网络协议分析作业 4
陈浚铭： Q1 到 Q4
杨锦程： Q5

Q1. 我们分别有两个 pcapng 文件， 一个总共有 100 个流(100flow.pcapng)，一个总共有 54 个流(54flow.pcapng)，然后通过使用 readFlow.py， 读取流数据到 flow_list.txt。注意我们使用了上一章的看视频的数据，当中使用的传输协议是 TCP， 因此，我们只是获取了流为 srcIP, dstIP, srcPort, dstPort, TCP。
Q2. 我们设计的哈希函数是使用了基于 Merkel Damagard construction 的哈希函数， 也就是 sha1, sha2, md5 。(对于 bloom filter 我们使用这三个函数，而 direct bitmap 只是使用 sha1, sha2, md5)
使用这类型的哈希函数 h, 我们就可以对于流为输入进行定义:
myHashFunction(srcIP, dstIP, srcPort, dstPort)= hash_function(srcIP||dstIP||srcPort||dstPort||“TCP”),
（当中 hash_function 可能为 sha1, sha2, md5 或者 c++ stl 的 hash function）
通过使用 sha1 函数,
Q3. 我们通过不同的流个数(54 和 100），direct bitmap 长度 L(16 bit，256 bit, 4096 bit），哈希结果长度（160 和 256）的关系, 讨论 L 和哈希结果长度对于碰撞情况的影响。

实现的代码： directBitmap.cpp
使用 sha1(输出结果长度为 160bit)得到碰撞次数

流个数 \ Bitmap 长度	16 bit	256 bit	4096 bit
54	51	11	5
100	97	26	4

从上图可见，碰撞的机会和 Bitmap 长度成反比，和流的数量成正比。
使用 sha2(输出结果长度为 256bit)得到碰撞次

流个数 \ Bitmap 长度	16 bit	256 bit	4096 bit
54	46	5	1
100	96	30	2

我们发现对于 sha2 哈希函数得出的结果与 sha1 哈希函数的结果相似。另外比较这两个哈希输出长度所得出的不同结果，我们发现， sha2 的碰撞机会并不跟 sha1 的结果有太大的差别，这可能是因为 merkle damagard construction 的哈希函数，这种构造方法的作用在于减小出现碰撞的机会，因此哈希值的分布比较均衡。

Q4. 根据讲义，如果 Bloom filter 的储存结构有 m 位， 流个数为 n, 使用的不同哈希函数的个数为 k，那么 Bloom filter 中任意一位为 “0”的概率为 p’ = (1 – 1/m)^(nk) ≈ e^(-kn/m) = p
因此插入 bloom filter 时候，误判律为 f(m, k, n) = (1- p^r)^k ≈ (1- e^(-kn/m))^k
我们尝试通过不同的 m k n 的值来判断结果，当中注意到，我们只能对于判断定义中的阳性是否误判，因为我们能够通过流数据集来判断是不是有重叠（也就是出现碰撞）的储存，使得结果是出现 “假阳性”的结果，则误判的结果。

实现的代码： bloomFilter.cpp
对于 k = 2 (h1 = sha1, h2 = sha2)

流个数 \ Bloom Filter 位数	16 bit	256 bit	4096 bit
54	50	16	0
100	96	54	2

从上图可以见， 类似于 direct bitmap 的结果，碰撞的机会，而因此误判律和 Bloom Filter 的大小成反比，和流的数量成正比。

对于 k = 3:

流个数 \ Bloom Filter 位数	16 bit	256 bit	4096 bit
54	50	21	0
100	96	52	0

我们注意到， k = 3 的碰撞机会比较小， 因此误判律也比较小，这是与公式符合的, 也就是 k 越大， 误判的机会越小。

Q5:

实现的代码: sketch.py

Wireshark 文件： 100flow.pcap

使用 count min sketch 获取流的结果截图：第三列为每个流的计数

第四列为字节长度，第五列为估计的负载量（按三分钟的每秒负载量计算

（count min sketch 宽度为 10 深度为 5）

	name	Math_A	len	load
0	{'Protocol'	141	2775	15.41667
1	{'Protocol'	124	13135	72.97222
2	{'Protocol'	79	1665	9.25
3	{'Protocol'	101	2220	12.33333
4	{'Protocol'	64	4625	25.69444
5	{'Protocol'	124	11840	65.77778
6	{'Protocol'	117	2035	11.30556
7	{'Protocol'	67	4995	27.75
8	{'Protocol'	79	2035	11.30556
9	{'Protocol'	118	2035	11.30556
10	{'Protocol'	143	1665	9.25
11	{'Protocol'	81	2035	11.30556
12	{'Protocol'	49	2405	13.36111
13	{'Protocol'	126	2590	14.38889
14	{'Protocol'	45	370	2.055556
15	{'Protocol'	197	9065	50.36111
16	{'Protocol'	100	9250	51.38889
17	{'Protocol'	105	370	2.055556
18	{'Protocol'	112	11840	65.77778
19	{'Protocol'	268	35705	198.3611
20	{'Protocol'	101	3330	18.5
21	{'Protocol'	81	3330	18.5
22	{'Protocol'	64	370	2.055556
23	{'Protocol'	113	7030	39.05556
24	{'Protocol'	143	5735	31.86111
25	{'Protocol'	81	1295	7.194444
26	{'Protocol'	45	1295	7.194444
27	{'Protocol'	96	1295	7.194444
28	{'Protocol'	118	1110	6.166667
29	{'Protocol'	143	3330	18.5
30	{'Protocol'	81	3330	18.5
31	{'Protocol'	2649	482110	2678.389
32	{'Protocol'	5278	969770	5387.611

改变宽度和深度的对比

减小宽度(宽度为 8):总体计数变多

	name	Math_A	len	load
0	{'Protocol'	113	2775	15.41667
1	{'Protocol'	125	13135	72.97222
2	{'Protocol'	113	1665	9.25
3	{'Protocol'	71	2220	12.33333
4	{'Protocol'	112	4625	25.69444
5	{'Protocol'	128	11840	65.77778
6	{'Protocol'	112	2035	11.30556
7	{'Protocol'	117	4995	27.75
8	{'Protocol'	113	2035	11.30556
9	{'Protocol'	112	2035	11.30556
10	{'Protocol'	115	1665	9.25
11	{'Protocol'	123	2035	11.30556
12	{'Protocol'	300	2405	13.36111
13	{'Protocol'	144	2590	14.38889
14	{'Protocol'	114	370	2.055556
15	{'Protocol'	154	9065	50.36111
16	{'Protocol'	132	9250	51.38889
17	{'Protocol'	143	370	2.055556
18	{'Protocol'	241	11840	65.77778
19	{'Protocol'	273	35705	198.3611
20	{'Protocol'	123	3330	18.5
21	{'Protocol'	113	3330	18.5
22	{'Protocol'	71	370	2.055556
23	{'Protocol'	186	7030	39.05556
24	{'Protocol'	71	5735	31.86111
25	{'Protocol'	114	1295	7.194444
26	{'Protocol'	114	1295	7.194444
27	{'Protocol'	114	1295	7.194444
28	{'Protocol'	113	1110	6.166667
29	{'Protocol'	117	3330	18.5
30	{'Protocol'	143	3330	18.5
31	{'Protocol'	2684	482110	2678.389
32	{'Protocol'	5311	969770	5387.611

减小深度(深度为 4)：可以看到部分数据没有变化，而一些数据的计数明显增大

	name	Math_A	len	load
0	{'Protocol'	179	2775	15.41667
1	{'Protocol'	124	13135	72.97222
2	{'Protocol'	79	1665	9.25
3	{'Protocol'	101	2220	12.33333
4	{'Protocol'	64	4625	25.69444
5	{'Protocol'	124	11840	65.77778
6	{'Protocol'	277	2035	11.30556
7	{'Protocol'	67	4995	27.75
8	{'Protocol'	79	2035	11.30556
9	{'Protocol'	124	2035	11.30556
10	{'Protocol'	143	1665	9.25
11	{'Protocol'	81	2035	11.30556
12	{'Protocol'	49	2405	13.36111
13	{'Protocol'	126	2590	14.38889
14	{'Protocol'	45	370	2.055556
15	{'Protocol'	350	9065	50.36111
16	{'Protocol'	100	9250	51.38889
17	{'Protocol'	105	370	2.055556
18	{'Protocol'	112	11840	65.77778
19	{'Protocol'	268	35705	198.3611
20	{'Protocol'	101	3330	18.5
21	{'Protocol'	81	3330	18.5
22	{'Protocol'	64	370	2.055556
23	{'Protocol'	113	7030	39.05556
24	{'Protocol'	143	5735	31.86111
25	{'Protocol'	81	1295	7.194444
26	{'Protocol'	45	1295	7.194444
27	{'Protocol'	96	1295	7.194444
28	{'Protocol'	124	1110	6.166667
29	{'Protocol'	143	3330	18.5
30	{'Protocol'	81	3330	18.5
31	{'Protocol'	2656	482110	2678.389
32	{'Protocol'	5311	969770	5387.611

总结变化规律：

深度为哈希函数的数量，会影响到哈希结果的最小值，如果较小可能使结果偏大，深度越大就更精确（对低频影响较小）过大则对结果的影响较小，宽度为哈希数组，宽度越大越不容易碰撞

同理，流数会影响的结果的准确率，流数越多结果越容易产生碰撞，结果偏大