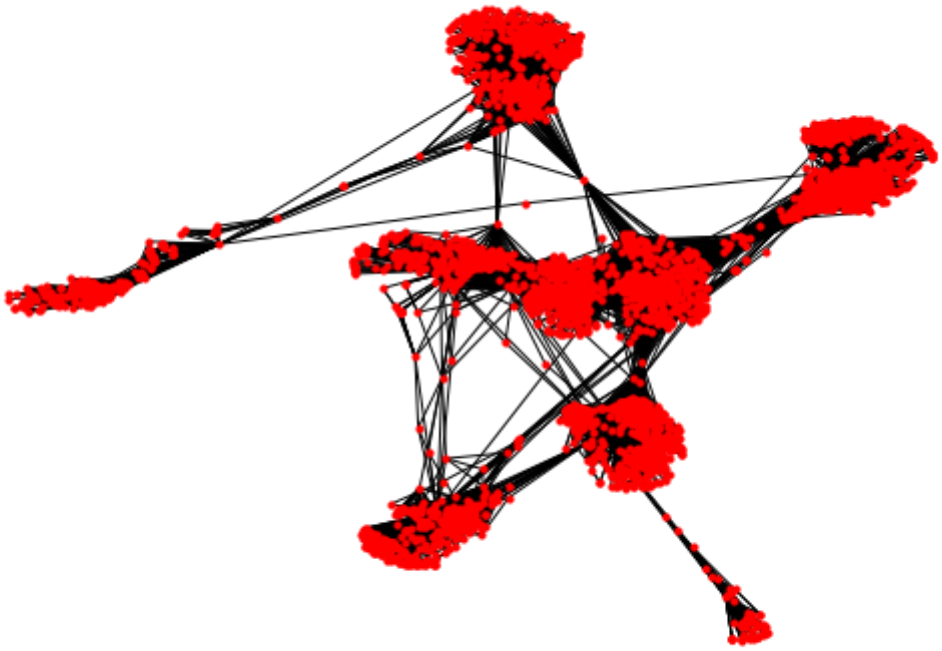
Assignment 4, Due Date 11.13.2017

In [1]:
```
import networkx as nx
import matplotlib.pyplot as plt
%matplotlib inline
fb = nx.read_edgelist('facebook.txt',create_using=nx.Graph(),nodetype=int)
print(nx.info(fb))
```

```
Name:
Type: Graph
Number of nodes: 4039
Number of edges: 88234
Average degree:  43.6910
```

In [2]:
```
plt.figure(figsize=(10,7))
plt.axis('off')
nx.draw_networkx(fb,with_labels=False,node_size=10)
```



In [11]:
```
# The diameter is the maximum eccentricity.
diam = nx.diameter(fb)
```

In [13]:
```
# The periphery is the set of nodes with eccentricity equal to the diameter.
periph = nx.periphery(fb)
```

```
[687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 699, 700, 701, 702, 70
4, 705, 706, 707, 709, 710, 711, 712, 714, 715, 716, 717, 718, 720, 721,
722, 723, 724, 725, 726, 727, 728, 730, 731, 732, 733, 734, 735, 736, 73
7, 738, 739, 740, 741, 742, 743, 744, 746, 748, 749, 750, 751, 752, 754,
755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 77
0, 771, 773, 775, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787,
788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 799, 801, 802, 806, 80
7, 808, 809, 812, 813, 814, 815, 816, 817, 818, 820, 821, 822, 824, 826,
827, 829, 831, 832, 833, 834, 835, 836, 837, 838, 839, 841, 842, 843, 84
4, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 3981, 3982, 398
3, 3984, 3985, 3986, 3987, 3988, 3990, 3991, 3992, 3993, 3994, 3995, 399
6, 3997, 3998, 3999, 4000, 4001, 4002, 4003, 4004, 4005, 4006, 4007, 400
8, 4009, 4010, 4012, 4013, 4014, 4015, 4016, 4017, 4018, 4019, 4020, 402
1, 4022, 4023, 4024, 4025, 4026, 4027, 4028, 4029, 4030, 4032, 4033, 403
4, 4035, 4036, 4037, 4038]
```

In [15]:
```
# The center is the set of nodes with eccentricity equal to radius.
center = nx.center(fb)
```

```
[567]
```

In [16]:
```
# The eccentricity of a node v is the maximum distance from v to all other n
eccentricity = nx.eccentricity(fb,567)
```

```
4
```

Question 1: Present the shortest path from the most popular node to the center node

In [78]:
```
mcenter = int(max(center))
dc = nx.degree_centrality(fb)
fbcenmax = max(zip(dc.values(), dc.keys()))[1]

print ("Most Popular:", max(zip(dc.values(), dc.keys()))[1])
print("Center:",mcenter)

path = nx.shortest_path(G,source=fbcenmax,target=mcenter)
print("Shortest Path:",path)
```

```
Most Popular: 107
Center: 567
Shortest Path: [107, 420, 567]
```

Question 2: Present the nodes connected (1st) to the center node

In [107]:
```
cenedges = nx.all_neighbors(fb, mcenter)
xx = list(cenedges)

#cenedges = nx.edges(fb,mcenter)
#xx = [x[1] for x in cenedges]
#print(xx)

print("Nodes Connected to Center Node:",xx)
print("Number of Nodes Connected to Center Node:",len(xx))
```

```
Nodes Connected to Center Node: [513, 387, 388, 645, 646, 391, 520, 395,
524, 525, 527, 400, 402, 3861, 537, 412, 542, 3487, 416, 417, 419, 420, 4
23, 553, 683, 428, 559, 392, 563, 414, 566, 439, 3723, 451, 580, 456, 65
1, 460, 461, 590, 591, 465, 561, 471, 472, 475, 348, 604, 353, 610, 483,
484, 360, 492, 3437, 497, 370, 373, 374, 503, 376, 3961, 3454]
Number of Nodes Connected to Center Node: 63
```

Question 3: Present the Eigenvector Centrality

In [96]:
```
ec = nx.eigenvector_centrality(fb)
#print (dc)
print ("Eigenvector Centrality;", max(zip(ec.values(), ec.keys())))
```

```
Eigenvector Centrality; (0.09540696149067635, 1912)
```

Question 4: Present the closeness centrality

In [97]:
```
cc = nx.closeness_centrality(fb)
#print (dc)
print ("Betweenness Centrality;", max(zip(cc.values(), cc.keys())))
```

```
Betweenness Centrality; (0.45969945355191255, 107)
```

Question 5: Present the betweenness centrality

In [98]:
```
bc = nx.betweenness_centrality(fb)
#print (dc)
print ("Betweenness Centrality;", max(zip(bc.values(), bc.keys())))
```

```
Betweenness Centrality; (0.4805180785560147, 107)
```