GSCI 1380: Day 9 (Tapestry Continued)

Last class

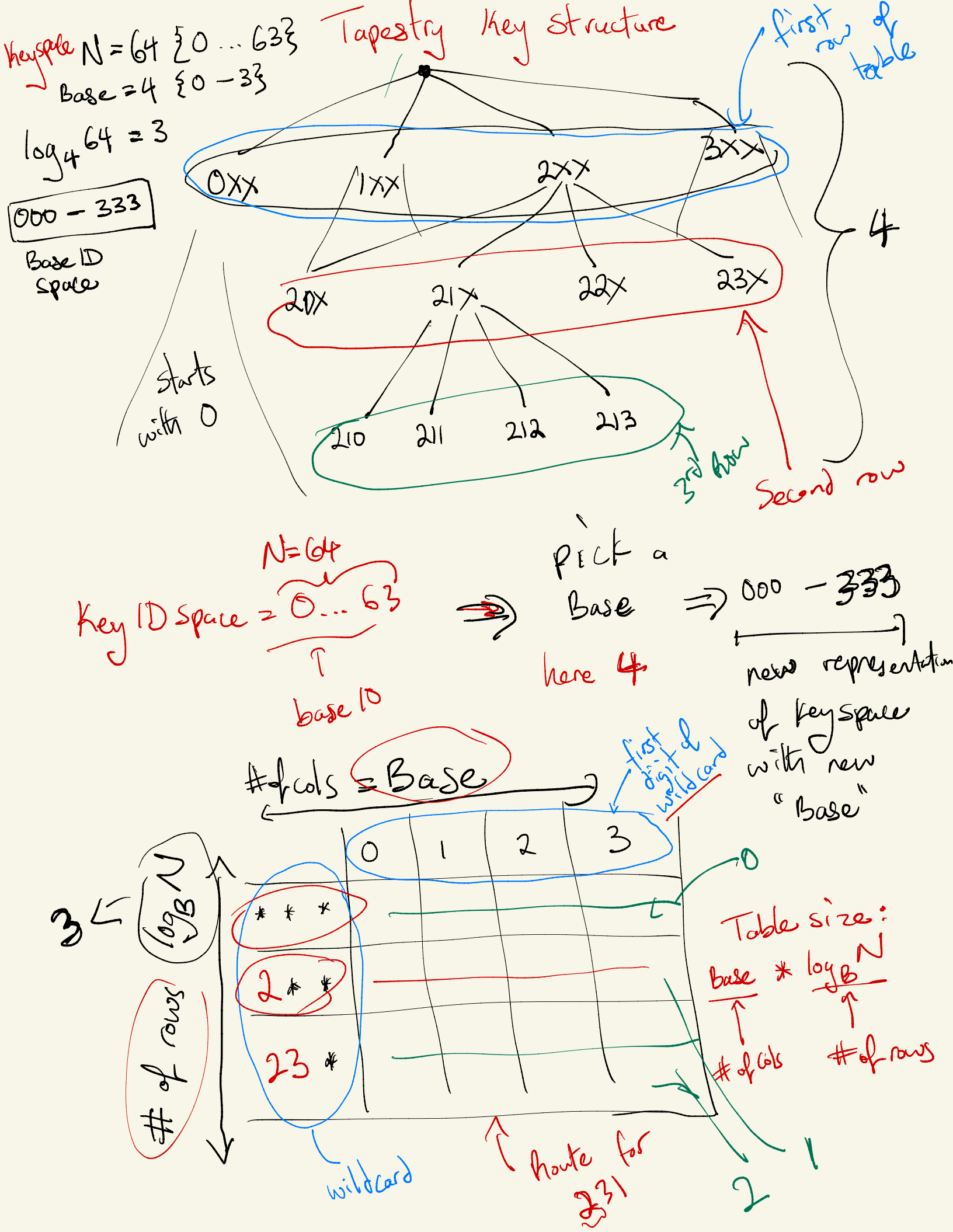
* Consistent hash recap
  - load imbalance ⇒ Virtual nodes
  - route table size (# of servers)
* Scaling consistent hash
  - Chord (Table size = $\log_2(N)$)
  - Tapestry (Table size = $\log_B(N) * B$)

MaxHop

This Class

* Tapestry In Detail
  1. Routing Table Creation (closest node)
  2. Key lookup (Root Node, Better Choice, MAXHOP) — fig #3 in paper
  3. Overview (back pointers, object store, Publishing)
  4. Deleting Nodes (Graceful/planned V. unGraceful/unplanned)
  5. Adding Nodes (AddNode Multicast, Need To know Nodes)

Node = Server ← Host


Tapestry key structure
keyspace N = 64 {0 ... 63}
Base = 4 {0 - 3}
$\log_4 64 = 3$
000 - 333
Base ID space
first row of table
0XX
1XX
2XX
3XX
20X
21X
22X
23X
210
211
212
213
4
starts with 0
3rd row
Second row
Key ID Space = 0 ... 63
N=64
base 10
Pick a Base
here 4
000 - 333
new representation of keyspace with new "Base"
#of cols = Base
first digit of wild card
0
1
2
3
* * *
2 * *
23 *
$3 \leftarrow \log_B N$
# of rows
wildcard
route for 231
0
1
2
Table size:
Base * $\log_B N$
#of cols
#of rows

Node IDs : 011, 211, 222, 231, 201, 300, 333

211

	0	1	2	3
0 * *	011	—	211	300
2 * *	201	211	222	231
21 *	~~210~~	211	~~212~~	~~213~~

011

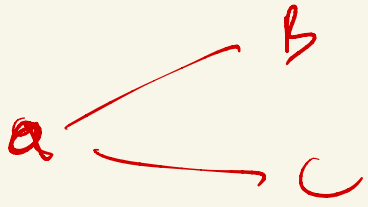
	0	1	2	3
* * *	011	—	222	333
0 * *	~~000~~	01*	~~210~~	~~231~~
01 *	~~010~~	011	~~212~~	~~213~~

Options for closeness

1. random
2. no hop (minimize latency)
3. min(|A − B|) = absolute distance / difference in IDs
   - project

	0	1	2	3
* * *				
* *				
*				

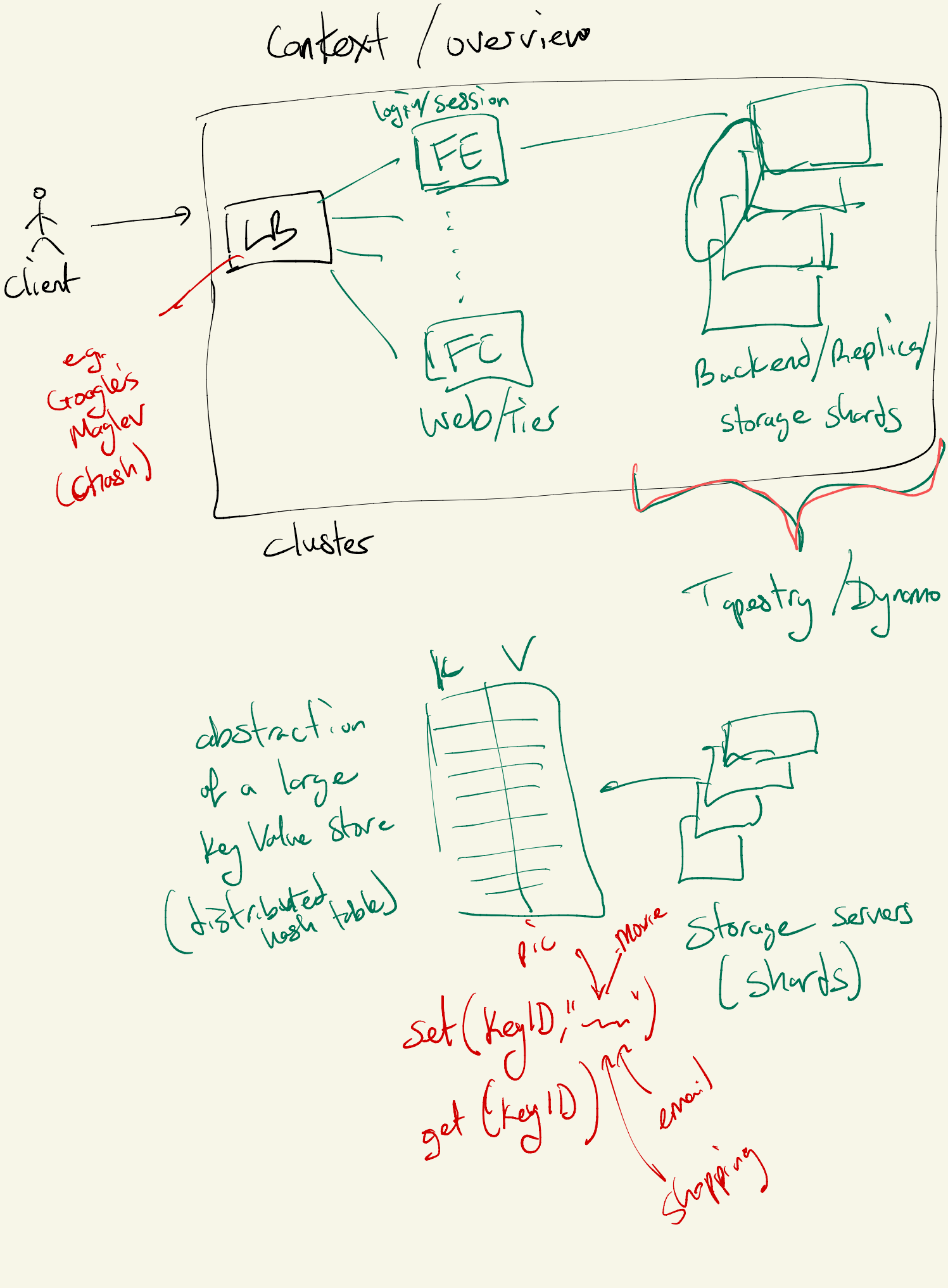
closeness



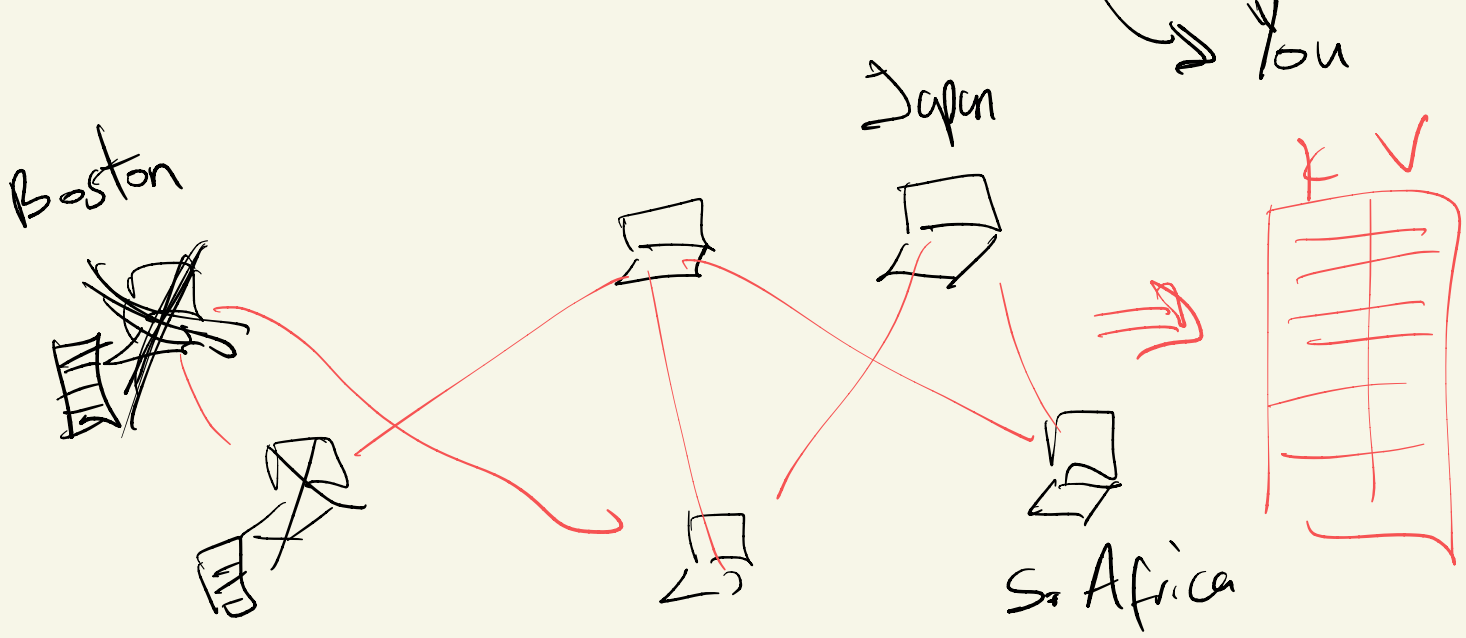
min (|a − B|, |a − c|)

Points about table creation

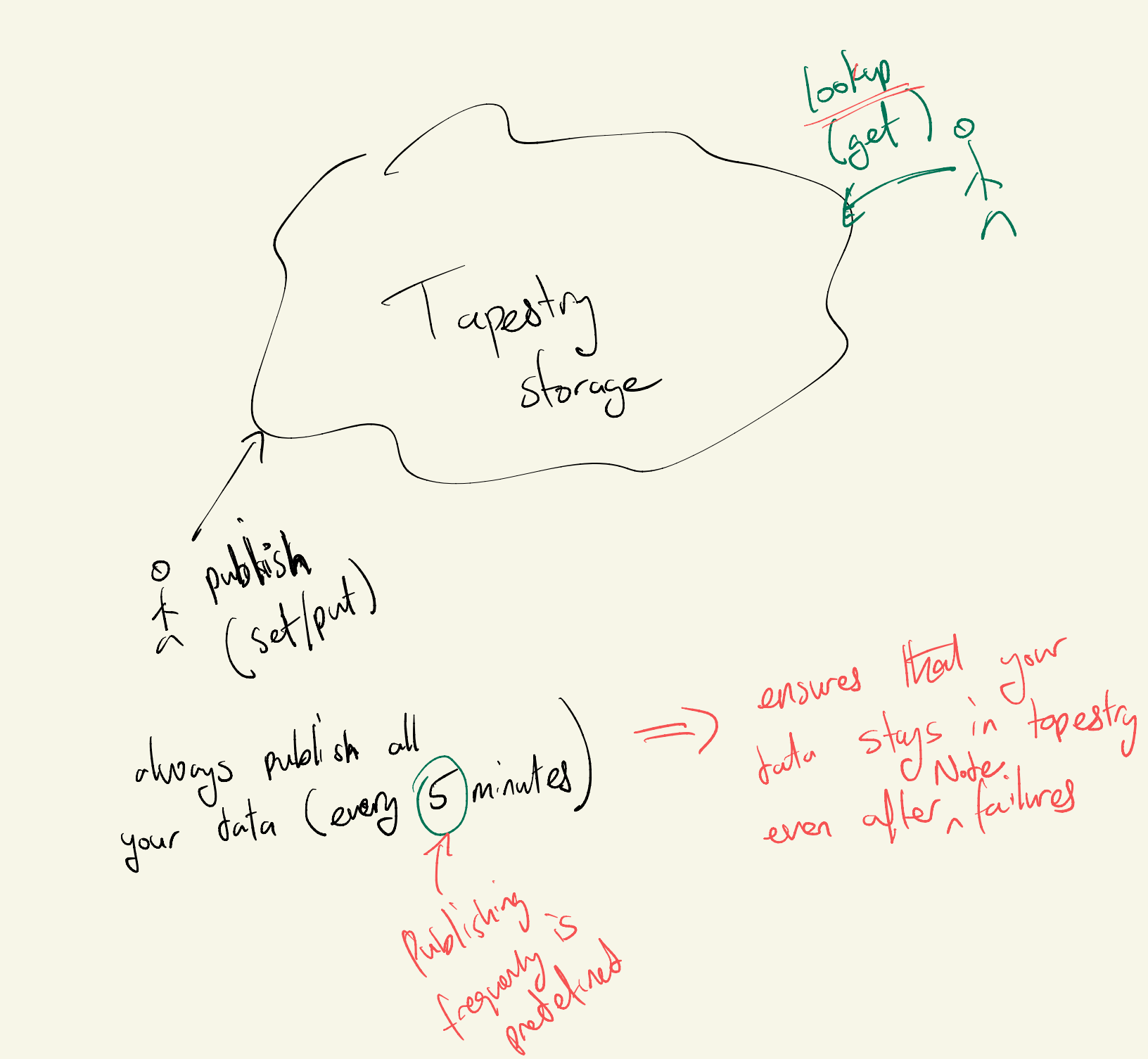
1. follow the rules (constraints placed by columns & rows)
2. if you have several options pick based on closeness
3. there may be empty entries
4. always one entry per row


Context / overview
login/session
FE
LB
Client
FC
web/files
Backend/Replica/
storage shards
eg. Google's Maglev (Chash)
cluster
Tapestry / Dynamo
K
V
abstraction of a large key value store (distributed hash table)
pic
movie
Storage servers (shards)
set(keyID, "...")
get(keyID)
email
shopping

P2P (Peer to Peer)



1. H/W ⇐ different resources
2. nodes come and Go at will
3. Geo distributed



always publish all your data (every 5 minutes) ⟹ ensures that your data stays in tapestry even after failures

Note.

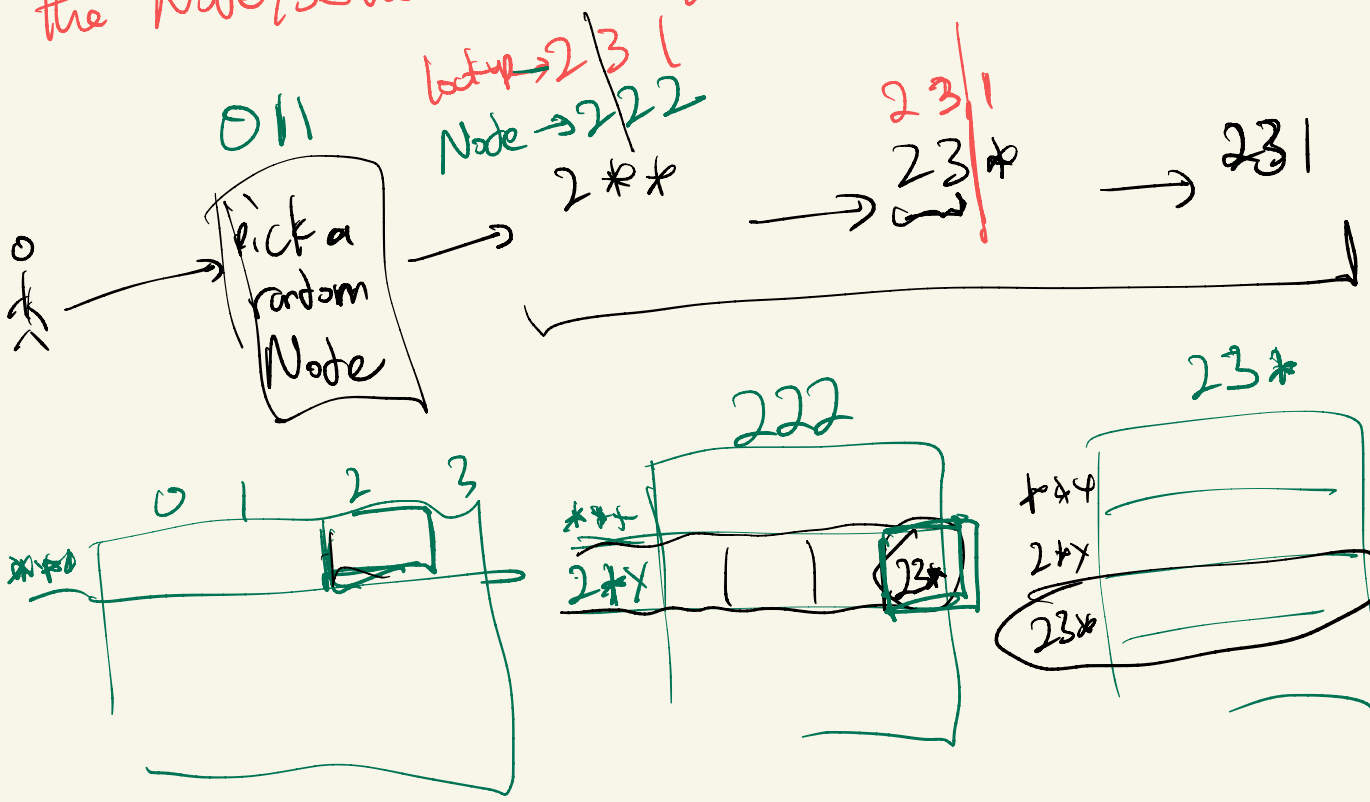
Publishing frequency is predefined

Lookups (in Tapestry)

① Goal: find "Root" → (Node/Server in charge of some set of IDs)

Root = Lookup (231)

└ AKA the Node/Server in charge of that ID



$O(\log_B N)$

Prefix Routing

* every node
* tries to find an entree in its table that has the largest overlap
* then forwards you to this table

Node IDs : 011, 211, 222, 231, 201, 300, 333

211

	0	1	2	3
* * *	011	—	211	300
2 * *	—	211	222	231
2 1 *	—	211	—	—

300

	0	1	2	3
* * *	011	—	222	333
3 * *	300	—	—	333
30 *	300	—	—	—

```
Lookup (keyID, step){
    row = step
    col = keyID.at(step)
    nextNode = Table[col,row]
    step++
    nextNode.lookup(keyID, step)
}
```

333

	0	1	2	3
* * *	011	—	211	300
3 * *	300	—	—	333
33 *	—	—	—	333

start @ 0

lookup (333, 0) → 211

lookup (333, 1) → 300

lookup (333, 2) → 333

Node IDs : 011, 211, 222, 231, 201, 300, 333

211

	0	1	2	3
* * *	011	—	211	300
2 * *	—	211	222	231
21 *	—	211	—	—

300

	0	1	2	3
* * *	011	—	222	333
3 * *	300	—	—	333
30 *	300	—	—	—

```
Lookup (keyID, step){
    row = step
    col = keyID.at(step)
    nextNode = Table[col,row]
    step++
    nextNode.lookup(keyID, step)
}
```

333

	0	1	2	3
* * *	011	—	211	300
3 * *	300	—	—	333
33 *	—	—	—	333

lookup(111, 0) → 300

lookup(111, 1) → 222

lookup(111, 2) → 211

row=2 col=1

21 *

211 ⇐ as the root for 111

Surrogate routing: routing when you take an empty entry

Node failure / removal

Graceful

* Say Good bye to nodes & hand over your data

(1) who to give your data to?

new root

(2) how to update broken routing tables

need to figure out which tables are impacted by exit!!!

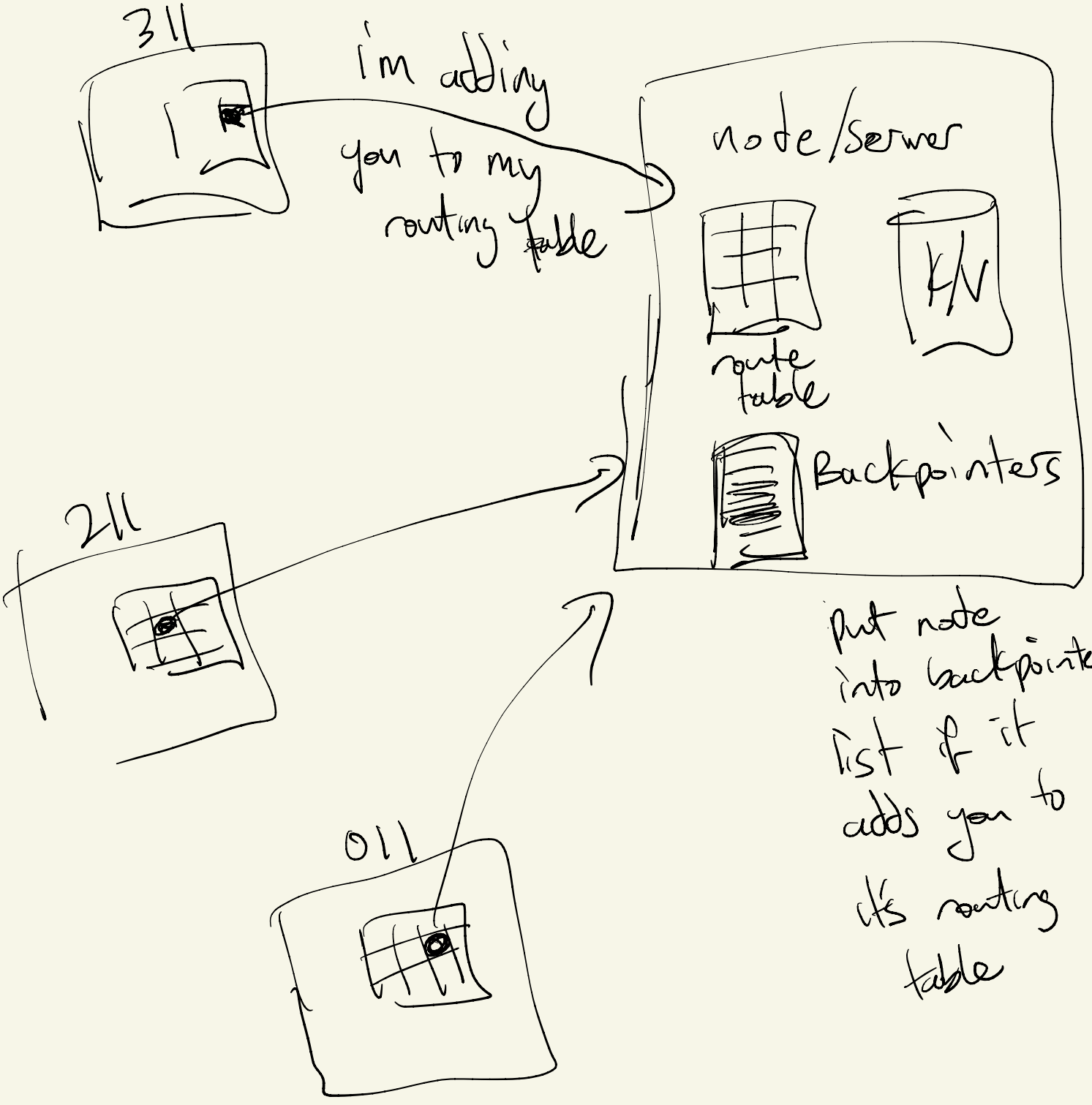
Use Backpointers to determine nodes to notify

not Graceful

Just disconnect

* failure is detected w/ heart beats

* data persists because clients republish the data


311
i'm adding you to my routing table
node/server
route table
K/V
Backpointers
211
011
put node into backpointer list & it adds you to it's routing table

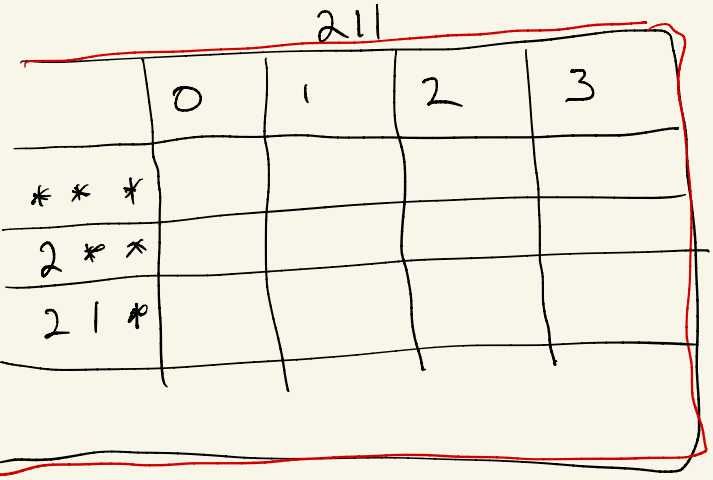
Summary

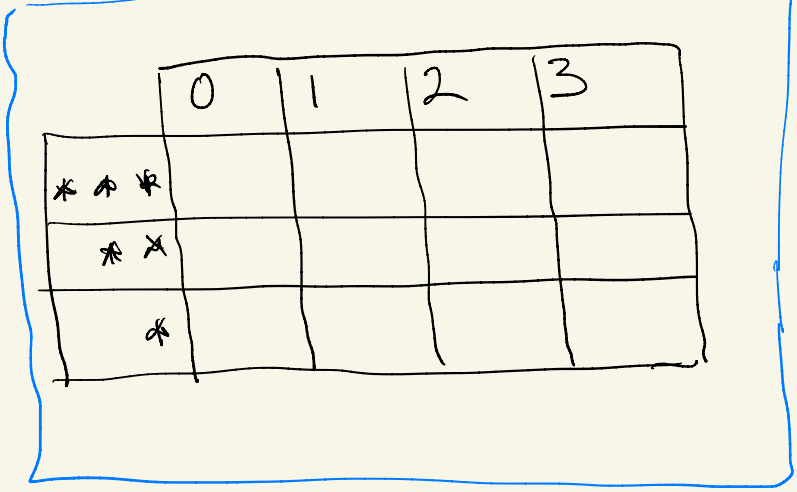
1. Overview / context for Tapestry
2. Building a route table
3. Routing in Tapestry (lookups)
4. Exiting the Tapestry P2P network
   - graceful & ungraceful

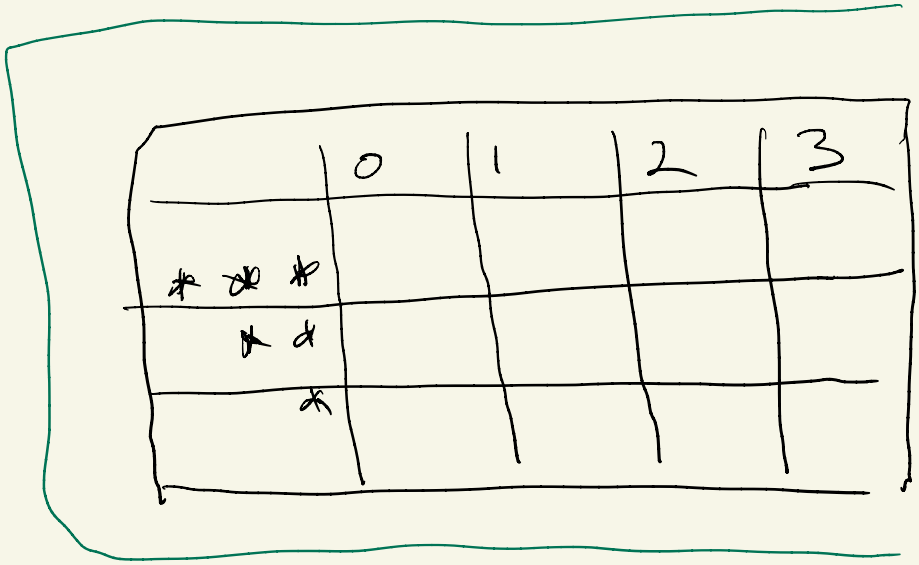
5. How to add nodes into the network

Node IDs : 011, 211, 222, 231, 201, 300, 333

211







Node IDs : 011, 211, 222, 231, 201, 300, 333