Metronome: Spectrum Sharing in Heterogeneous Wireless Networks

Ramki Gummadi (MIT)

Joint work with Hari Balakrishnan (MIT)
Srinivasan Seshan (CMU)

Motivation

- Heterogeneous wireless n/w increasing
  - ISM: 802.11, ZigBee, Bluetooth,…
- New spectrum usage opportunities
  - 700 MHz US TV bands
  - 3.65 GHz band with “lite” licensing
- MAC in homogeneous n/w, but heterogeneous?

Problem statement

- Better spectrum sharing in heterogeneous n/w
  - Goal: High aggregate throughput
- Support different constraints for different n/w
  - Primary/secondary interference in whitespaces
  - 802.11, ZigBee coexistence

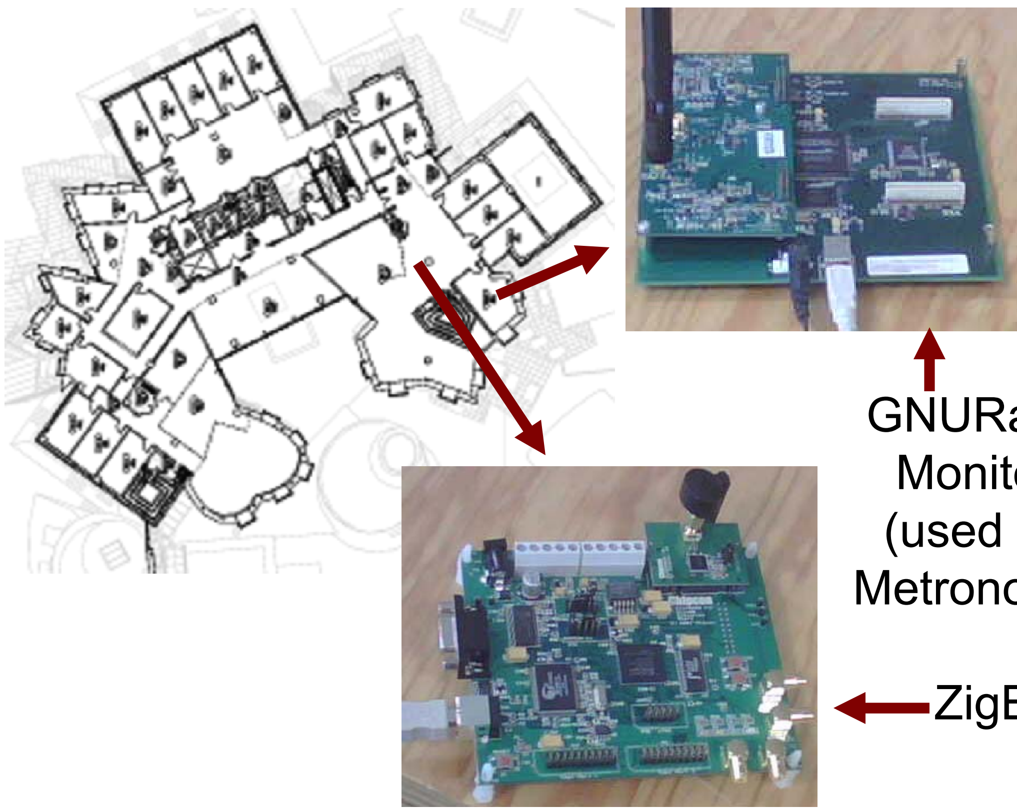
Status quo

- Consistently poor performance with some n/w combinations

- Interference impact varies with n/w
  - Different power levels
  - Different workloads
  - Different frequency widths

- Sub-optimal throughput in aggregate

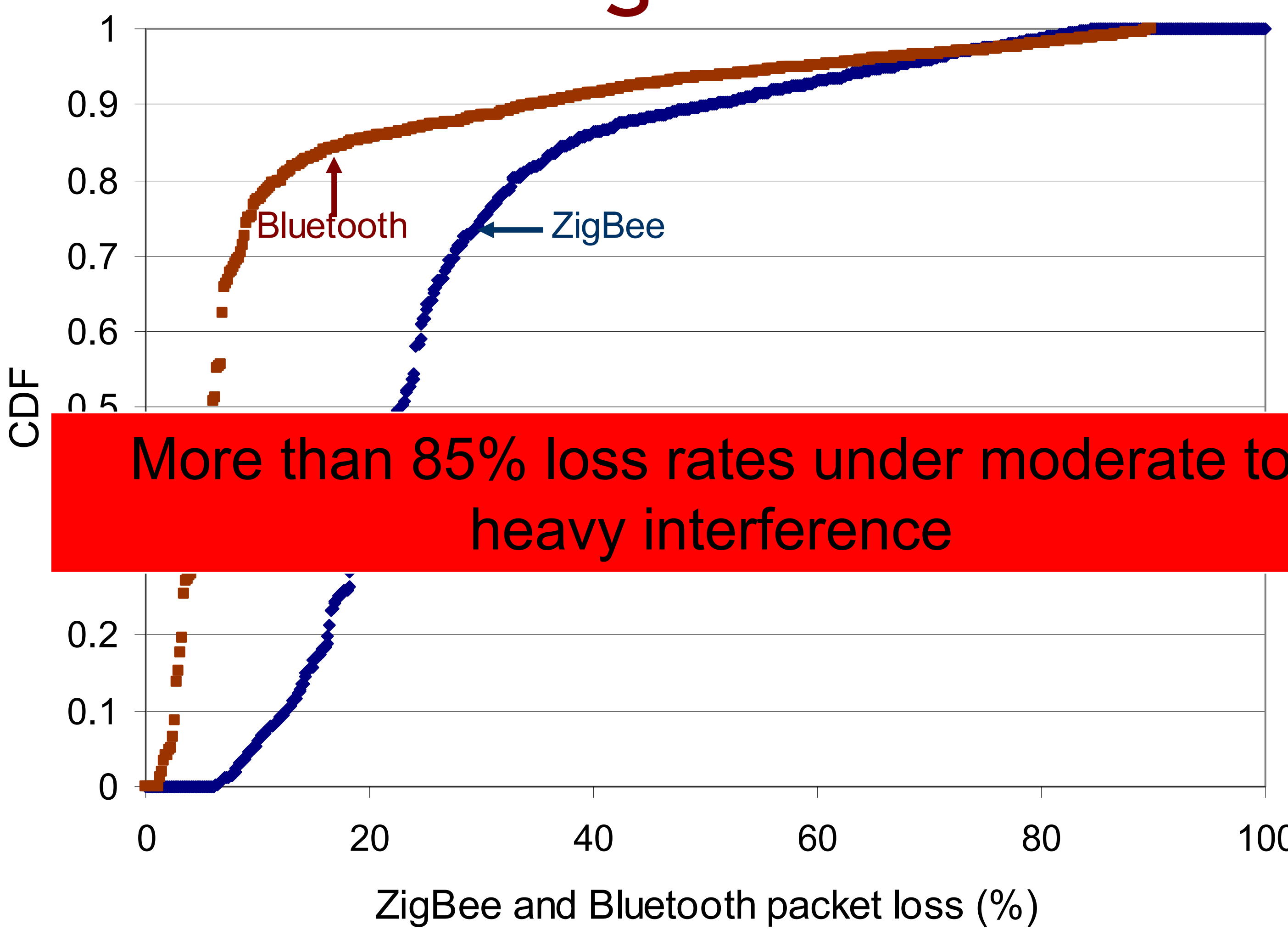
Experimental Setup



GNURadio Monitor (used by Metronome)

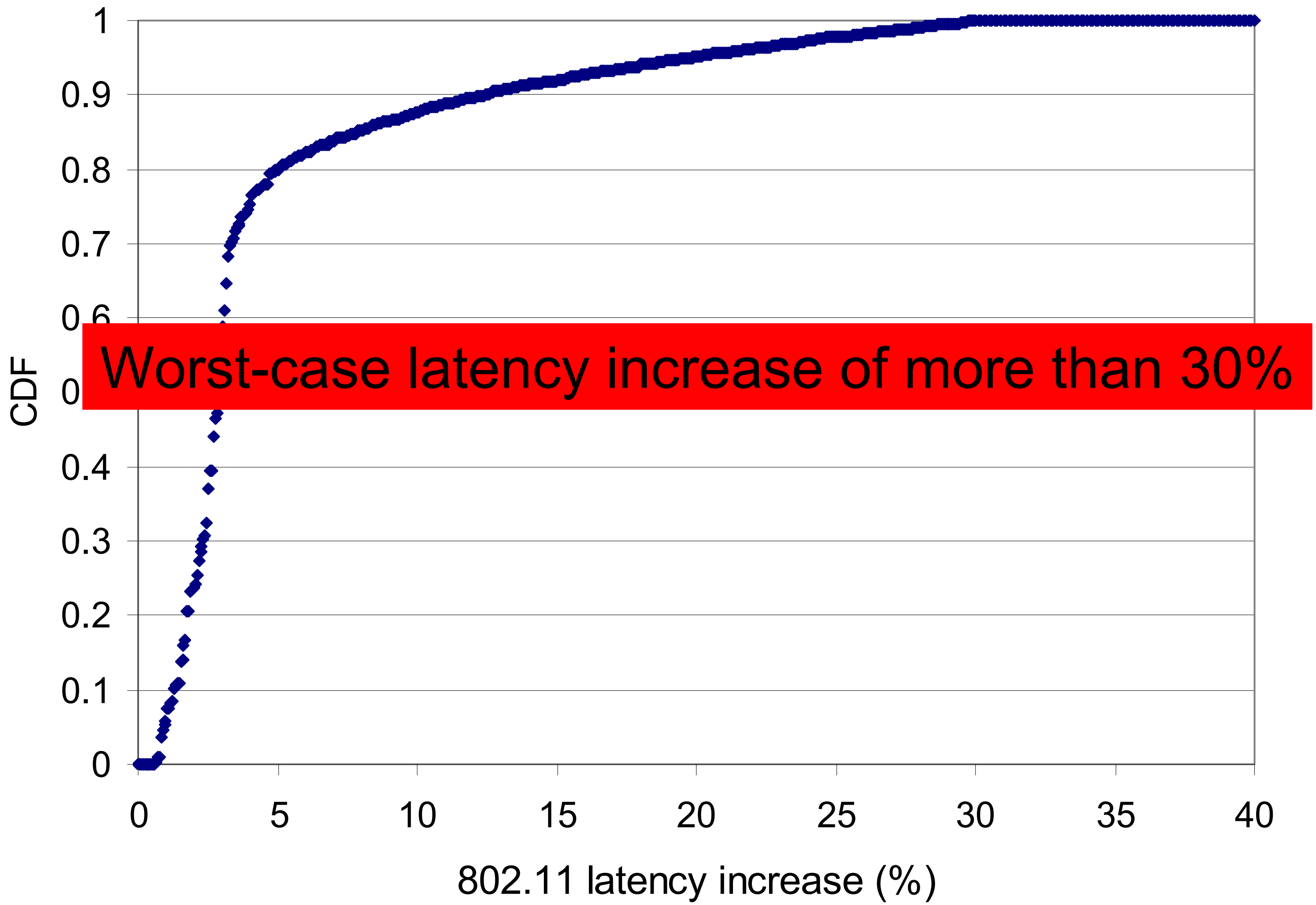
ZigBee

802.11 interference on Bluetooth, ZigBee



More than 85% loss rates under moderate to heavy interference

ZigBee interference on 802.11 latency



Mitigation approaches

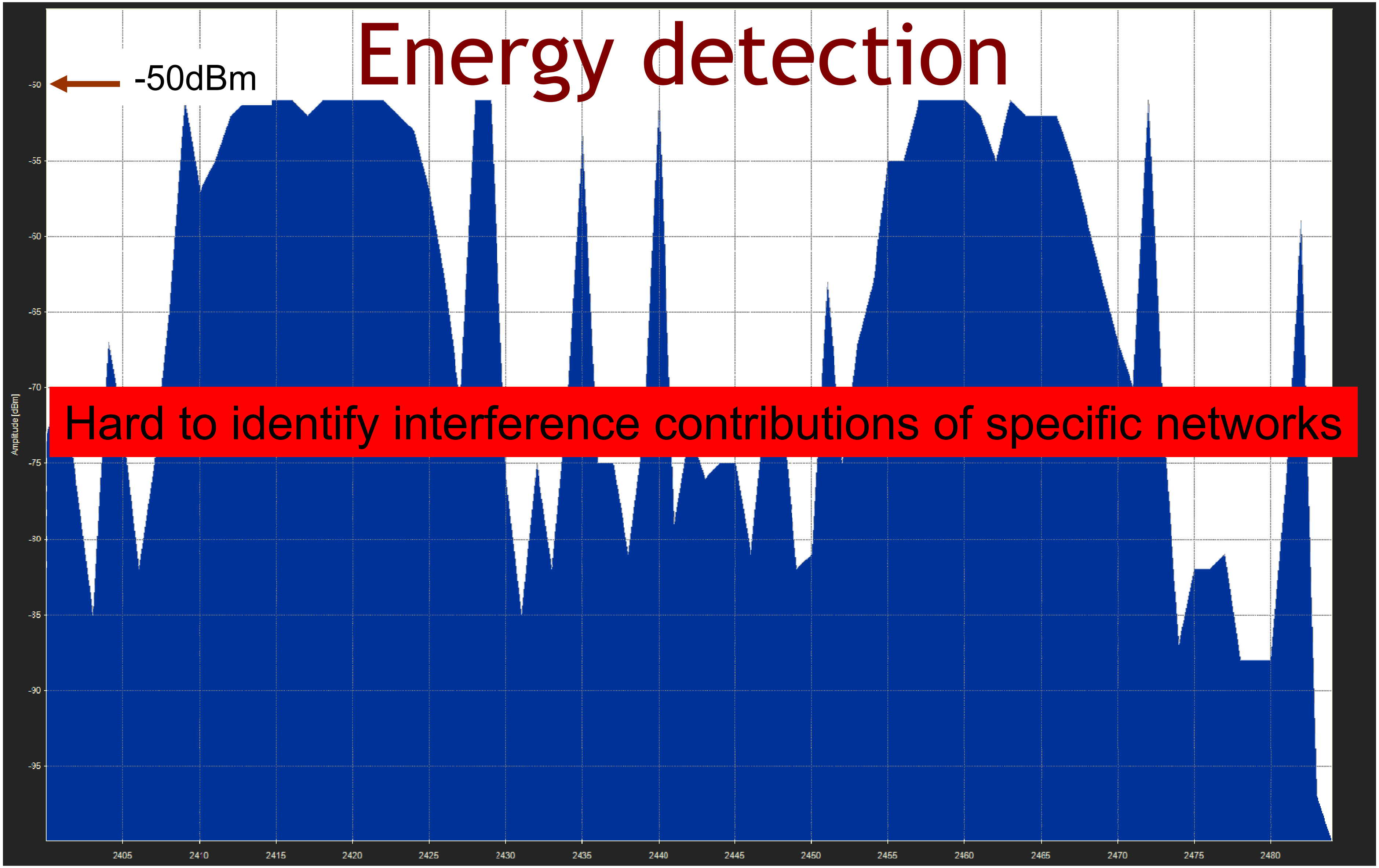
- Hard to identify and isolate interference through energy detection

- Manually configuring transmitter settings once interference is detected is hard

Energy detection

- Interference isolation through energy detection
  - Use spectrum analyzers
    - Inability to identify the signal and interference contributions of each network type within measured energy
  - Use multiple protocol-specific scanners
    - One piece of hardware for each network type
    - No global interference picture
    - Inability to identify a particular 802.11 device as the interference source (multiple devices on the same channel)

Energy detection

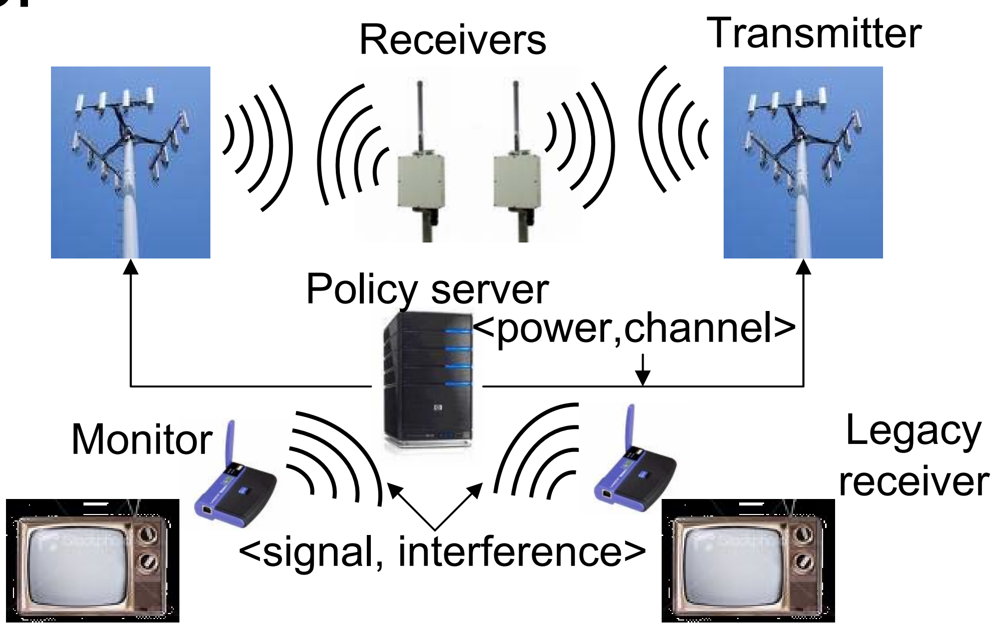


2.400GHz

2.483GHz

Metronome: Architecture for spectrum sharing

- Monitors sprinkled close to receivers
- Policy language for regulating sharing
- Transmitter parameters controlled by the policy server



Architectural Requirements

- Support different policies
- Capture receiver-perceived interference
- Support legacy communication systems
- Support evolution
- Control individual participating transmitters
- Require no new wireless network protocols

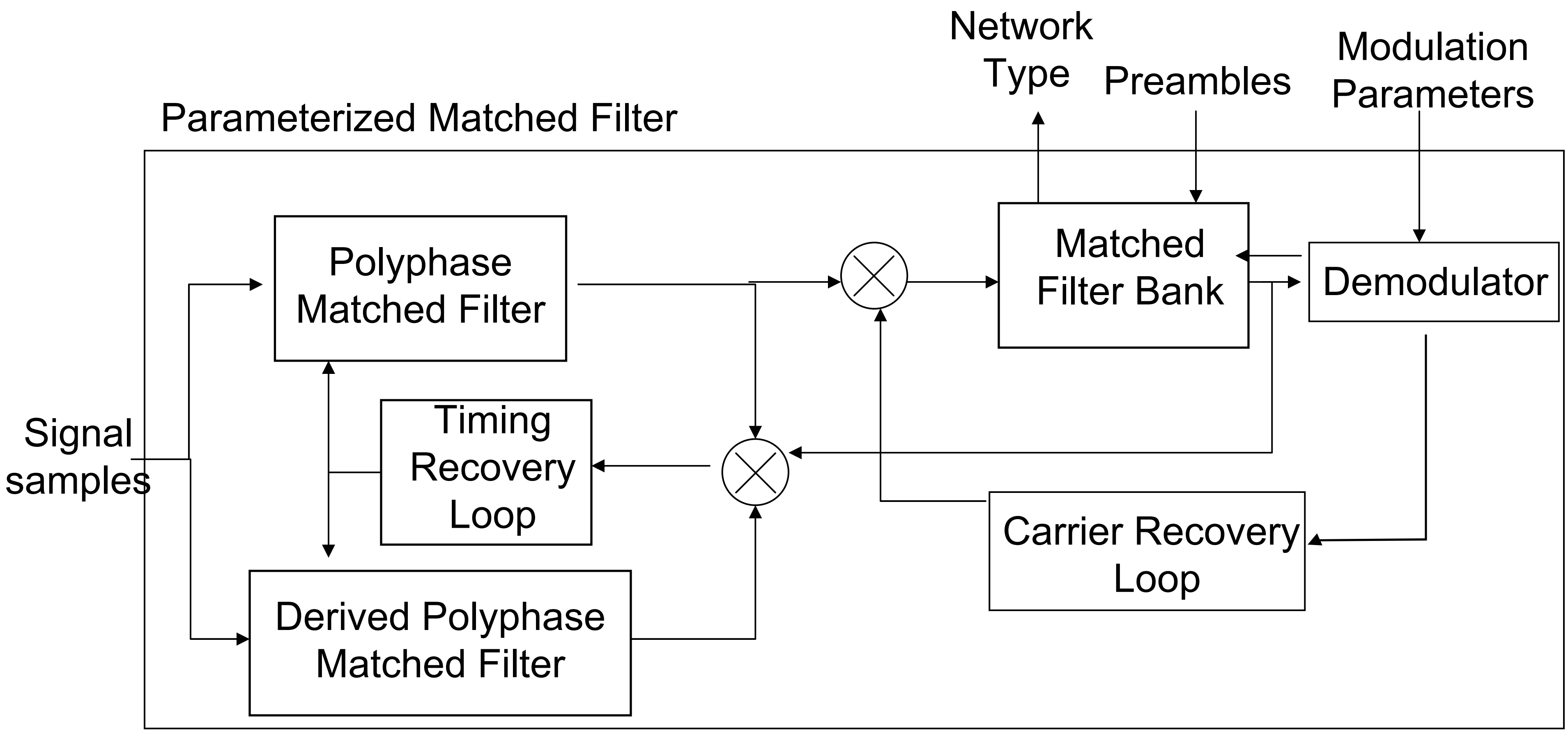
Measuring interference contributions

- Find potential interferers
  - Participating transmitter broadcasts a special beacon
  - Monitors measure interference power of each transmitter
- Identify individual transmitters
  - Interference temperature at monitors may only exceed when multiple transmitters are active
  - When monitors notice high interference temperatures, they report to the policy server, which asks transmitters to timestamp their transmissions
  - Uses these timestamps to identify what is the average signal and interference power of each transmitter at a monitor

Parameterized Matched Filtering

- Extensible “universal” signal and interference power detection at monitors
- Maximizes signal detection by looking for known preambles in packets
- Extensible because signal waveforms can be parameterized as <modulation type, modulation encoding, constellation size, pulse shape> tuples

Components of the parameterized matched filter



Policy Language

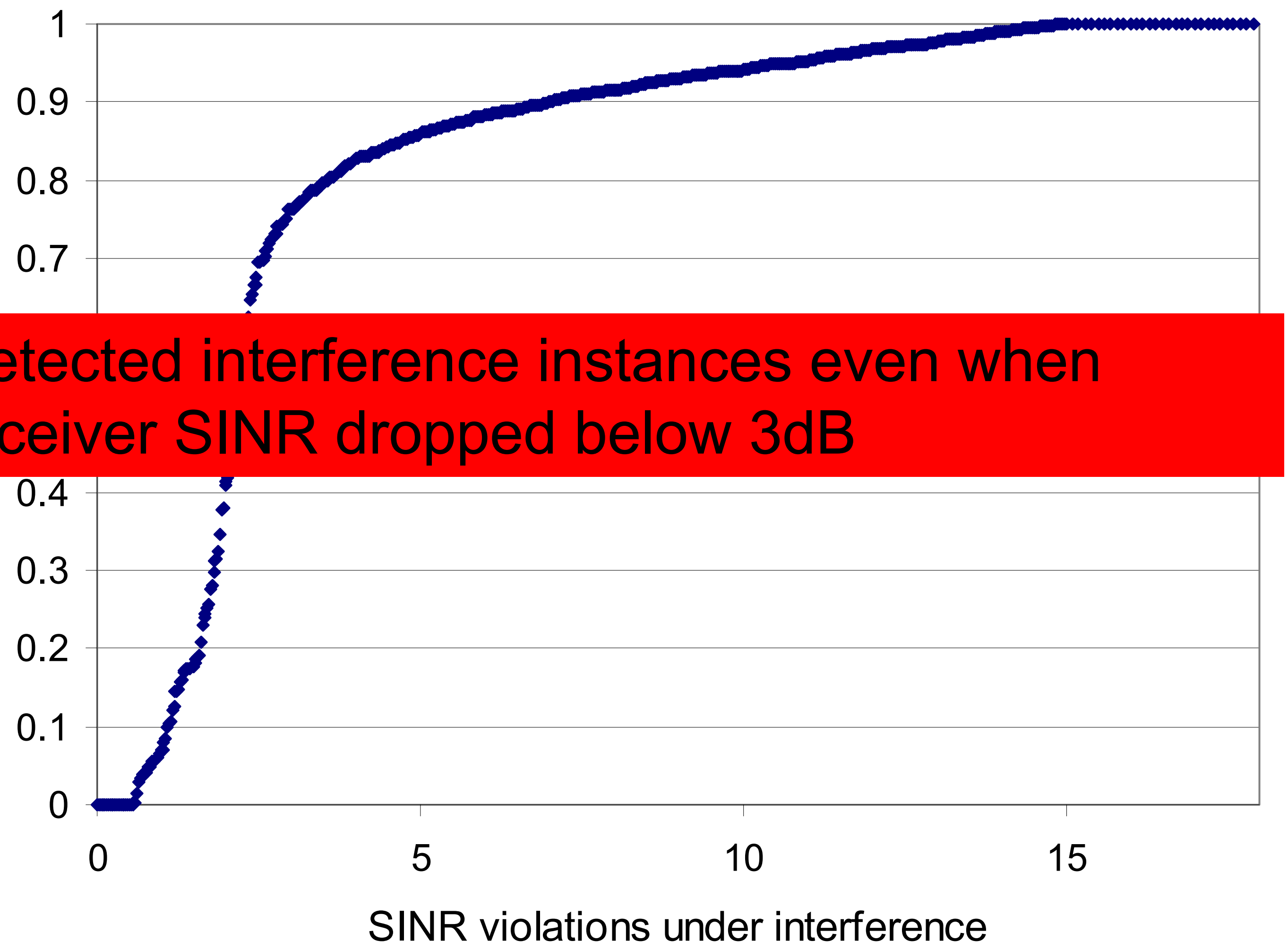
- Set of linear SINR constraints and goals
- LP-based algorithm for maximizing the goal by solving for transmitter parameters of power and channel

$$
\begin{array}{l}
\textbf{for } i \leftarrow 1 \textbf{ to } N \\
\textbf{do} \left\{ \begin{array}{l}
\textbf{for } c \leftarrow c_{\min} \textbf{ to } c_{\max} \\
\textbf{do} \left\{ \begin{array}{l}
\textbf{for } j \leftarrow 1 \textbf{ to } N \\
\textbf{do} \left\{ \begin{array}{ll}
\textbf{if } i = \text{transmitter}(j) & \\
\quad \textbf{then } \text{signal}[j] \leftarrow H[i][j][c]p[i] & (i) \\
\textbf{if } i = \text{heterogeneous}(j) & \\
\quad \textbf{then } \text{interfer}[j] += H[i][j][c]p[i] & (ii) \\
\text{SINR}[i,j,c] = \text{signal}[j] - \text{interfer}[j] &
\end{array} \right. \\
\begin{array}{lll}
\text{Find} & p[i] & \\
\ni & \max \text{policy}(\text{SINR}[i,j,c]) & (iii) \\
\text{given} & \forall i \; p_{\min}[i] \leq p[i] \leq p_{\max}[i], \& & (iv) \\
 & \text{SINR}_{\min}[] \leq \text{SINR}[] \leq \text{SINR}_{\max}[] & (v)
\end{array}
\end{array} \right.
\end{array} \right. \\
\textbf{return } (c[i], p[i] \ni \max_{c[], p[]} \text{policy}(\text{SINR}[]))
\end{array}
$$

Evaluation

- Implemented for 802.11, Bluetooth, ZigBee using USRPs
- Evaluated in a busy office floor over 20 days
- Five key questions
  - Can we isolate interfering transmitters?
  - Can we provide flexible policy support?
  - Can we deal with external interference?
  - Is Metronome scalable?

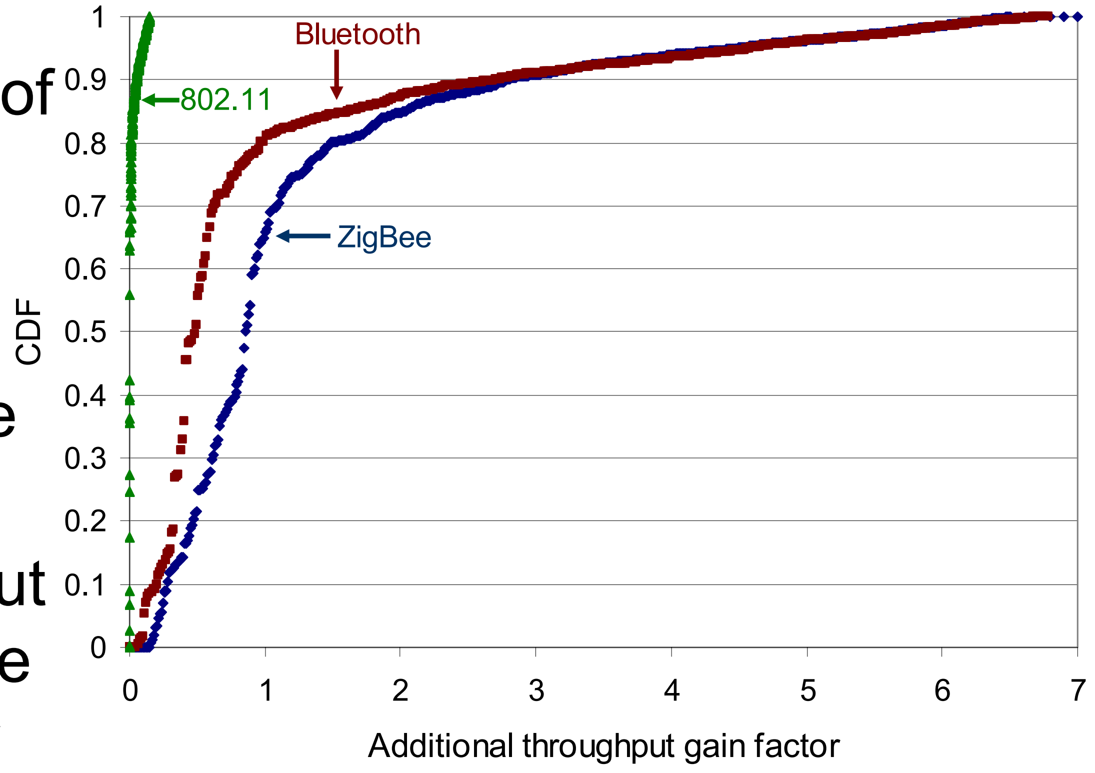
Isolating interfering transmitters



Detected interference instances even when receiver SINR dropped below 3dB

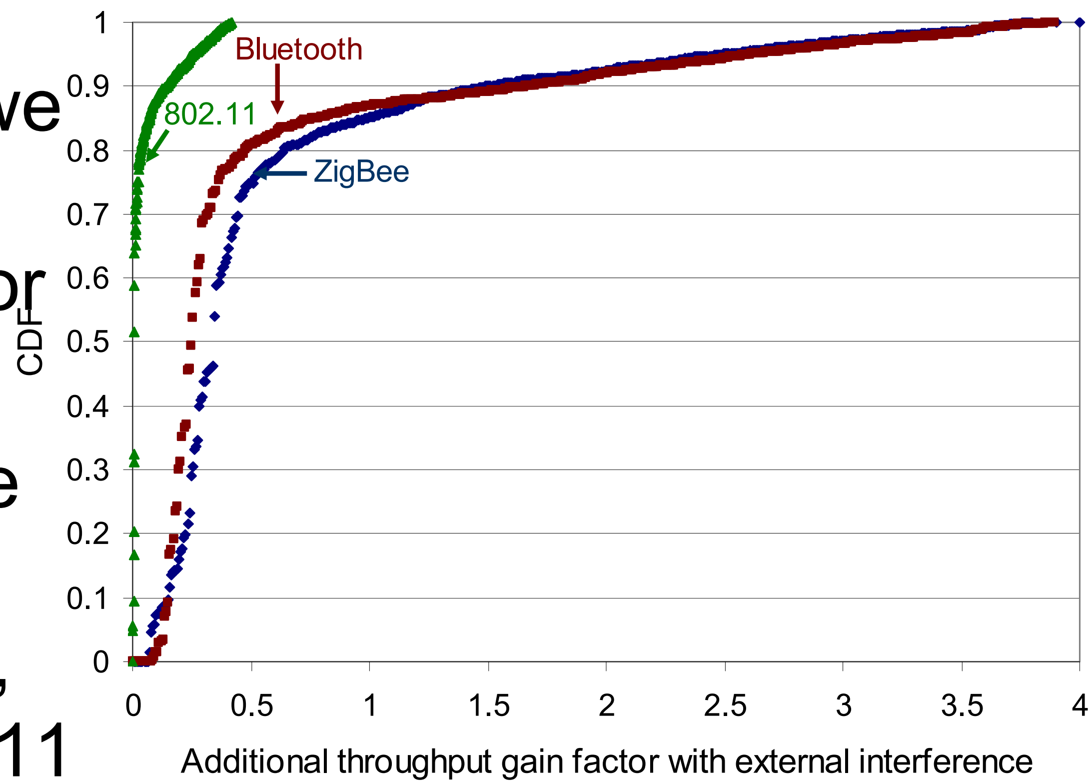
Policy Support Flexibility

- Specified policy of 18dB SINR for ZigBee and Bluetooth, leaving rest of spectrum to 802.11
- Policy server selected optimal channels and power levels for ZigBee and 802.11
- More than 6x throughput improvement for ZigBee and Bluetooth, 15% for 802.11



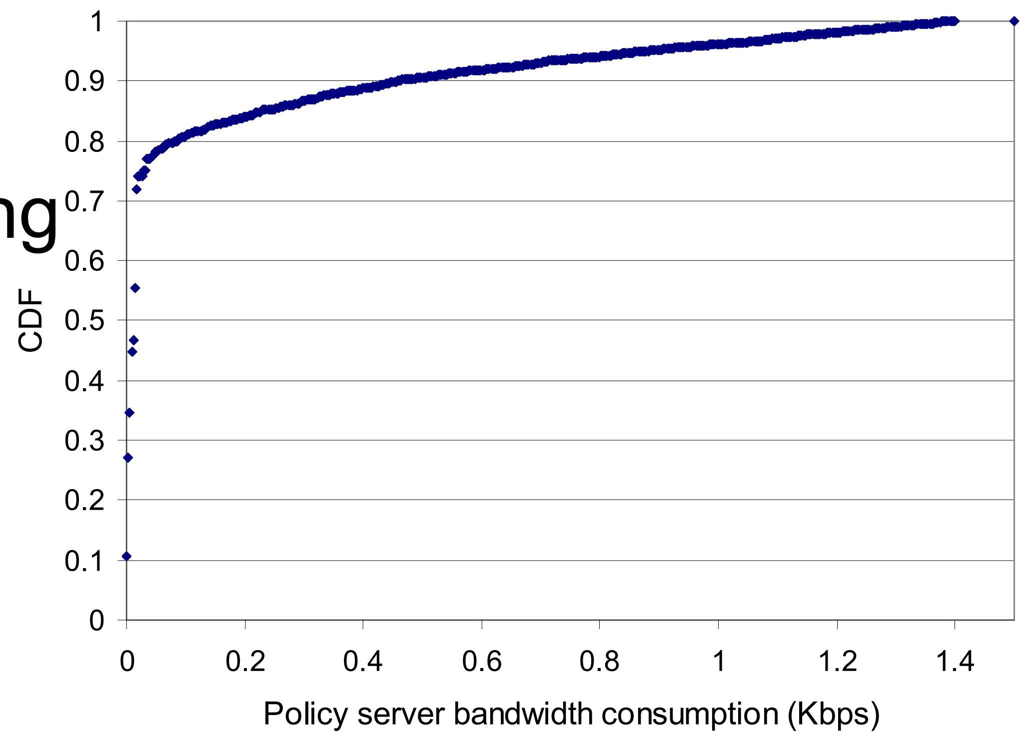
Dealing with external interference

- In order not to penalize 802.11, we simultaneously constrain SINRs for all three networks
- So, constraints are 18dB SINR for ZigBee, Bluetooth, and 10dB for 802.11 (for loss rates < 10%)



Metronome Cost and Scalability

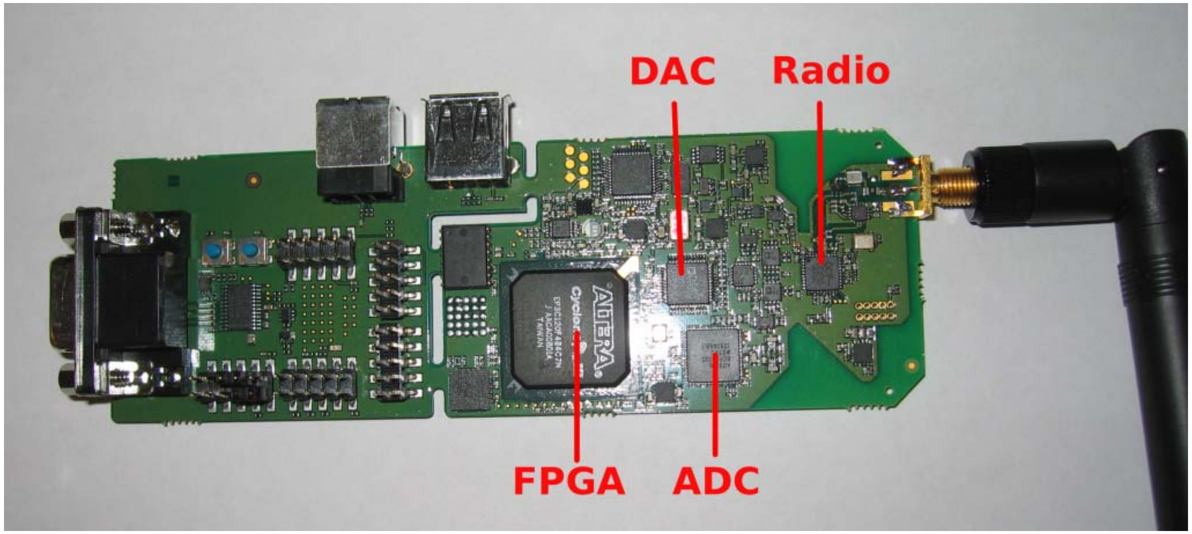
- Needed six monitors for 18,000 sq. feet
- Matched filter in the monitors can be using cheap (~ $50/node) ADC hardware
- Policy server consumes low bandwidth



Conclusions

- Metronome is an extensible architecture for spectrum sharing among heterogeneous devices
- Incorporates flexible and expressive policy language
- Allows interference to be isolated and transmitter parameters to be optimally set

Demo



- 802.11 bit-rates
- Programmable down to the RF
  - More flexible than commodity cards
- Software: How to write protocols easily and correctly?
- Hardware: How to achieve high throughput, low latency, low power?