CS 188

P2P Systems
Winter 2010
MIDTERM

Professor Giovanni Pau
Tuesday, February 9, 2010

This test is closed book

Estimated time: 1h and 50 minutes.

Name _____

Student ID _____

Do **NOT** turn to the next page until told to do so.

Exercise	Value	Score
1	30	
2	30	
3	30	
4	30	
TOTAL	120	

Intentionally empty.

1) Answer the following True/False (5 min)

1. Kademlia uses the XOR to decide the distance between two nodes in the routing space **TRUE**
2. Napster central server stores the files to be exchanged. **FALSE**
3. The MD5 hashing algorithm can be used to store a <key,vaue> record in a DHT. **TRUE**
4. A DHT can be used as distributed index to build a p2p storage system. **TRUE**
5. Gnutella networks use the Ping-Pong protocol to perform content searching. **FALSE**
6. The Random Walk protocol is used by Skype to locate users on the network. **FALSE**
7. Napster is a fully distributed system and very hard to shutdown. **FALSE**
8. Gnutella trades efficiency for potential failure in performing search. **TRUE**
9. The buffer for real-time streaming applications is used to compensate changes in network performance over time. **TRUE**
10. Chord neighbors are sufficient to perform routing. **TRUE**
11. In Skype Super Peers are used for audio packet forwarding in each call. **FALSE**
12. Chord is more efficient of Kademlia in performing routing (efficiency is computed as number of hops) **FALSE**
13. The failure of 1 single entry in the pastry leafset compromises the pastry routing and partitions the network. **FALSE**
14. Pastry neighborhood set essential to perform routing. **FALSE**

2) Multiple choice (**correct answer can be 1, 2, all or none**)

1. Consider a Chord Ring in the address space 0..512, the routing table will contain **up to:**

   ~~2 Fingers~~ **8 Fingers** ~~0 Fingers~~ ~~4 Fingers~~

   ~~None of the Above~~

2. Consider a Chord Ring with 1,000,000 nodes, the a**average** number of hops to route a node is:

   ~~2 hops~~ ~~8 hops~~ ~~7 hops~~ ~~4 hops~~

   None of the Above

3. The BitTorrent tracker:

   Returns as first a random list of nodes
   ~~Returns as first the nodes that contain the rarest chunk~~
   ~~Forces a node to connect to a specific peer~~

4. Pastry Routing table for node **A:**

   ~~In row k contains the IPs of all nodes that share K bit with A~~
   In row k contains IPs for first 16 nodes sharing K bits with A
   ~~In row k contains the IPs of all nodes that share K-1 bit with A~~
   ~~In row k contains IPs of Log16(K) nodes that share K bit with A~~
   ~~None of the above~~

5. Kadmelia Buckets:
   ~~Contain exactly 1 record~~
   ~~Contain a number of records proportional to the network size~~
   Contain a pre-defined number of records
   ~~None of the above~~

6. In a three based P2P streaming system:

   ~~Nodes store the content associated to the hash of a Key~~
   ~~Nodes store a <key,value> in closest ID to a given Key~~
   ~~Nodes store a <key,value> and related content in the same place~~
   Nodes store data chunks in temporary buffer for forwarding
   ~~None of the above~~

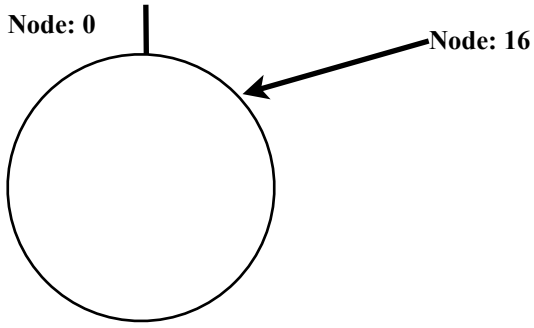
7. A tree based streaming system:

   Needs to be designed as a deep tree to minimize the contribution by each single node
   ~~Needs to be designed as a deep tree to minimize the delay at the leafs~~
   ~~None of the Above~~
   ~~=~~

1. **CHORD Routing**

Consider the following ring space 0..512 in Cord. :

The following nodes are in the ring: 0, 2, 4, 8, 14, 16, 20, 22, 32, 64, 127, 128, 224, 255



Write the routing table for NODE. 16

Range	Index	Node
[17,18)	0	20
[18,20)	1	20
[20,24)	2	20
[24,32)	3	32
[32,48)	4	32
[48,80)	5	64
[80,144)	6	127
[144,16)	7	224

2. **Pastry Routing**

Consider the following ring space 0..255 in Pastry. :

The following nodes are in the ring: 0, 2, 4, 8, 14, 16, 20, 21, 22,23, 24, 32, 33, 34, 35, 64, 127, 128, 224, 255
d
A) For node **16 write down the pastry leafset** assuming there are a total of 16 entries. Point out which nodes are part of the **lower** leafset and which one are part of the **higher** leafset.

Lower	14	8	4	2	0	255	224	128
Upper	20	21	22	23	24	32	33	34

B) For the same ring as in point A write down for node 16 the row 0 of the pastry routing table (TIP: Think in Binary)

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	0x00	0x14	0x20		0x40			0x7f	0x80						0xed	0xff

Dec	Hex
0	00
2	02
4	04
8	08
14	0E
16	10
20	14
21	15

Dec	Hex
22	16
23	17
24	18
32	20
33	21
34	22
35	23
64	40

Dec	Hex
127	7F
128	80
224	E0
255	FF

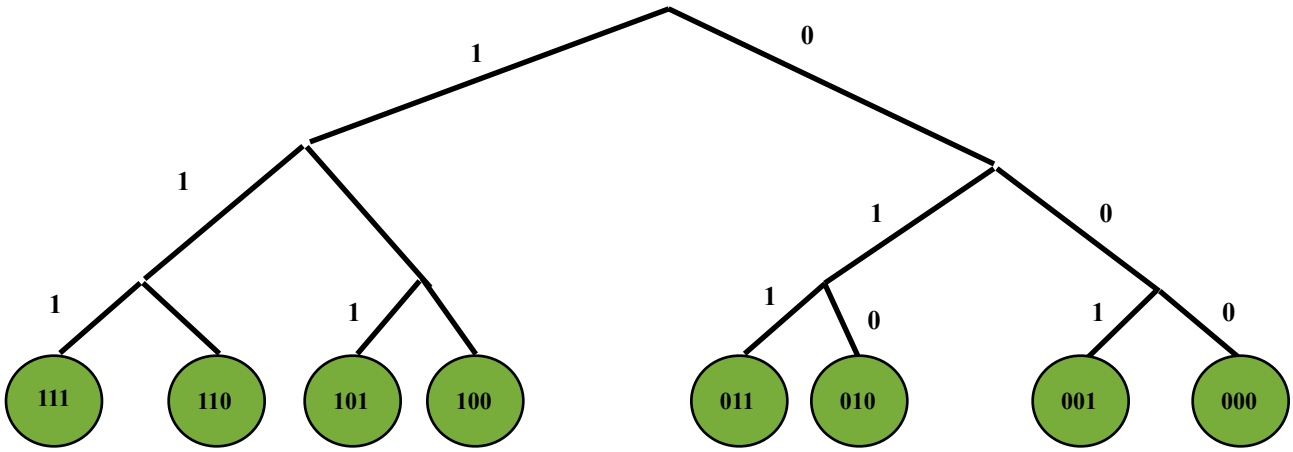
Dec	Bin
0	00000000
2	00000010
4	00000100
8	00001000
14	00001110
16	00010000
20	00010100
21	00010101

Dec	Hex
22	00010110
23	00010111
24	00011000
32	00100000
33	00100001
34	00100010
35	00100011
64	01000000

Dec	Hex
127	01111111
128	10000000
224	11100000
255	11111111

3. Consider a Kademlia network with 2^3 nodes please write down the follwoing:

A) The network topology assuming all the nodes are present.



B) Indicate the K-bukets and the nodes that fall in each K-buckets considering a K bucket of size 2.

Seen from Node **000**

Distance	K-Bucket
[0,1]	**1**
[2,3]	**010, 011**
[4-7]	**100, 101**

C) What would be the situation with a K-bucket size 3?

Seen from Node **000**

Distance	K-Bucket
[0,1]	**1**
[2,3]	**010, 011**
[4-7]	**100, 101,110**

4. Consider a tree with a depth of 4 levels and a fanout of 3 nodes per level fully populated. The average delay to cross a level is 400msec:

A) what is the delay for streaming a a video at Level 4?
B) and at Level 2?

LEVEL 4: 4x400=1600ms
LEVEL 2: 2x400=800ms

C) if the stream is 300Kbit/sec what is the minimum buffer size needed at each level to avoid interruption in a static ideal condition? (**i.e. the network does not change performance)**

There is no Buffer needed 0 is the size due NO Jitter!!!

D) With a 300Kbit/s streaming, Can a node with connected to a DSL performing 1024/768 join this tree at the leaf level?, Why?

YES--> A leaf does NOT upload ANYTHING

E) With a 300Kbit/s streaming, Can a node with a DSL 1024/768 join the tree at Level 1. (Root is considered level 0). Why?

NO---> Recall that the Fanout of this Tree is 3 so if we have 300Kbit/s would be 300*3=900Kbps >> 768Kbps