

160924 Downlink 2Hop Plots

September 25, 2016

```
In [ ]: %pylab inline
        from __future__ import division
        import scipy as sp
        from scipy import stats
        from scipy.signal import argrelextrema
        from scipy.interpolate import interp1d
        import operator as op
        from decimal import *
        # import mpmath as mp
        import pickle
        from scipy.stats import binom
        import mpld3
        mpld3.enable_notebook()

In [ ]: %run ../cow.py
        %run ../optimize_pkg_2.py

In [ ]: # codingscheme = filepath_down = "../data_tables/hs_rs_downlink"
        hsrs_up = "../data_tables/hs_rs_uplink"
        shannon_scheme = '../data_tables/shannon_tables'
        poly_scheme = '../data_tables/polyanskiy_up'

        protocolbit = 4*10**4
        target = 10**(-9)
        dSNR = 0.01
        paddratio = 0.1 # padd = 10**(-10)
        tSNR_range = np.arange(0, 92, 1)
        rSNR1_range = np.arange(-1, 7, 0.01)

        start_nodes, end_nodes = 2, 36
        user_range = xrange(start_node, end_node)
```

1 Loudest Talker

```
In [ ]: %%time
        start_SNR = 0

        hsrs_lt = loudest_talker(hsrs_up, dSNR, target, paddratio, start_SNR, start_nodes, end_nodes)
        print hsrs_lt
        print '\n'

        shannon_lt = loudest_talker(shannon_scheme, dSNR, target, paddratio, start_SNR, start_nodes, end_nodes)
        print shannon_lt
```

```
print '\n'
```

```
polyup_lt = loudest_talker(poly_scheme, dSNR, target, paddratio, start_SNR, start_nodes, end_no
print polyup_lt
```

2 2 Fade Gap

```
In [ ]: # hsrs_fadegap = np.array([ 35. , 13.59 , 4.91 10.91],
# [ 34. , 13.72 , 4.98 10.92],
# [ 33. , 13.84 , 4.98 10.92],
# [ 32. , 13.97 , 4.96 10.92],
# [ 31. , 14.9 , 3.99 10.93],
# [ 30. , 11.87 , 3.63 , 8.27],
# [ 29. , 12.45 , 2.99 , 8.29],
# [ 28. , 12.74 , 2.99 , 8.31],
# [ 27. , 11.13 , 2.94 , 6.87],
# [ 26. , 11.34 , 2.99 , 6.89],
# [ 25. , 11.57 , 2.98 , 6.91],
# [ 24. , 10.92 , 2.41 , 5.95],
# [ 23. , 11.52 , 1.99 , 5.97],
# [ 22. , 10.83 , 1.97 , 5.28],
# [ 21. , 11.15 , 1.99 , 5.31],
# [ 20. , 10.9 , 1.81 , 4.77],
# [ 19. , 11.28 , 1.94 , 4.83],
# [ 18. , 11.22 , 1.68 , 4.39],
# [ 17. , 11.94 , 0.99 , 4.05],
# [ 16. , 11.64 , 0.99 , 3.73],
# [ 15. , 11.94 , 1.43 , 3.8 ],
# [ 14. , 12.07 , 0.99 , 3.3 ],
# [ 13. , 12.43 , 0.99 , 3.11],
# [ 12. , 12.98 , 0.98 , 2.98],
# [ 11. , 13.48 , 0.87 , 2.68],
# [ 10. , 14.18 , 0.72 , 2.44],
# [ 9. , 14.97 , 0.53 , 2.11],
# [ 8. , 16.12 , 0.5 , 1.9 ],
# [ 7. , 17.5 , 0.28 , 1.55],
# [ 6.00000000e+00 , 1.98100000e+01 -1.00000000e-02 , 1.33000000e+00],
# [ 5.00000000e+00 , 2.23200000e+01 -1.00000000e-02 , 1.03000000e+00],
# [ 4. , 26.43 -0.07 , 0.71],
# [ 3. , 33.25 -0.25 , 0.34],
# [ 2. , 46.33 -0.38 -0.06])
```

```
In [ ]: %%time
hsrs_fadegap = np.zeros((end_nodes-start_nodes, 4))

endpoint = 2
# dfade = [10**(-3), 10**(-10), 10**(-3)]
dfade = 10**(-4)

hsrs_tSNR = {}
hsrs_rSNR = {}
for N in xrange(35, 1, -1):
    hsrs_tSNR[N] = 8
```

```

for N in xrange(35, 1, -1):
    filename = hsrs_up + '/n' + str(N) + '.in'
    downTable = load_table(filename)
    downfunc = interp1d(downTable[0], downTable[1], kind='linear', bounds_error=False, fill_val=0)

    for dSNR in [10**i for i in range(0, -3, -1)]:
        tSNR_range = np.arange(hsrs_tSNR[N], 92, dSNR)
        if N in hsrs_rSNR:
            rSNR1_range = np.arange(np.floor(hsrs_rSNR[N]), np.ceil(hsrs_rSNR[N]), 0.01)
        else:
            rSNR1_range = np.arange(-1, 5, 0.01)

        rSNR2 = downTable[0][np.where(np.array(downTable[1])<=0.1*protocol_target)[0][0]]
        pa2 = downTable[1][np.where(np.array(downTable[0]==rSNR2))[0][0]]

        hsrs_fadegap[N-start_nodes] = down_fade_gap_inner(N, downfunc, protocol_target, tSNR_range)
        hsrs_tSNR[N] = hsrs_fadegap[N-start_nodes][1] - dSNR
        hsrs_rSNR[N] = hsrs_fadegap[N-start_nodes][2]
        print hsrs_fadegap[N-start_nodes]

```

3 Integral

```

In [ ]: %%time
        %run ../cow.py

        endpoint = 2
        # dfade = [10**(-3), 10**(-10), 10**(-3)]
        dfade = 10**(-4)

        hsrs_lti = np.zeros((end_nodes-start_nodes,))

        hsrs_lti_dct = {}
        for N in xrange(35, 1, -1):
            hsrs_lti_dct[N] = 0
            for dSNR in [10**i for i in range(0, -3, -1)]:
                start_tSNR = hsrs_lti_dct[N]
                hsrs_lti[N-start_nodes] = loudest_talker_integral(N, hsrs_up, start_tSNR, dfade, dSNR)
                hsrs_lti_dct[N] = hsrs_lti[N-start_nodes] - dSNR
            print N, hsrs_lti[N-start_nodes]

In [ ]: %%time

        endpoint = 2
        # dfade = [10**(-3), 10**(-10), 10**(-3)]
        dfade = 10**(-4)

        poly_lti = np.zeros((end_nodes-start_nodes,))

        poly_lti_dct = {}
        for N in xrange(35, 1, -1):
            poly_lti_dct[N] = 0
            for dSNR in [10**i for i in range(0, -3, -1)]:
                start_tSNR = poly_lti_dct[N]
                poly_lti[N-start_nodes] = loudest_talker_integral(N, poly_scheme, start_tSNR, dfade, dSNR)

```

```

poly_lti_dct[N] = poly_lti[N-start_nodes] - dSNR
print N, poly_lti[N-start_nodes]

```

4 Plot

```

In [ ]: figure(figsize=(12, 8))
        plot(user_range, hsrs_lt, lw=2.0, label='Hamming Loudest Talker')
        plot(user_range, shannon_lt, lw=2.0, label='Shannon Loudest Talker')
        plot(user_range, polyup_lt, lw=2.0, label='Polyanskiy Loudest Talker')
        plot(user_range, hsrs_fadegap[:,1], lw=2.0, label='Hamming 2 Fade Gap')
        legend(loc=0)
        xlabel('Number of Nodes', fontsize=18)
        ylabel('Transmit SNR Needed (dB)', fontsize=18, labelpad=20)
        title('Downlink 2 Hop', fontsize=24)
        savefig('downlink_2hop.pdf', bbox='tight')

# figure(figsize=(12, 8))
# plot(user_range, hsrs_down-lt_fadegap_2[:,1], lw=2.0, label='Loudest Talker')
# # legend(loc=0)
# xlabel('Number of Nodes', fontsize=18)
# ylabel('Transmit SNR Needed (dB)', fontsize=18, labelpad=20)
# title('Loudest Talker Downlink Savings', fontsize=24)
# savefig('loudest_talker_savings_2.pdf', bbox='tight')

# figure(figsize=(12, 8))
# plot(user_range, lt_fadegap_2[:,3], lw=2.0, label='Loudest Talker')
# # plot(user_range, lt_fadegap_1[:,2], lw=2.0, label='Fade Gap No C')
# plot(user_range, lt_fadegap_2[:,2], lw=2.0, label='Fade Gap Conditioning')
# legend(loc=0)
# xlabel('Number of Nodes', fontsize=18)
# ylabel('Receiver SNR Needed (dB)', fontsize=18, labelpad=20)
# title('Loudest Talker Downlink rSNR-1', fontsize=24)
# savefig('loudest_talker_rSNR_2.pdf', bbox='tight')

In [ ]:

```