DSSS and FHSS

Mobile Communications Project - Group 6

**Urs Gerber, Gian-Luca Mateo,
Niclas Scheuing, Roger Stebler**

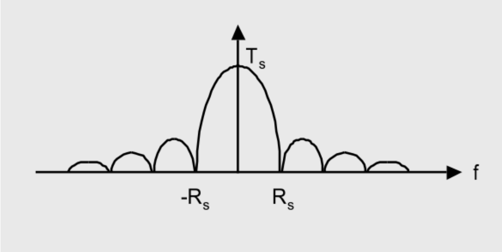
Outline

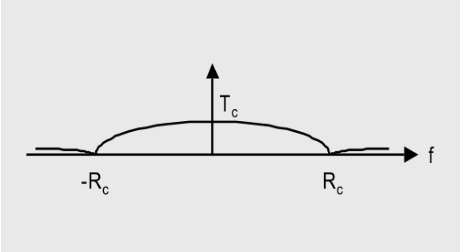
- Recall Spread Spectrum
  - DSSS
  - FHSS
- Architecture
- Implementation
- Test Scenarios
- Results
- Questions

Recall: Spread Spectrum

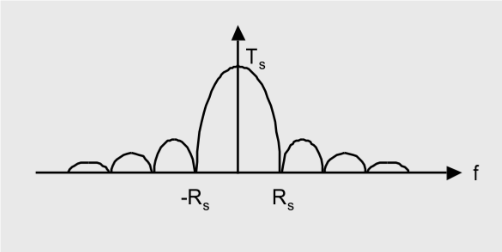
- Idea: increase bandwidth of signal by spreading
  - robust against narrowband interference
- Pseudo-random chip sequence pn

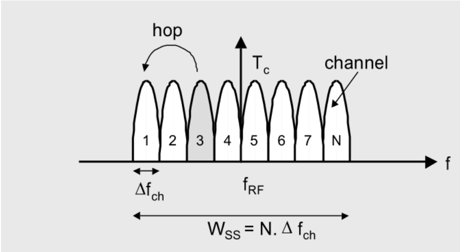
DSSS:





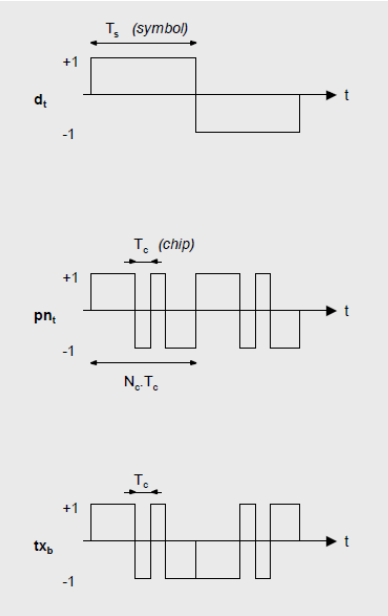
FHSS:

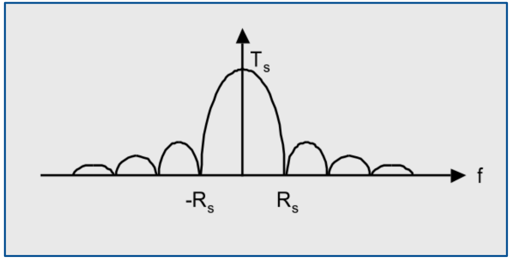


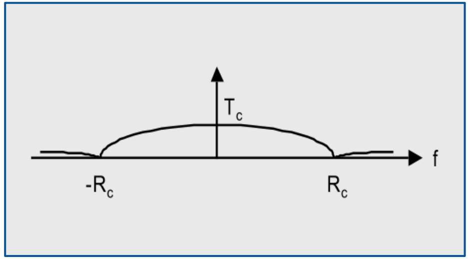


DSSS - Direct-Sequence Spread Spectrum

- Apply BPSK to $pn \oplus data$
- Spreads signal to $[-f_c, +f_c]$



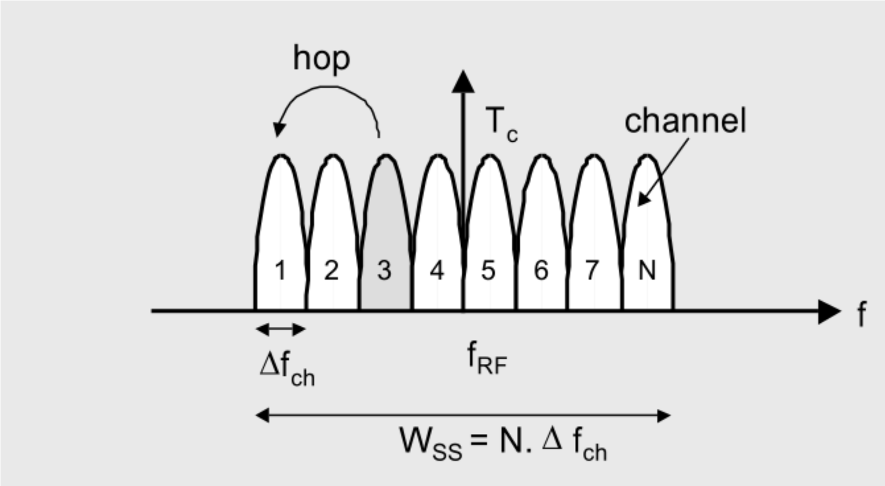




- NB: this is baseband
- our implementation is bandpass

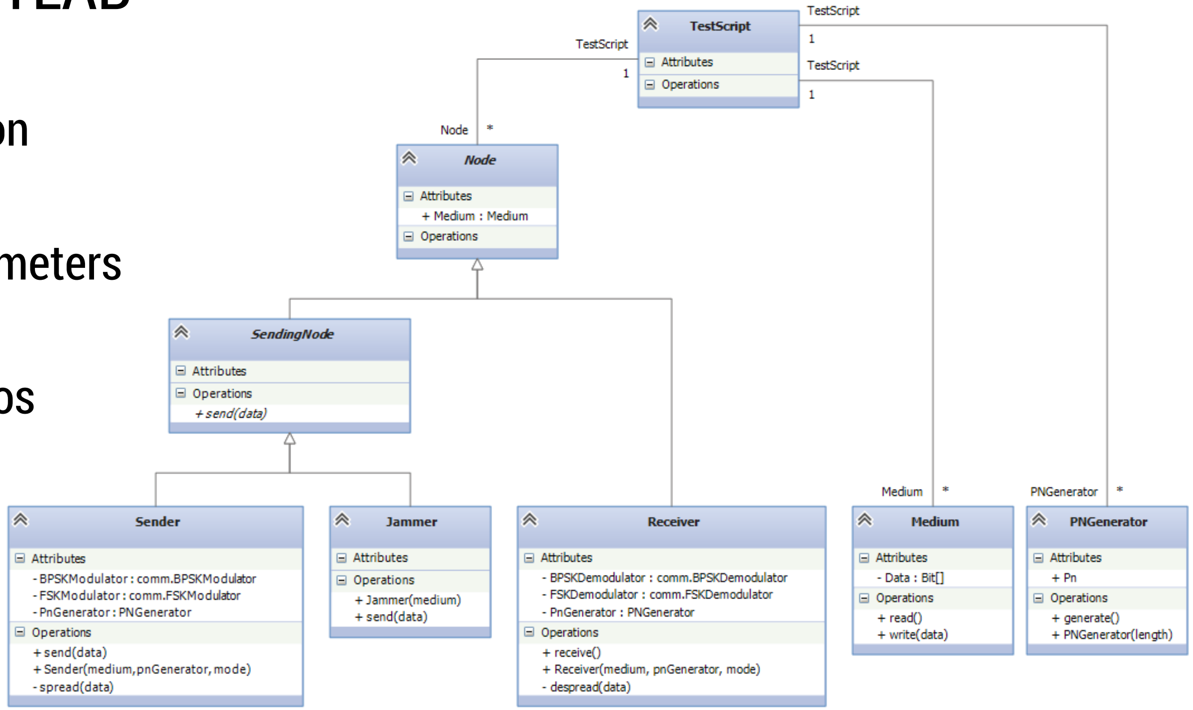
FHSS - Frequency Hopping Spread Spectrum

- Compute hopping sequence from chip sequence
- Switch channels each chip
    - Slow hopping: multiple symbols per chip
    - Fast hopping: multiple chips per symbol



Architecture

- Object Oriented MATLAB
- Regression Tests
    - check implementation
- Jamming Tests
    - check jamming parameters
- Test Script
    - execute test scenarios



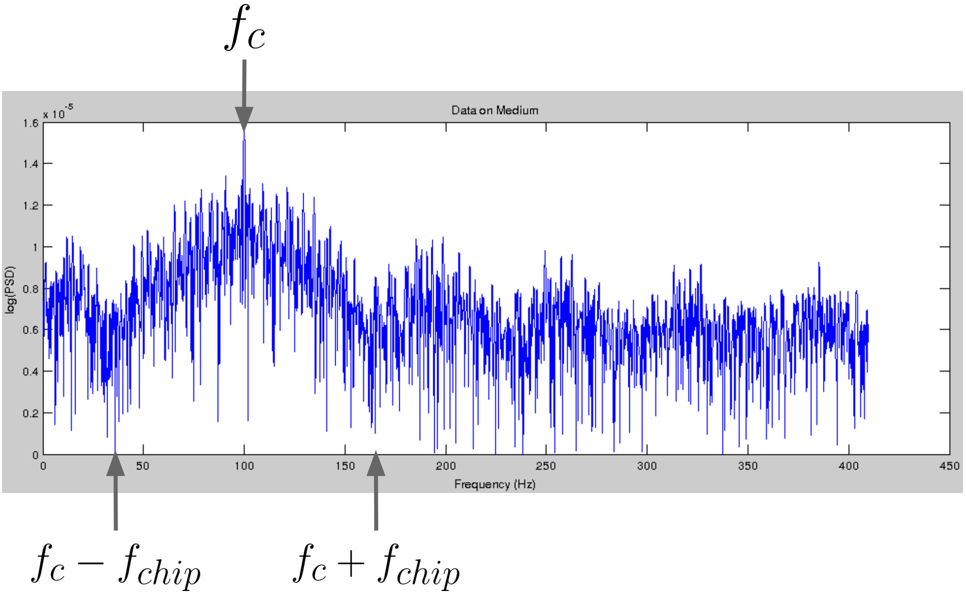
Architecture

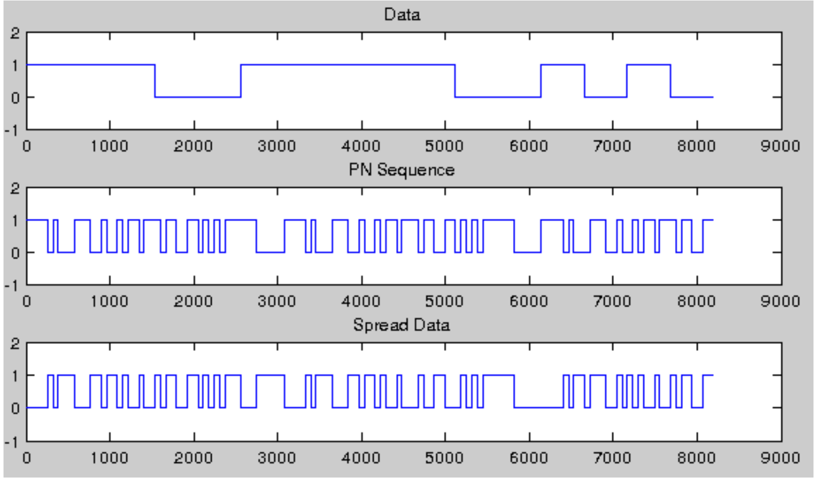
- Sender
  - sample data and pn-sequence
  - apply spreading (DSSS/FHSS)
  - apply BPSK modulation
- Jammer
  - jam data on medium
  - with specific bandwidth
  - with specific power
- Medium
  - perform FFT on sender write
  - handle interference
  - perform IFFT on receiver read
- Receiver
  - demodulate signal
  - apply despreading
  - reconstruct signal

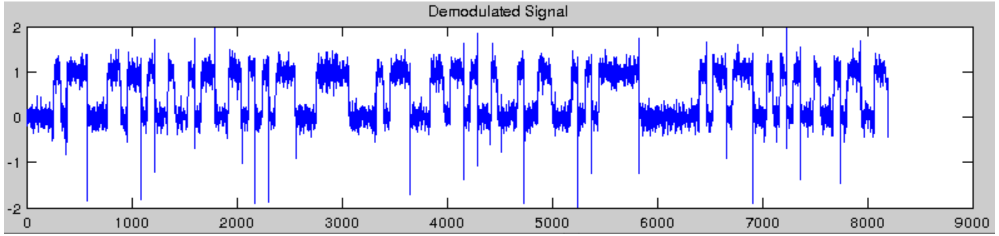
- We add AWGN with SNR of 10dB to simulate interference of various origins

Implementation DSSS

- Example
  - $f_c = 100\text{Hz}$
  - $f_{chip} = 64\text{Hz}$
  - $l_{pn} = 48$

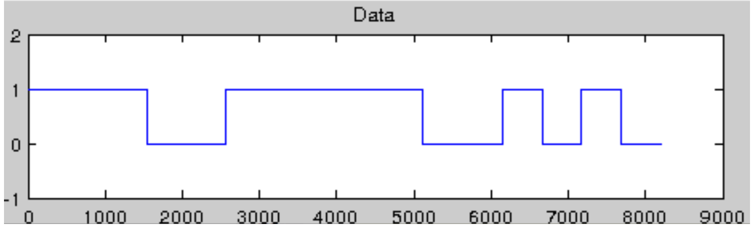


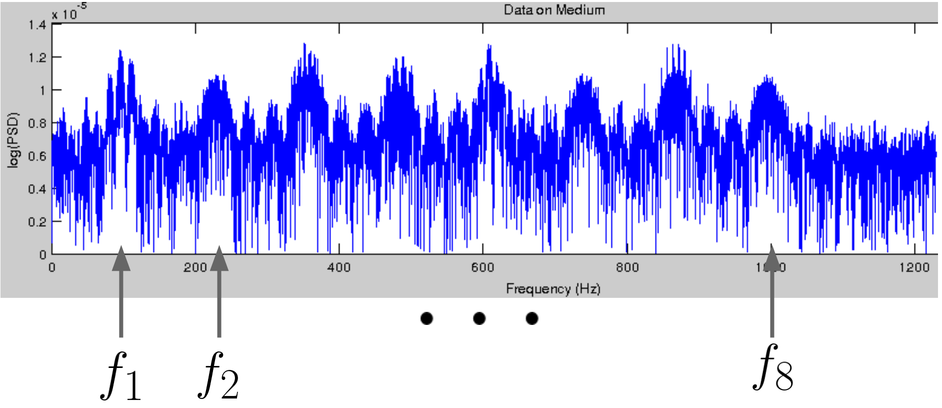




Implementation FHSS

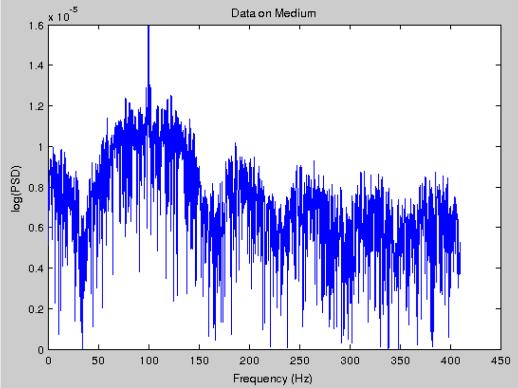
- Constant number of sub-carrier channels: 8
- Constant bandwidth per sub-carrier: 128 Hz
- Fast hopping performed worse than expected
- Example
    - $f_{sym} = 8\text{Hz}$
    - $f_{chip} = 32\text{Hz}$



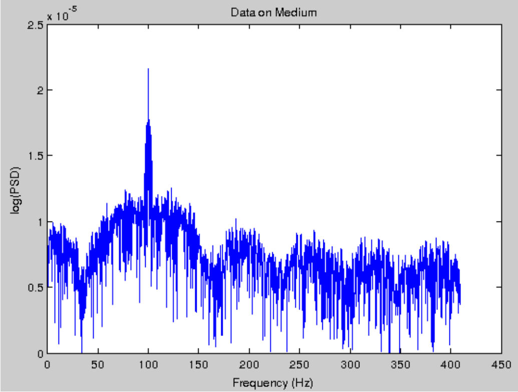


Implementation Jammer

- Tested various methods
- Time domain + bandpass filter
  - not reliable and hard to control
- Trequency domain + measure SNR
  - incorrect results, possibly flawed implementation
- DSSS user
  - control bandwidth using chipping rate
  - proved to be most reliable
  - power control still hard
- Not entirely satisfied with solution



not jammed



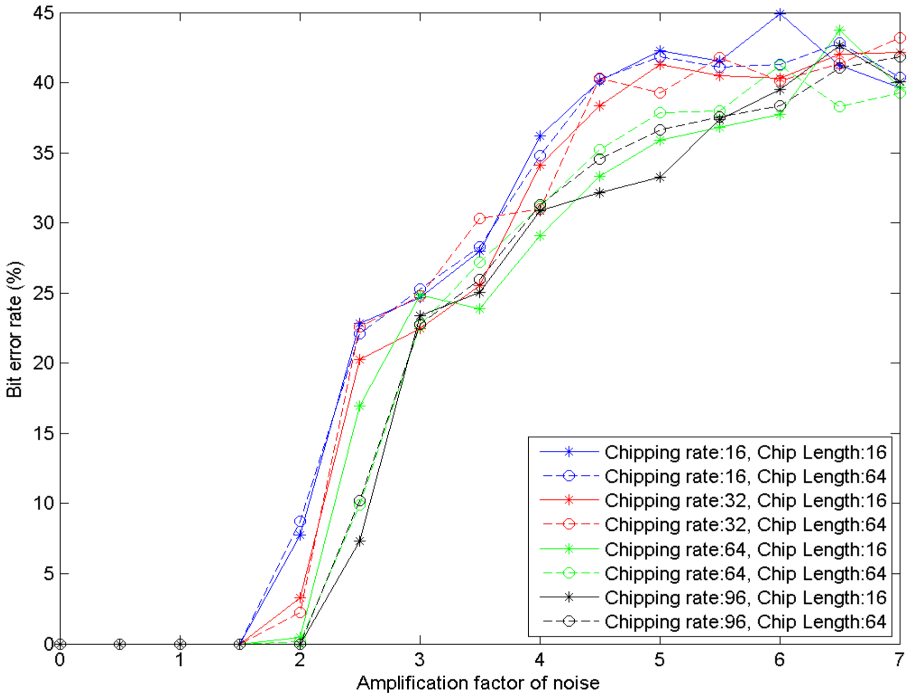
jammed, 8Hz, Power factor 1

Test Scenarios

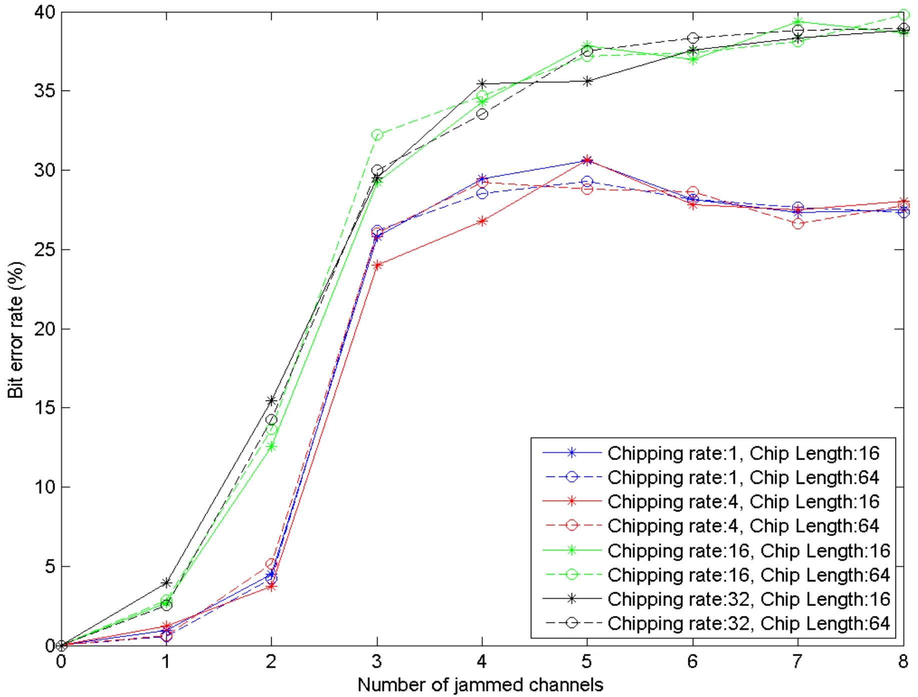
- Base Scenario: 1 Sender + 1 Receiver
  - Metric: BER with 4 different chipping rates and 2 pn-sequence lengths
- Narrowband jamming
  - both DSSS and FHSS should be robust
- Wideband jamming
  - both DSSS and FHSS should be susceptible
- Multiple users: 2-15 senders
  - DSSS: high chip rates should be better
- Different noise bandwidths
  - both robust for small bandwidth, error prone for large bandwidth
- Different levels of AWGN
  - expect both to be equal

Results - Narrowband Jamming

DSSS - 8 Hz

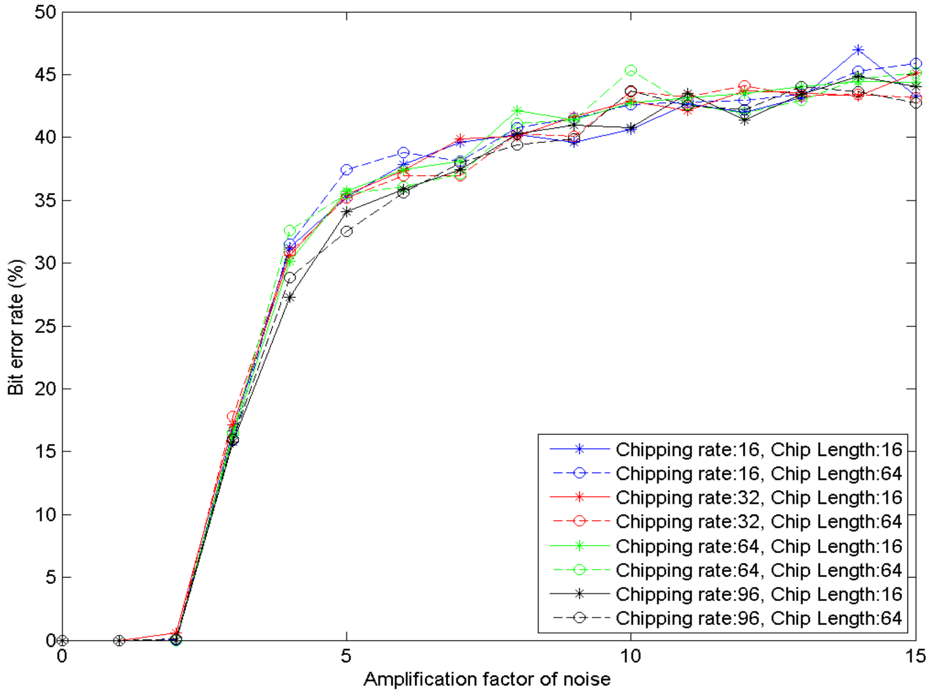


FHSS - 64 Hz

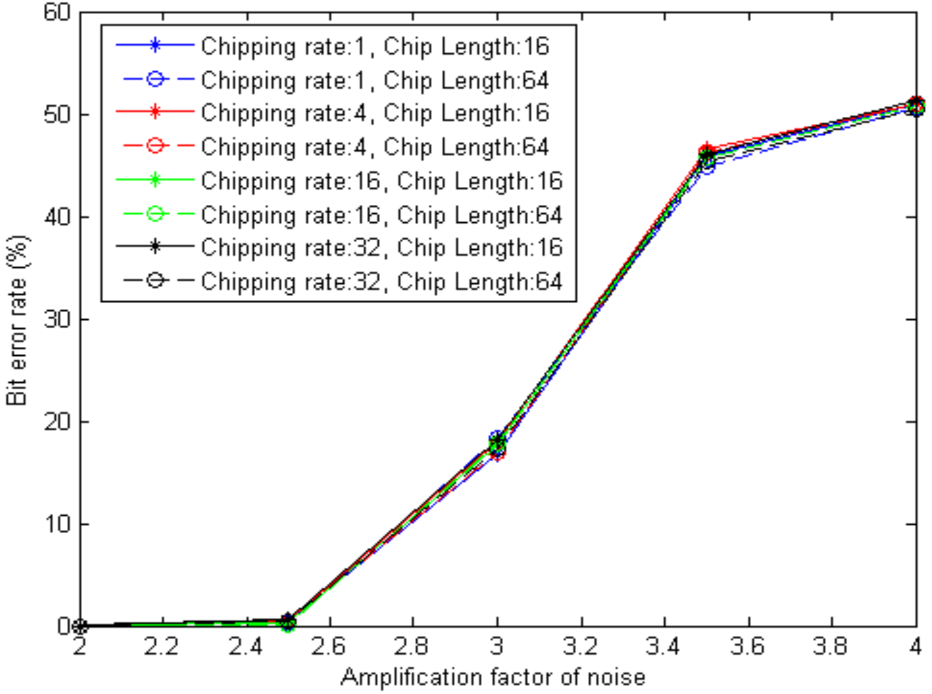


Results - Wideband Jamming

DSSS - 200 Hz

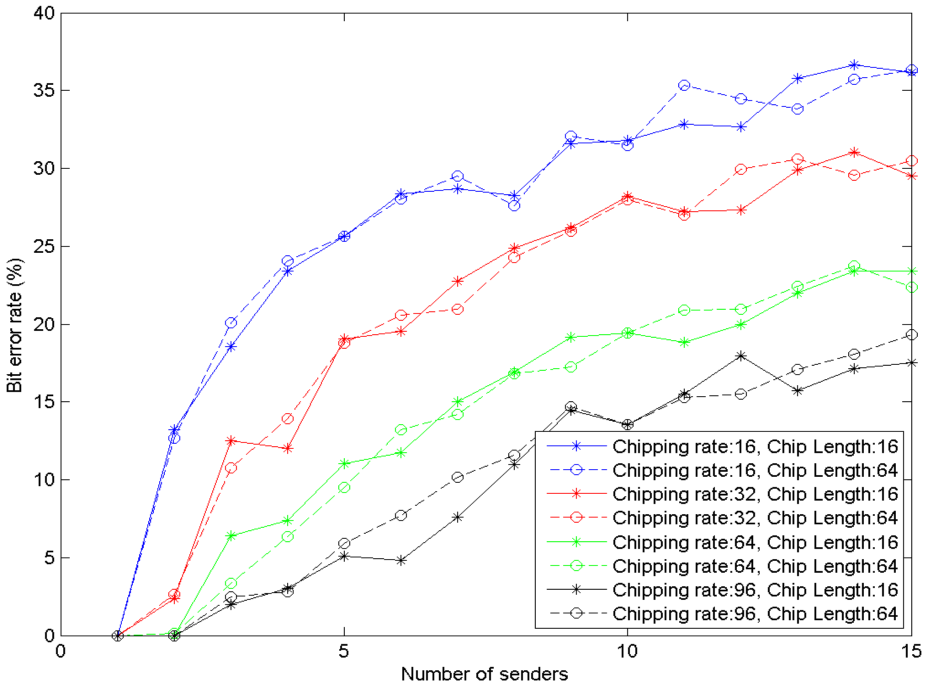


FHSS - 1000 Hz

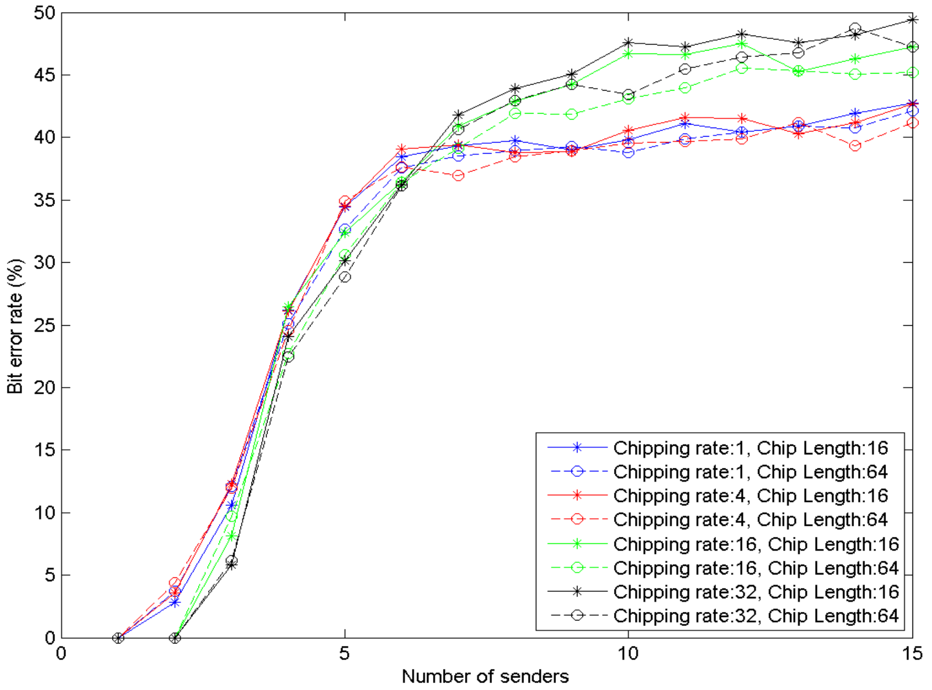


Results - Multiple Users

DSSS

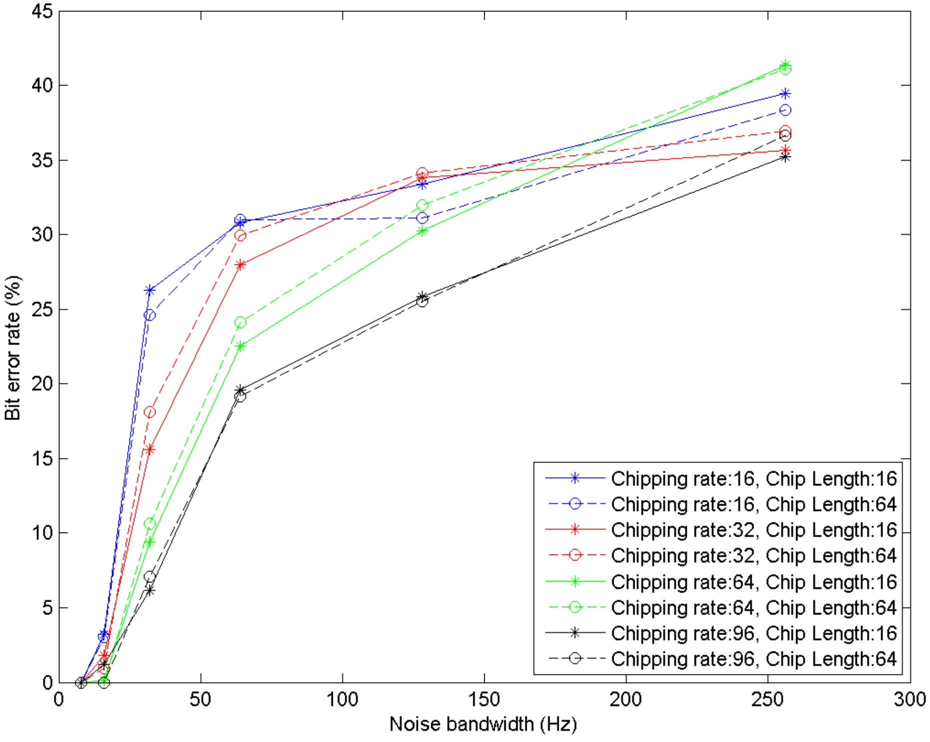


FHSS

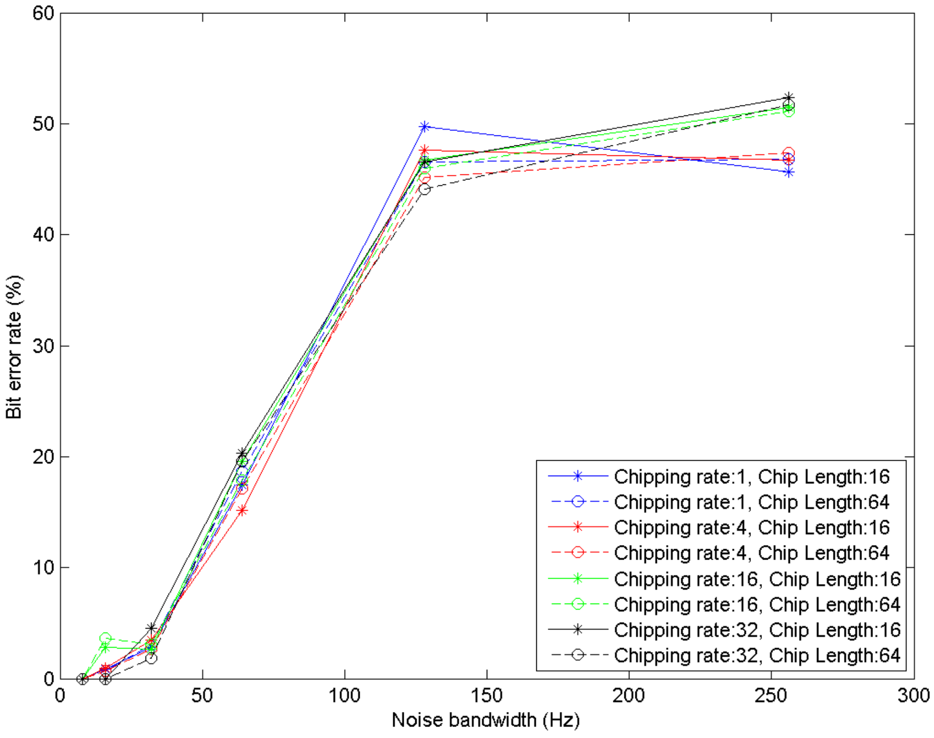


Results - Adaptive Bandwidth

DSSS

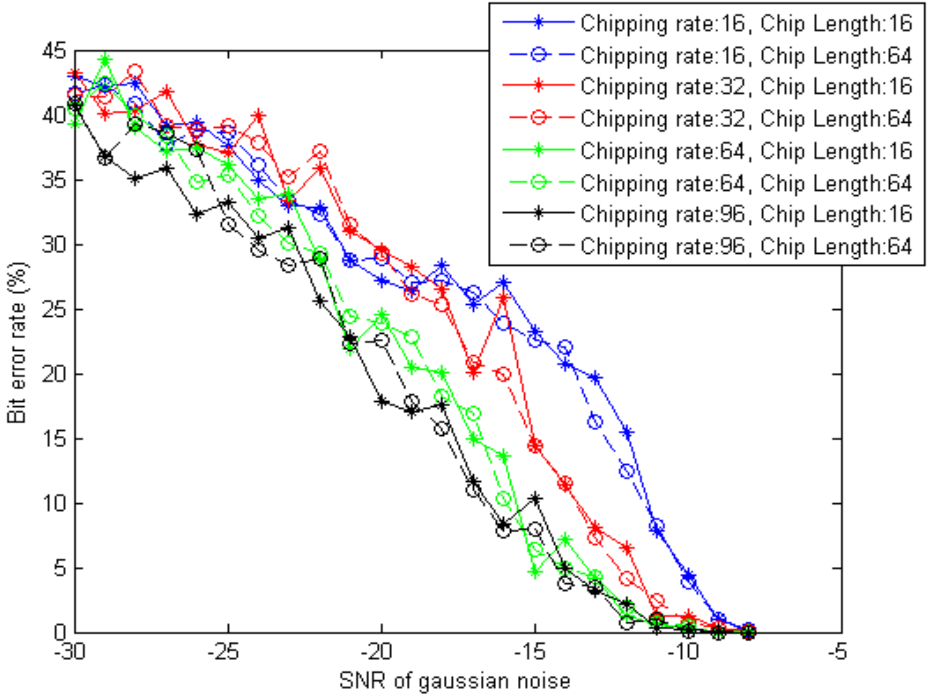


FHSS

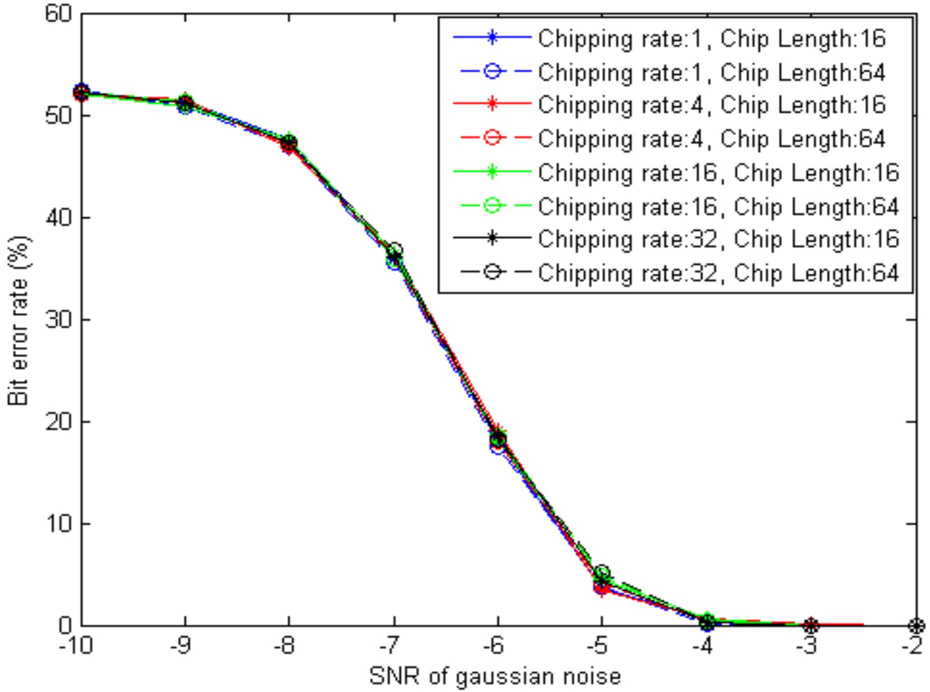


Results - White Gaussian Noise

DSSS



FHSS



Conclusion

- FHSS fast hopping: FSK instead of BPSK
- Jammer: Adaptive power
  - More background needed

- DSSS more efficient for multiple users
  - More robust to Gaussian noise
  - Performed better in our mixed test
- FHSS more robust towards narrowband
  - Probably more flexible

Questions ?