every round, choose a channel at random.
  - Broadcast/listen according to specified distribution.

Epoch #	**1**	**2**	**…**	**n-1**	**n**
Length	$\Theta(\log n)$	$\Theta(\log n)$	…	$\Theta(\log n)$	$\Theta(t \log n)$
Prob.	1/n	2/n	…	1/4	1/2

Robustness

The Wireless Synchronization Problem

Dolev, Gilbert, Guerraoui, Kuhn, Newport

- Jammers disrupt *t* channels
    - Assume $t < C/2$

g

- Basic idea:
    - In every round, choose a channel $[1..2t]$ at random.
    - Broadcast/listen according to specified distribution.
    - If you never abort, become the leader.
- Running time: $O(t \log n + \log^2 n)$
    - Within $\log\log(n)$ of optimal.

Robustness

The Wireless Synchronization Problem

Dolev, Gilbert, Guerraoui, Kuhn, Newport

- Jammers disrupt t channels
  - Assume $t < C/2$
  - What if t is unknown (and $t < C/2$)?

g

- More complicated variant:
  - If all the processes arrive at the same time, then running time: $O(t \log^3 n)$
  - Otherwise, running time: $O(C \log^3 n)$

Robustness

Example: Key Establishment

- Two devices
  - No previous interactions
  - No pre-shared secrets

- Problem: jamming and interference

- Goal: agree on a shared secret key
  - Anti-jamming techniques rely on shared keys!

Robustness

Jamming-resistant Key Establishment using Uncoordinated Frequency Hopping

Strasser, Pöpper, Čapkun, Čagalj

- Random frequency hopping
  - Sender and receiver choose random channels.
- Message transmission:
  - Secret fragments sent repeatedly to ensure delivery.
  - Secret may consist of Diffie-Hellman key…
- Secret reassembly:
  - Hash-chain structure ensures that the secret is correctly reassembled.

Robustness

Open Questions

- Broadcast / Multicast
  - Preliminary results (Dolev, Gilbert, Khabbazian, Newport)
  - Multiple messages / throughput / capacity?
  - Network coding?

- Communication complexity
  - Many gossip / information exchange protocols use large messages.
  - Minimum number of bits needed to overcome jamming?

Robustness

SDR Model of Dynamic Spectrum Access

- More powerful robustness techniques
  - Send and receive data in parallel
  - Adapt channel usage to optimize throughput
    - Estimate load / demand
    - Estimate link quality
  - Reduce contention faster

- Caveat:
  - More powerful adversary can listen to many channels at the same time too.

Better…

- More robust?
  - Tolerate disruption
- More secure?
  - Tolerate malicious users
- More energy efficient?

Security

Malicious Users

- Denial of service
  - Disruption
  - Fake requests
- Byzantine users
  - Bad messages
  - Protocol disruption
- Compromised privacy
  - Eavesdropping and snooping

Security: 802.11 networks

Secure Communication Over Radio Channels

Dolev, Gilbert, Guerraoui, Newport

- Authenticated Message Exchange
  - Sign messages: ensure that sender is who you think it is
  - Significantly reduce Byzantine threats

- Shared keys
  - Send Diffie-Hellman messages using AME

- Long-lived communication
  - Construct virtual secure channels among nodes
  - Enable secure communication

Security: 802.11 networks

Authenticated Message Exchange

- Challenge:
  - Cannot identify source of message.
  - Only way to authenticate:
    - "I promise to send a message at 1pm on channel 7."
    - Fixed schedule => authentication
  - Fixed schedules are easy to jam!

Security: 802.11 networks

Authenticated Message Exchange

- Idea: two phases
  - Phase 1: Broadcast data
    - Links are scheduled deterministically based on history.
    - Adversary can jam some subset.
    - Authentication achieved.

  - Phase 2: Feedback
    - Randomized frequency hopping for feedback.
    - Ensures efficient scheduling of future phases.

Security: 802.11 networks

Authenticated Message Exchange

- Results: for $|E|$ simultaneous message, all but t complete:
  - $C > t + 1 : O(\ |E|\ t^2 \log n)$
  - $C > 2t \quad : O(\ |E| \log n)$
- Long-lived communication:
  - Setup: $O(\ n\ t^3 \log n)$
  - Round emulation: $O(\ t \log n)$

Security: 802.11 vs. SDR networks

What about…

- Secrecy via radio limitations?
  - Malicious users can only listen on some (but not all) channels?

- Authentication
  - Malicious users can only broadcast on a subset of channels?

Security: SDR networks

What about…

- Secrecy via radio limitations?
  - Malicious users can only listen on some (but not all) channels?
- Authentication
  - Malicious users can only broadcast on a subset of channels?
- Many open questions…

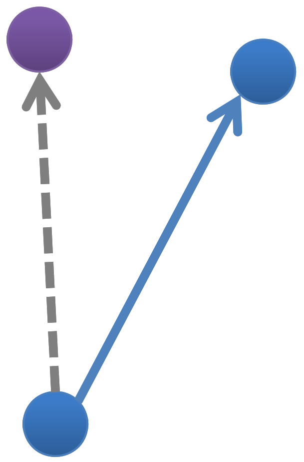
Better...

- More robust?
  - Tolerate disruption
- More secure?
  - Tolerate malicious users
- More energy efficient?

Efficiency

How to save energy?

- Finish faster, send fewer messages
  - Broadcast / receiving costs energy
- Sleep more
  - Reduce active time
- Listen less
  - Overhearing is expensive
  - Avoid messages you don't want



Efficiency

Avoiding Unnecessary Messages

- Sleep more
  - No messages received when asleep.
  - Wastes (useful?) time
  - How to decide when to sleep without knowing which messages are being sent?

Efficiency

Avoiding Unnecessary Messages

- Extreme DSA:
  - Each process has its own dedicated channel.
  - Only listen on your own channel.
  - Never receive an unnecessary message

- Problems:
  - Too many channels.
  - More than one designated receiver?

Efficiency

Avoiding Unnecessary Messages

- Geographic Spectrum Partitioning:
  - Channels assigned based on geographic location.
  - Choose broadcast channel based on location of destination.
- Application:
  - Geo-routing
- Problems:
  - Requires location information (of self and neighbors)

Efficiency

How to save energy?

- Trade-off:
  - Number of channels used
  - Amount of energy
  - Robustness to interference
  - Power (and range)

Summary

Summary

Dynamic Spectrum Access

Two basic implementations:

- 802.11 networks
  - Existing hardware
  - Small number of channels

- Software Defined Radios
  - Experimental hardware
  - Still in development
  - Huge amounts of flexilibity

Summary

Dynamic Spectrum Access

Two basic flavors:

- Cooperative
  - – All users tolerate non-exclusive access to the spectrum

- Non-cooperative
  - – Some (primary) users require exclusive access.
  - – Other (secondary) users must avoid primary users.
  - – Tolerates legacy users.

Summary

Many open problems

Faster, more robust, more secure, more efficient:

- Broadcast, multicast, gossip
- Synchronization
- Overlay structures
  - Creation
  - Maintenance
- Shared memory (e.g., geographic data repository, GHT)
- Aggregation / data collection
- Contention resolution

Summary

Many open problems

Spectrum allocation problems

- Scheduling problems
  - Who should which channels when
  - Minimize overhead (guardbands)
  - Spectrum reallocation

- Energy optimization
  - Minimize number of channels used
  - Minimize overhearing

Challenges for next FOMC

- Models that capture DSA technology
- Techniques for addressing the problems of dynamic spectrum access.
- Algorithms that are faster and better...