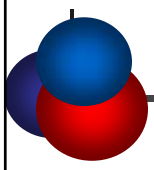
Dealing with Interference on Today's Wireless Hardware



Peter Steenkiste
Departments of Computer Science and Electrical and Computer Engineering
Carnegie Mellon University

intel

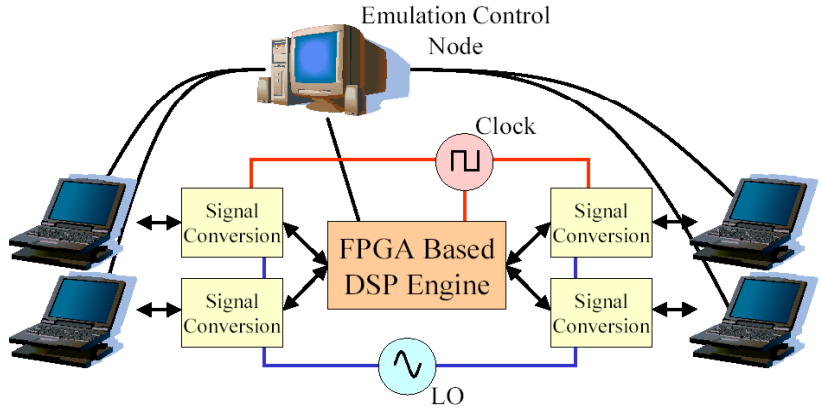
1

Outline

- Context
  - Self-managing chaotic wireless networks
  - Wireless network emulator testbed
- Interference model (Xi Liu, Srini Seshan)
  - A networking view
- Auto transmit rate selection (Glenn Judd, Xiaohui Wang)
  - Interference a non-issue (really)
- Auto transmit power selection (Xi Liu, Srini Seshan)
  - Interference a big issue

2

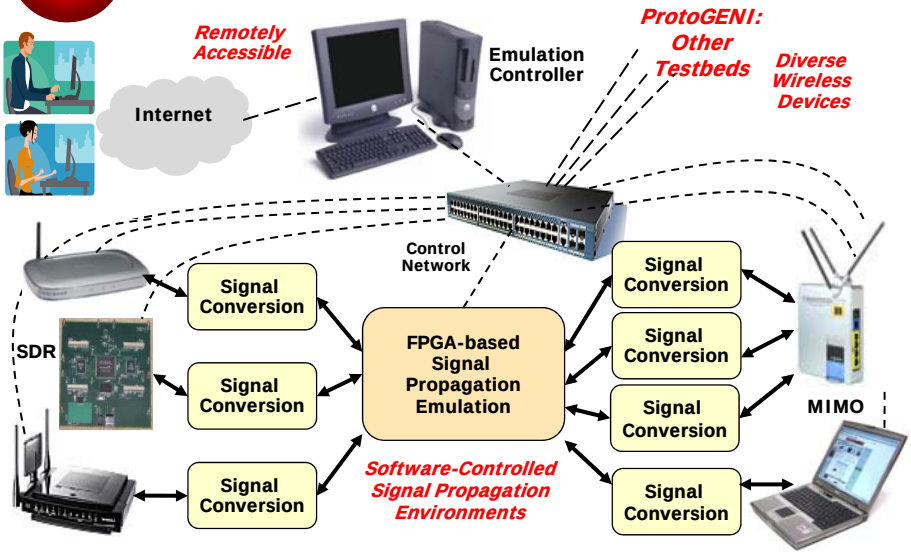
Testbed based on Signal Propagation Emulation



- Real hardware ➔ high degree of realism
- Digital emulation of channels ➔ full control
- Isolated from environment ➔ fully repeatability
- Programmable ➔ very diverse experiments

3

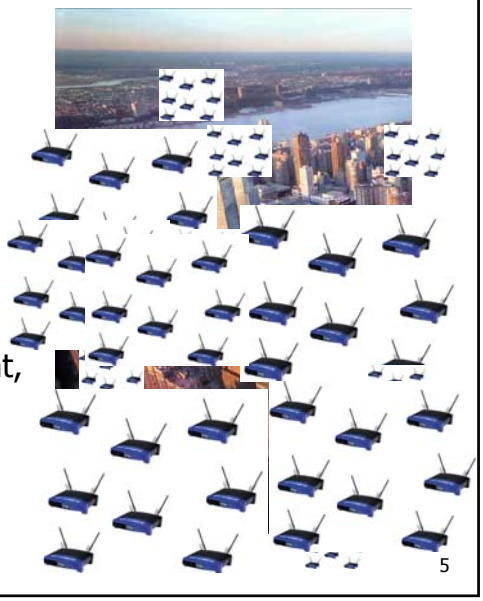
Current System



Chaotic Wireless Networks

- Unplanned:
  - Independent users set up APs
  - Spontaneous
  - Variable densities
  - Other wireless devices
- Unmanaged:
  - Configuring is a pain
  - ESSID, channel, placement, power
  - Use default configuration

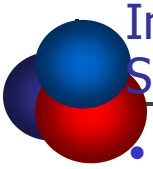
→ "Chaotic" Deployments



5

Chaotic Project Roadmap

- Goal: self-configuration and self-optimization
- What can we do with today's commercial hardware?
  - Automatically tune parameters to optimize network performance
  - E.g.: channel, transmit power, transmit rate
- Leverage emerging wireless technologies
  - Tomorrow's commercial hardware
  - Software defined radios, smart antennas
- Optimize use of the scarce wireless spectrum
  - Dynamic spectrum sharing

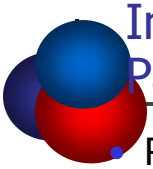


Interference: So Many Models to Choose From!

- Circle model => Use low power levels reduce interference
- SINR model => Use higher power levels provides better performance by reducing effects of noise

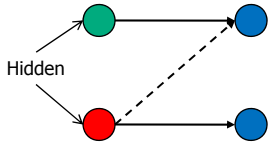
$$SINR = \frac{S}{I + N}$$

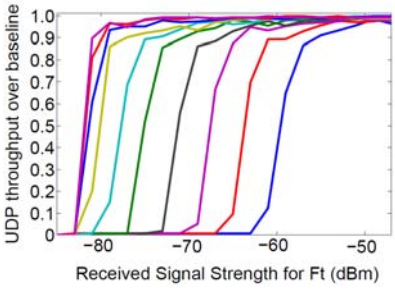
- Capture effect is key: Can higher signal power overcome effect of interference?
  - What does real hardware do?



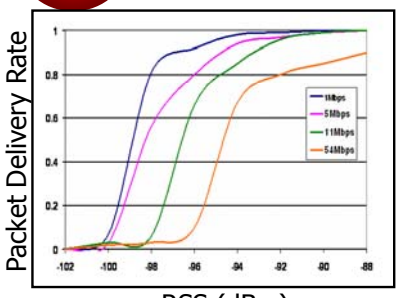
Impact of Interference on Packet Reception Rate

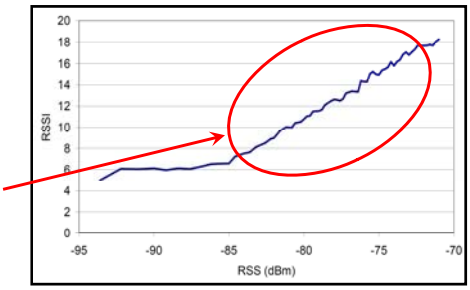
- Ran experiment on wireless emulator
  - Atheros cards + create hidden terminal
- Measure packet success rate as function of transmit power for different levels of interference
  - Interference changed in steps of 4db
- SINR formula holds
  - Increasing interference = reducing power





Automatic Transmit Rate Selection

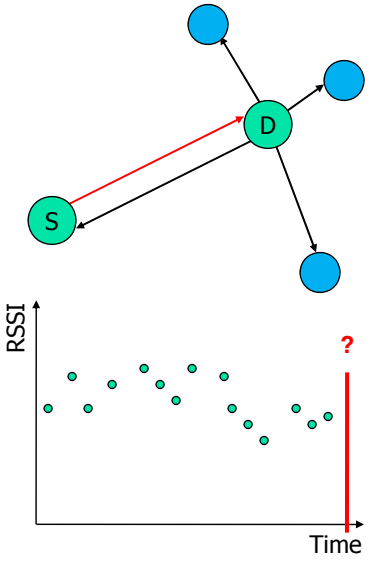




- Best transmit rate depends on the SINR
  - Signal to noise and interference ratio
- Can be estimated on 802.11 cards based on RSSI
  - Can measure received signal strength using RSSI
  - Can exchange information about transmit power, noise, etc.

Charm: Channel-Aware Rate Selection

- Leverage channel reciprocity: overhear packets sent by destination to learn about channel conditions
  - Build history of path loss for each channel
- When transmitting packet, use path loss history to "predict" path loss
- Select best transmit rate from look up table
  - Per destination rate threshold table
  - Thresholds dynamically adjusted based on experience



The Formulas

$= PL_{(Rx\ to\ Tx)}$ (*Reciprocity Theorem*)

$$RSS_{(at\ Rx)} = P_{Tx} + G_{Tx} - PL_{(Tx\ to\ Rx)} + G_{Rx}$$

$$PL_{(Rx\ to\ Tx)} = P_{Rx} + G_{Rx} + G_{Tx} - RSS_{(at\ Tx)}$$

$$RSS_{(at\ Rx)} = RSS_{(at\ Tx)} + P_{Tx} - P_{Rx}$$

Constant

$$SINR_{(at\ Rx)} = RSS_{(at\ Rx)} - N_{Rx}$$

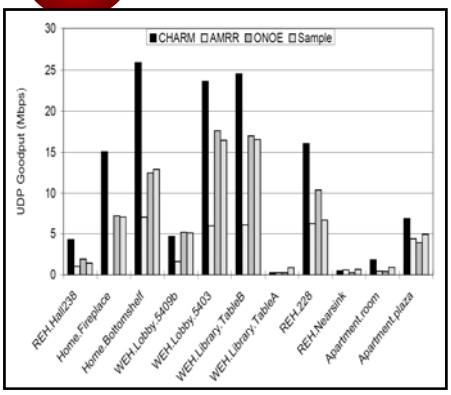
Note: no I_{Rx}
No interference

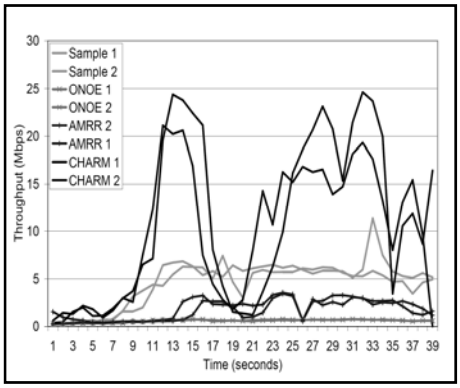
But hold your guns, please!

P_{Tx}/P_{Rx}	: Transmit Power at transmitter/receiver
G_{Tx}/G_{Rx}	: Transmit Antenna Gain/Receive Antenna Gain
PL	: Path Loss

11

Charm Performance

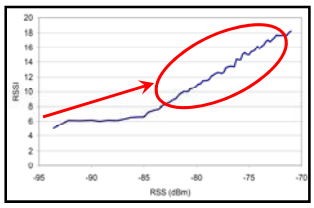


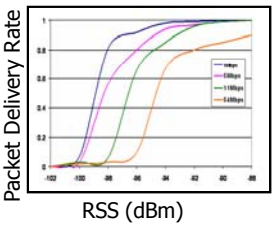


- Charm performs better in both static and dynamic scenarios

Dealing with Real Hardware

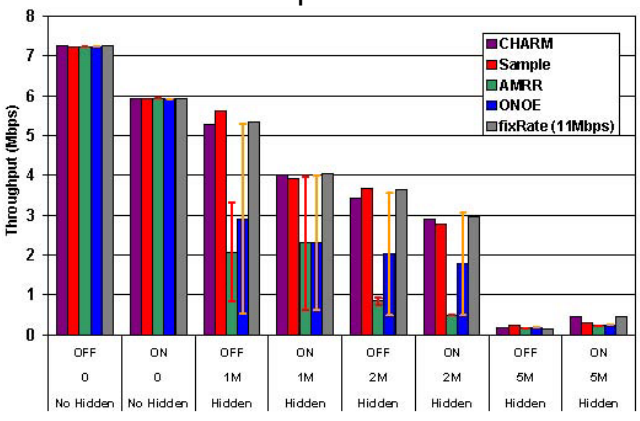
- RSSI versus RSS
  - Fairly linear but there can be an offset
    - Automatically dealt with by auto-tuning
  - Some noise in RSSI measurements
    - Filter out with "time-aware" algorithm
- Interference can affect Tx RSSI reading and SINR at Rx
  - Not really – lots of reasons
- Lack of calibration of transmit power, noise values, RSSI offset, etc.
  - Automatically dealt with by auto-tuning
- Calibration of xmit rate thresholds
  - Adjust automatically based on observed success/failure of transmissions
  - Deals with above calibration issues

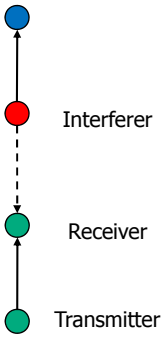




Transmit Rate Selection and Hidden Terminals

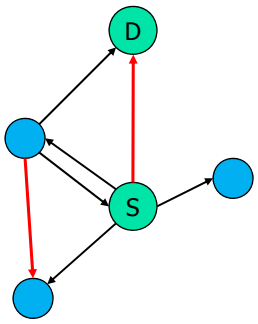
- Some rate selection algorithms perform poorly in hidden terminal situations
  - Collision -> reduce rate -> increased chance of collisions
- Create simple hidden terminal scenario on emulator





Transmit Power Control to Minimize the Effect of Interference

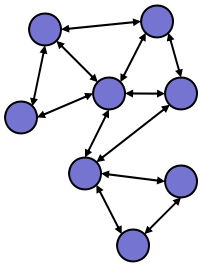
- Simple idea: reduce transmit power to minimum needed to reach destination
  - Based on SINR
- Does not work!
  - Interference is not constant but affected by transmit power used by other nodes
  - Reducing transmit power makes receiver more susceptible to interference

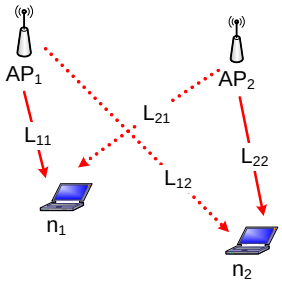


- Simple experiment: if all nodes cut transmit power in half, SINR stays the same
  - Assuming noise is not a concern

15

Automatic Power Control: Concepts





- Any transmission creates interference on all links
- Captured in pair-wise interference conflict graph:
  - Nodes are wireless links
  - Edge if simultaneous transmission not possible
- Concurrent transmission is possible if

$$SINR_1 + SINR_2 \geq 2 * SINR_{threshold}$$

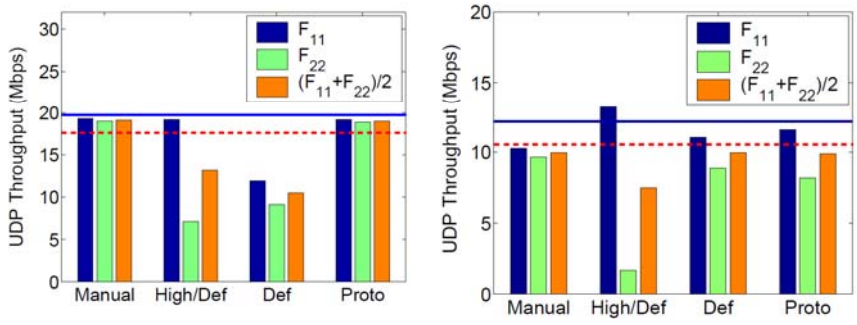
Power Control Algorithm

- Greedily remove edges from conflict graph by adjusting transmit power for links
  - Converges when no more edges can be removed
- Must also adjust "Clear Channel Assessment" threshold
  - Done in a separate phase using variant of existing algorithm (altruistic Echos)
- Centralized algorithm is quite simple - distributed algorithm is a bit more involved
  - Nodes exchange information about transmit power and RSS observed from neighbors
  - Each node operates on local conflict graph

17

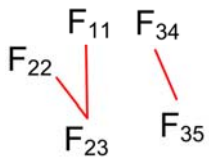
UDP Throughput

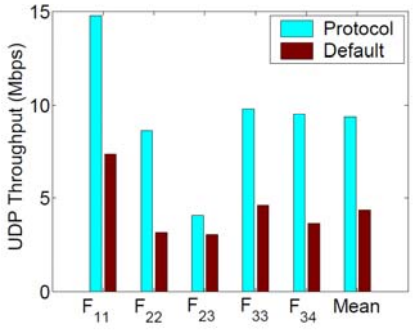
- 36Mbps: F_{11} interferes with F_{22} using default txpower
  - Concurrent transmission possible by reducing F_{11}'s txpower
  - Not fair even with default low CCA
- 48Mbps: no concurrent transmission
  - fairness of the protocol is slightly worse because of relatively high CCA
  - fairness can be achieved by reducing F_{11}'s txpower



Experiment with 8 nodes

- F_{11} interferes with F_{23}, but not with F_{22}
- Pair-wise assumption inaccurate on F_{34}
- Default behavior is better than expected

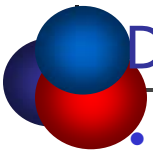




Hardware We Would Like

- Per-packet transmit power and CCA threshold
  - Only on Intel 2915/2200 with AP driver (kind of)
- Receiver threshold control separate from CCA
  - Tied together on above platform
  - Problem: cannot hear weak signals when CCA is high
- Accurate RSSI measurement and transmit power control
  - Depends on card: linear RSSI readings on Atheros, linear transmit power control on Intel card
  - But have per-card offsets

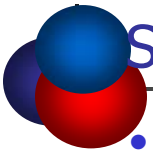
20



Dealing with Real Hardware

- Smoothing of RSSI readings
  - Both to deal with occasional spurious reading and to get estimates that are stable enough
- Sensitivity of CCA offset and transmit power
  - Need a certain margin to work reliably
- Calibration of transmit power control and RSSI readings
  - Automated protocol to account for card offsets
  - Really messy: 2 cards ➔ N cards
- Need to mix cards to get what you want
  - Really ugly – you don't want to know
  - Cards were optimized for today's WiFi

21



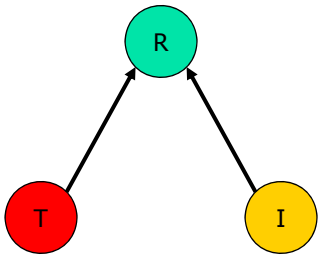
Summary

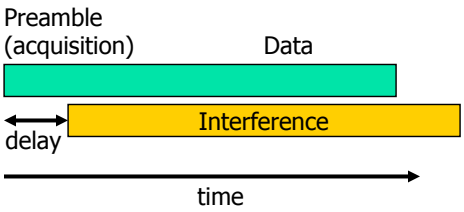
- Today's cards provide several readings and controls that are useful in fighting interference
  - RSSI, CCA, transmit power
  - Linear on some cards
- But need to deal with different offsets on cards and some noise, imprecision
- Requires on the fly calibration
  - Complexity depends on application
  - Not clear you can avoid this

22

More on Capture

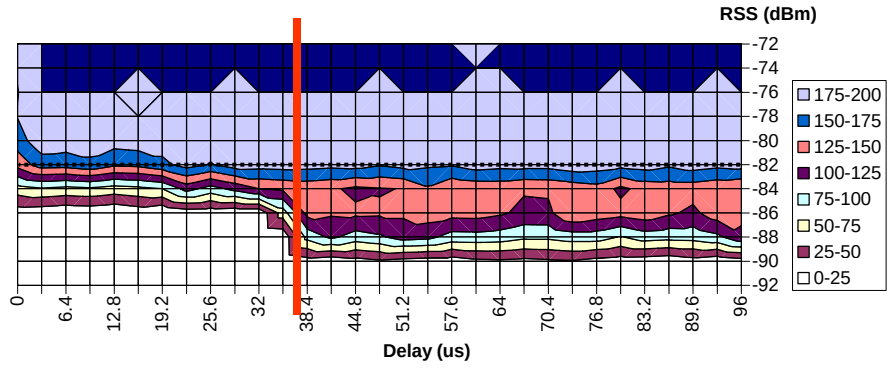
Capture vs. Collision Delay





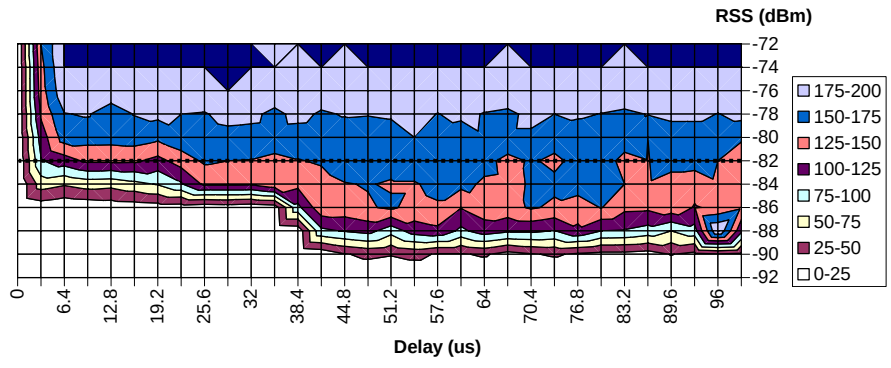
- Interference fixed at -82 dBm
- Change target signal strength and delay
- 1 & 2 Mbps have strong capture after acquisition
- 5.5 & 11 stick with the stronger signal
- These results for Prism II cards!

1Mbps



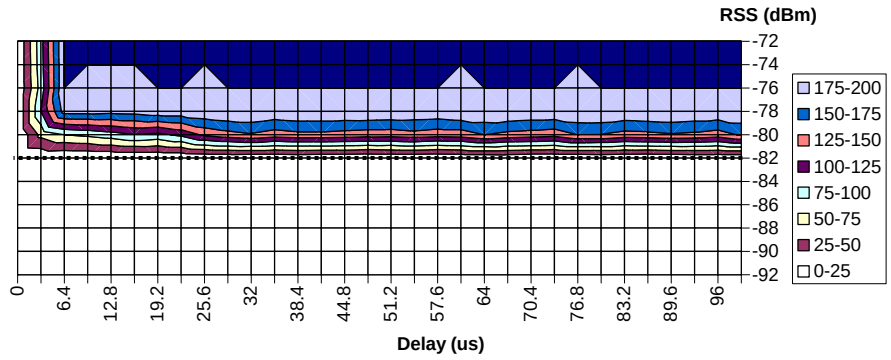
25

2Mbps



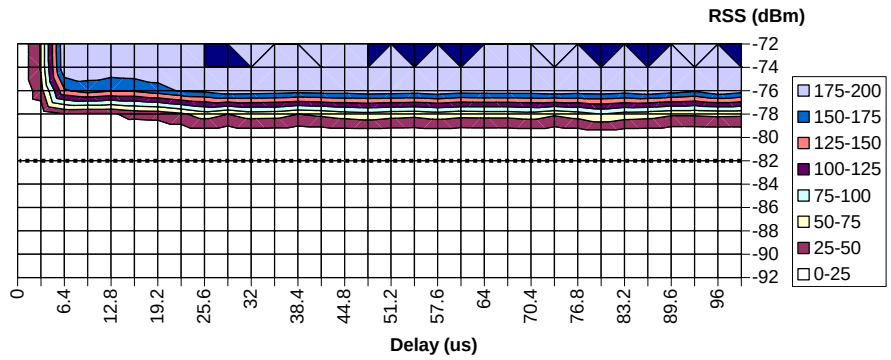
26

5.5Mbps



27

11Mbps

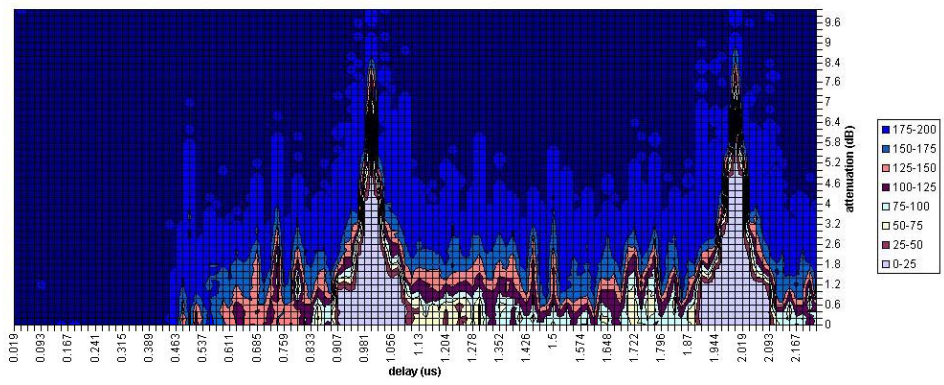


28

More on Multi-Path

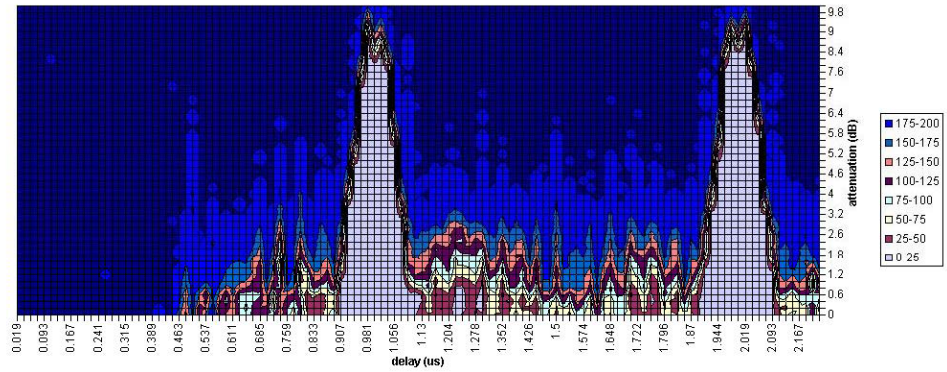
- Two-path channels
- Keep the primary path constant
- Change channel delay and strength of second path

29

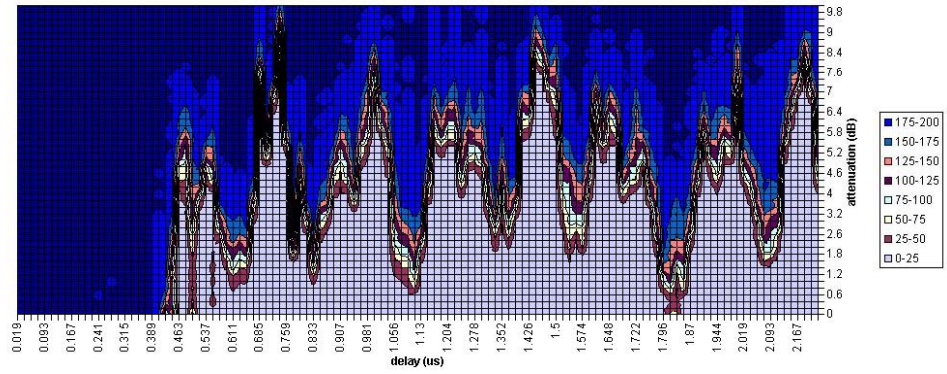


30

Impact of Delay and Attenuation (2 Mbs)

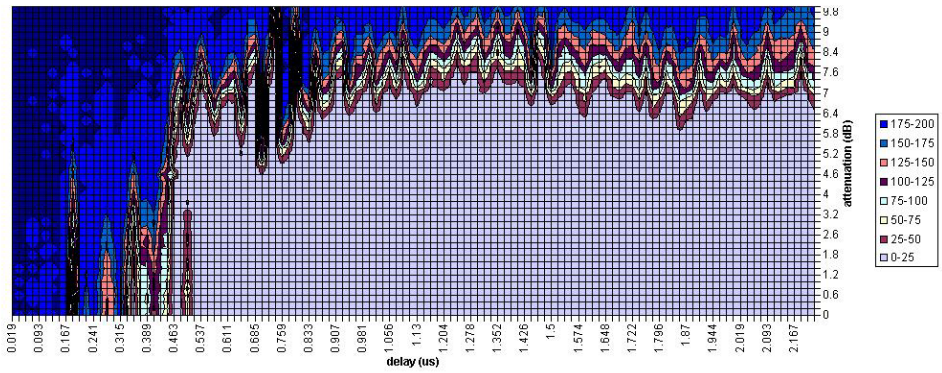


31



32

Impact of Delay and Attenuation (11 Mbs)



Card design for indoor environments