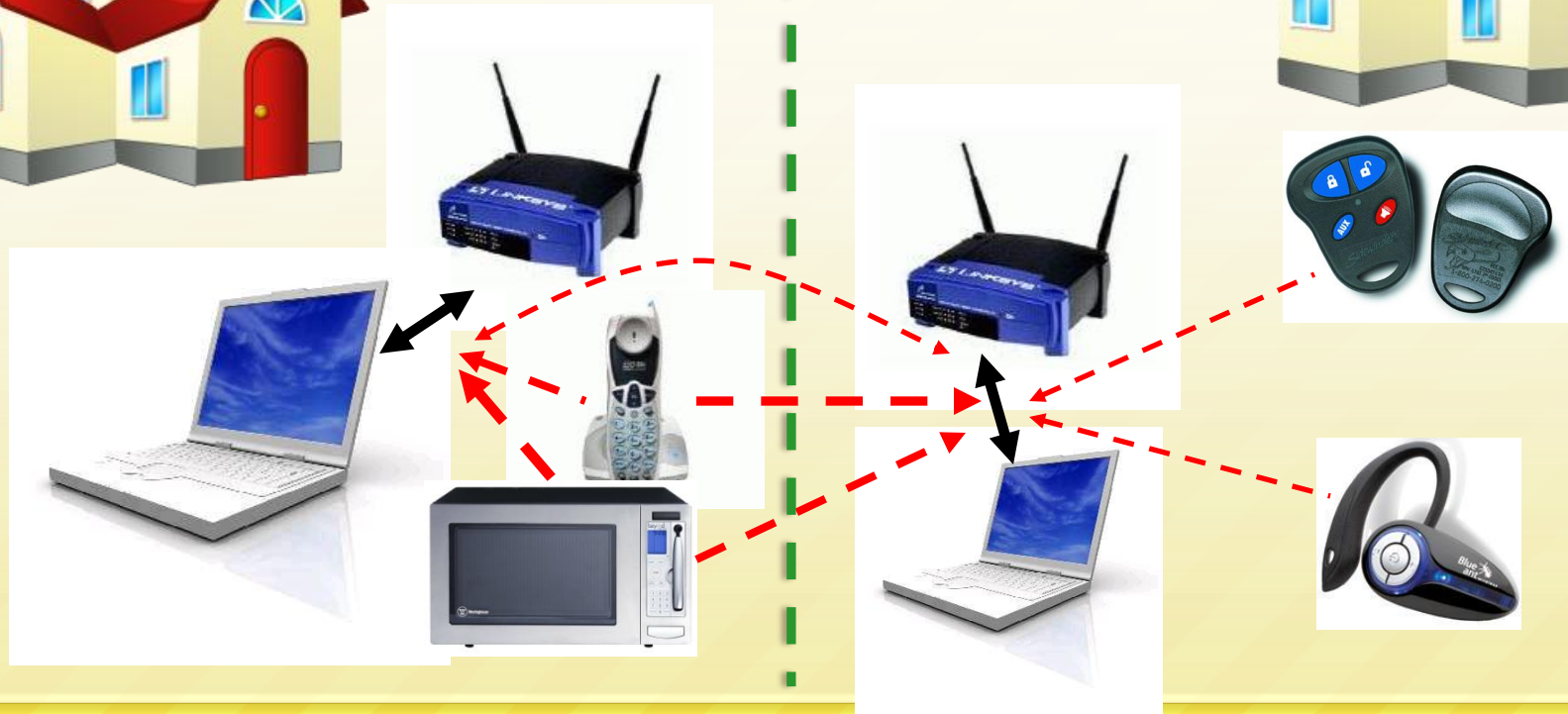


Low-Level Tools for Diagnosing Wireless Problems

Srinivasan Seshan (and many collaborators)
Carnegie Mellon University

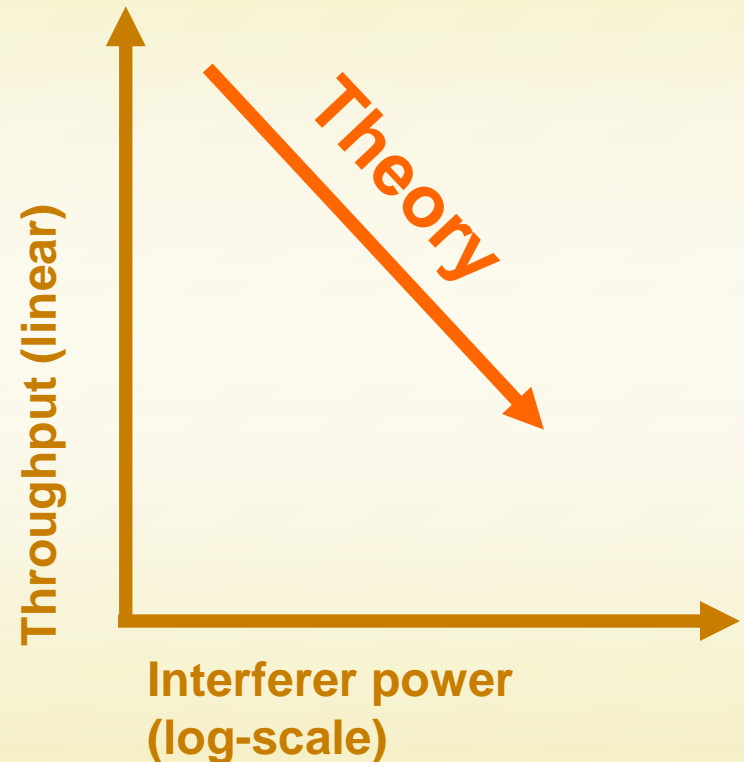
Growing Interference in Unlicensed Bands

- Anecdotal evidence of problems, but how severe and how do we fix this?



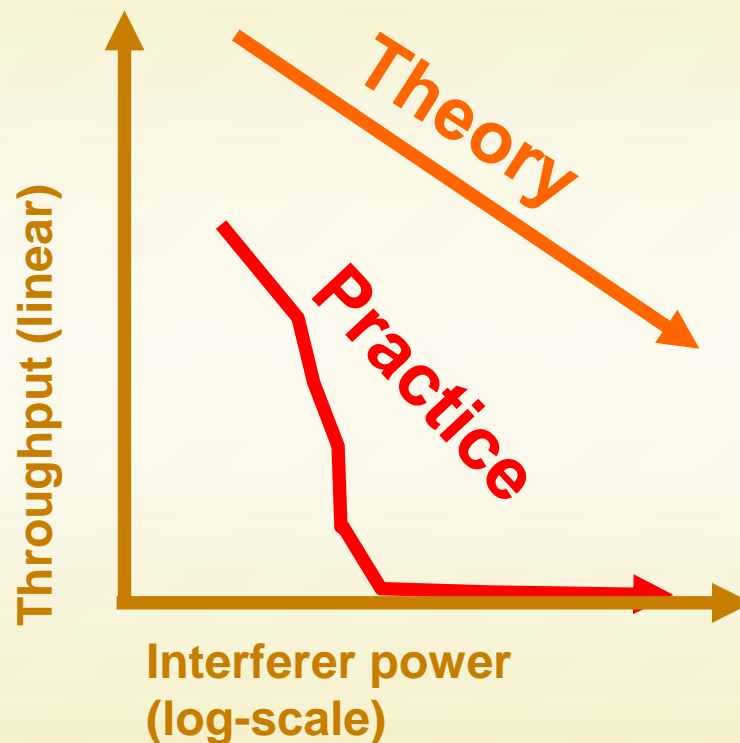
What do we expect?

- Throughput to decrease linearly with interference
- Easy to understand and predict
- Coexistence between technologies/neighbors easy → just design each link to gracefully adapt to interference level



What we see...

- Hard to predict real behavior
- Effects of interference more severe in practice
 - Hardware limitations of commodity cards, which theory doesn't model
 - Protocols often designed to backoff
 - Most polite or most sensitive protocol loses ☹

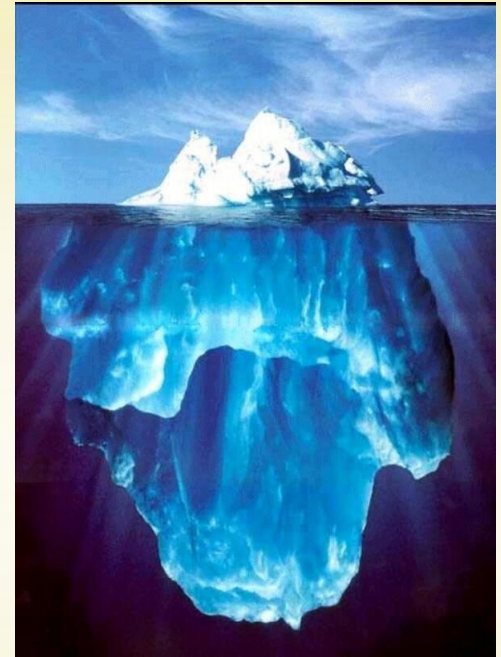


Paths to Solutions

- Pair-wise coexistence
 - Bluetooth/802.11
 - Zigbee/802.11
 - Etc...
- Better coexistence
 - Explicit spectrum management
 - Controlled spatial reuse
 - Transmission power control
 - Directional antennas
 - These techniques also lead to better performance
 - But.... require changes to many parts of protocols
- Better diagnosis

Wireless Diagnosis

- State-of-the-art (DAIR, Jigsaw, Wit)
 - Enterprise settings with dense monitoring
 - 802.11 focus
- Home environments
 - Multiple technologies
 - Multiple administrative domains (i.e. homes)
 - Cheap, inexpensive devices with long lifetimes
 - Often external one-time help like GeekSquad



Diagnosis Tools

Wired Networks

- Ping
 - Traceroute
 - Tcpdump
 - Etc.
-
- Solve problems related to node and reachability failures
 - Led to “connectivity wizards” of today

Wireless Networks

- Alternative: spectrum analyzers
 - Expensive, fail to provide context

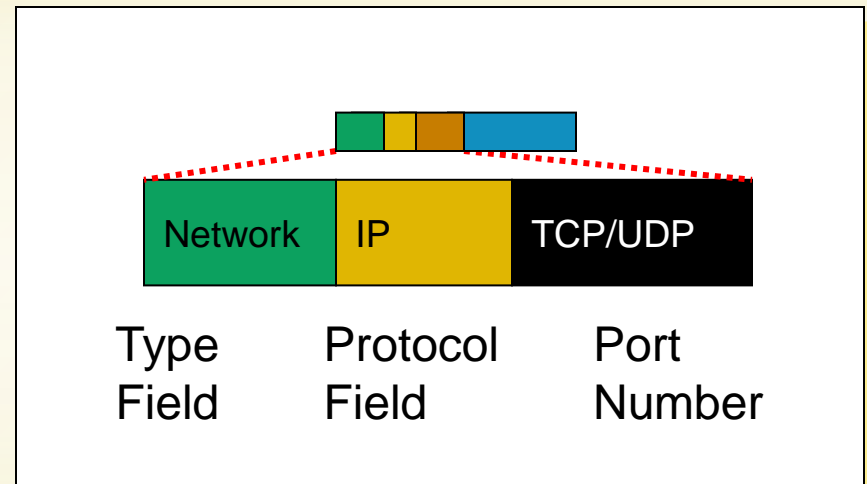
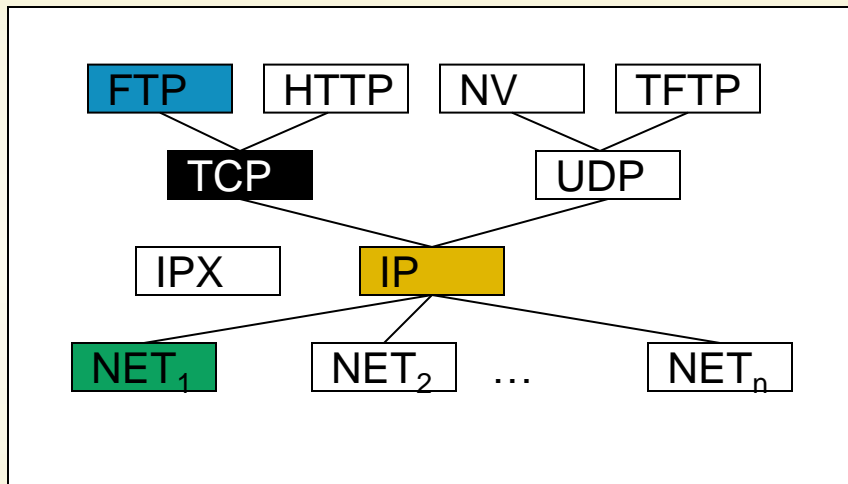
- Existing tools fail to observe source of the problem
 - Problems related to low-level link behavior
 - Interactions between technologies
 - Soft failures

Better Monitoring of Wireless Networks

- Combine tcpdump & spectrum analyzer
- Requirements:
 - Multi-protocol
 - Support at least a small (5-10) number of protocols/RF sources
 - Real-time detection
 - Near real-time throughput requirement
 - Some latency is ok
 - Protocol Extensible
 - Add support for newer protocols

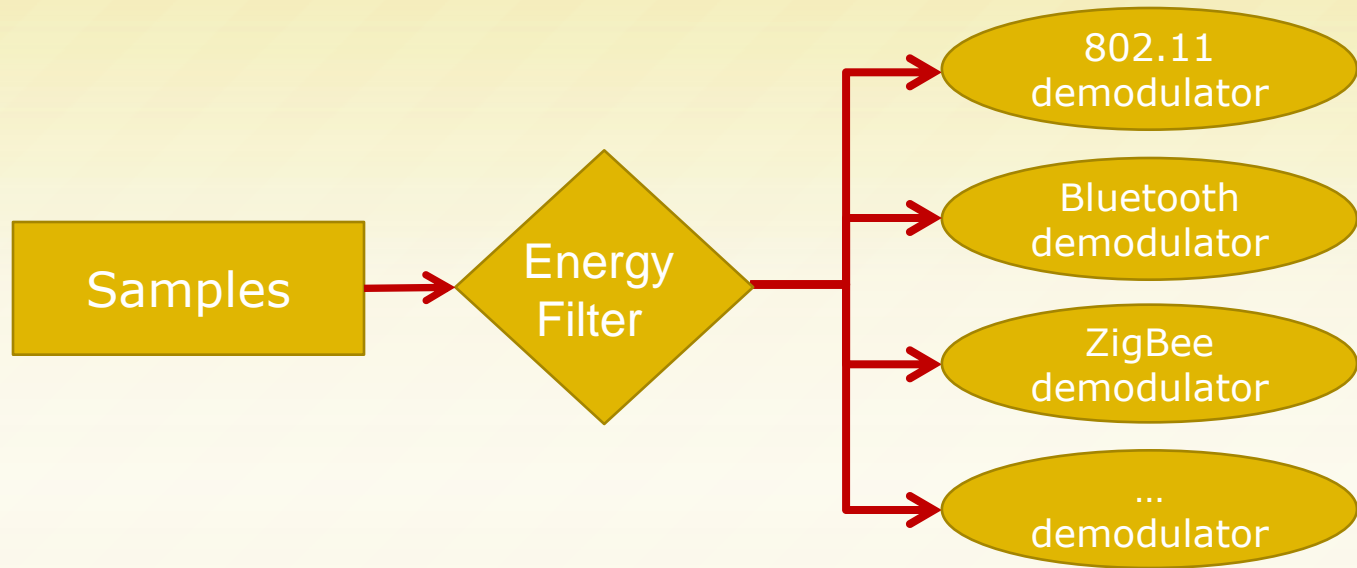
Tcpdump Approach

- Tcpdump – easy demultiplexing/decoding
 - Each layer specifies the next layer protocol



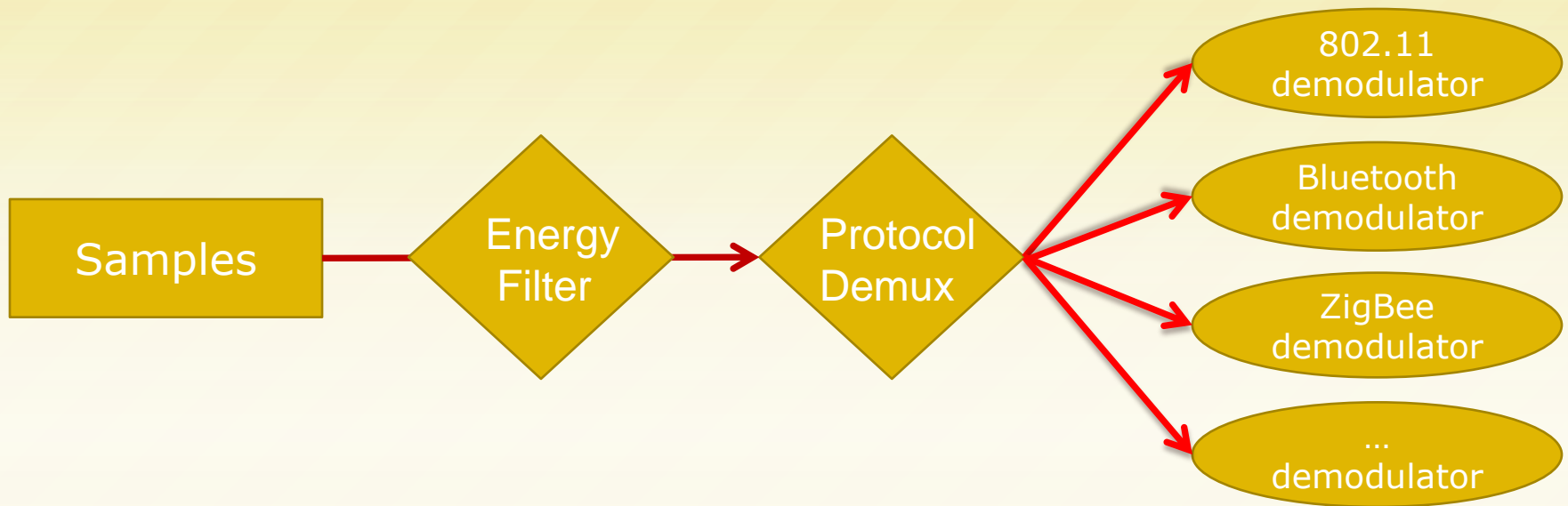
- Wireless medium shared by diverse datalink protocols
 - Physical layer gives no information on the nature of active transmissions

A Solution?



- Start with a software-defined radio
- Why is this unstatifactory?
 - Demodulation is costly
 - Especially bad if medium utilization is high ☹

An even better solution...



- Protocol demux quickly identifies protocol type
 - Basically adds “protocol” tag that tcpdump can use
- How to build a demux that is less expensive than demodulation?
 - Can accommodate error and latency → enables shortcuts
 - Optimize common techniques

Relevant Features for Detection

Protocol		Timing (μ s)		Phase (Modulation)		Channel width
802.11	(Mbps)	Slot	SIFS	Scheme	Spreading	22 20
	b (1)	20	10	DBPSK ^a	Barker	
	b (2)	20	10	DQPSK ^a	Barker	
	b (5.5/11)	20	10	DQPSK ^a	CCK	
	g	9	10	OFDM ^{b,c}		
Bluetooth		Slot 625		GFSK	FHSS	1
802.15.4 (ZigBee)		Slot 320	IFS 192/600	QPSK		5
Residential Microwave		AC cycle 16667/20000				10-75

^aPreamble is sent using DBPSK

^bCTS-to-self packets use one of the 802.11b rates

^cUses BPSK, QPSK, 16-QAM or 64-QAM for the subcarriers

So what next?

- Current prototype very limited by SDR hardware
 - Better SDR hardware now available
- Tcpdump != diagnosis
 - Enables collection of data
 - Observation and explanation of adverse interactions
 - Need to still fix things
 - Build signatures of poor interactions and appropriate corrective actions
- Not the only tool needed → e.g. active tests
- 802.11/other wireless cards exposing greater information
 - How far can this get us?
- Is the network working correctly?