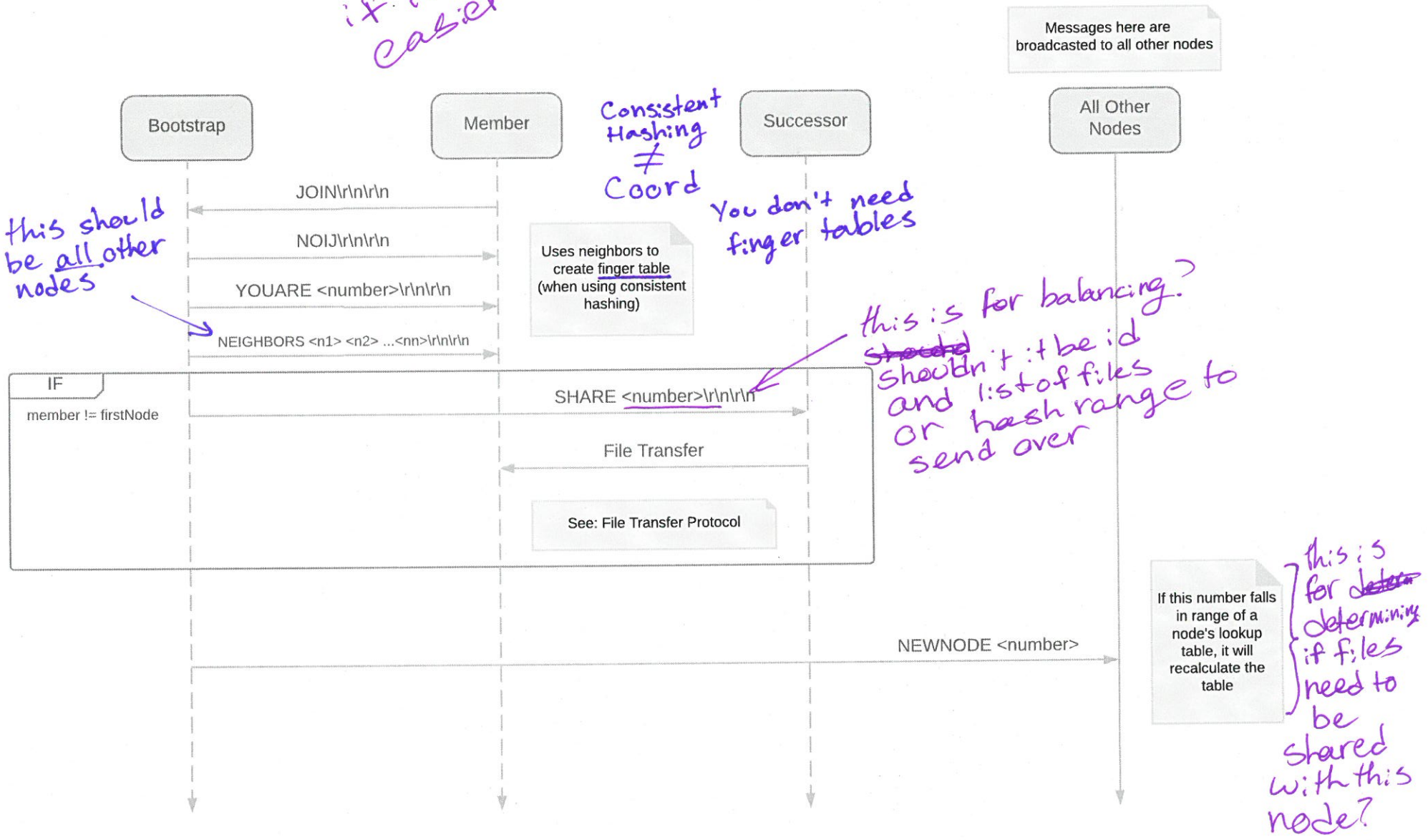
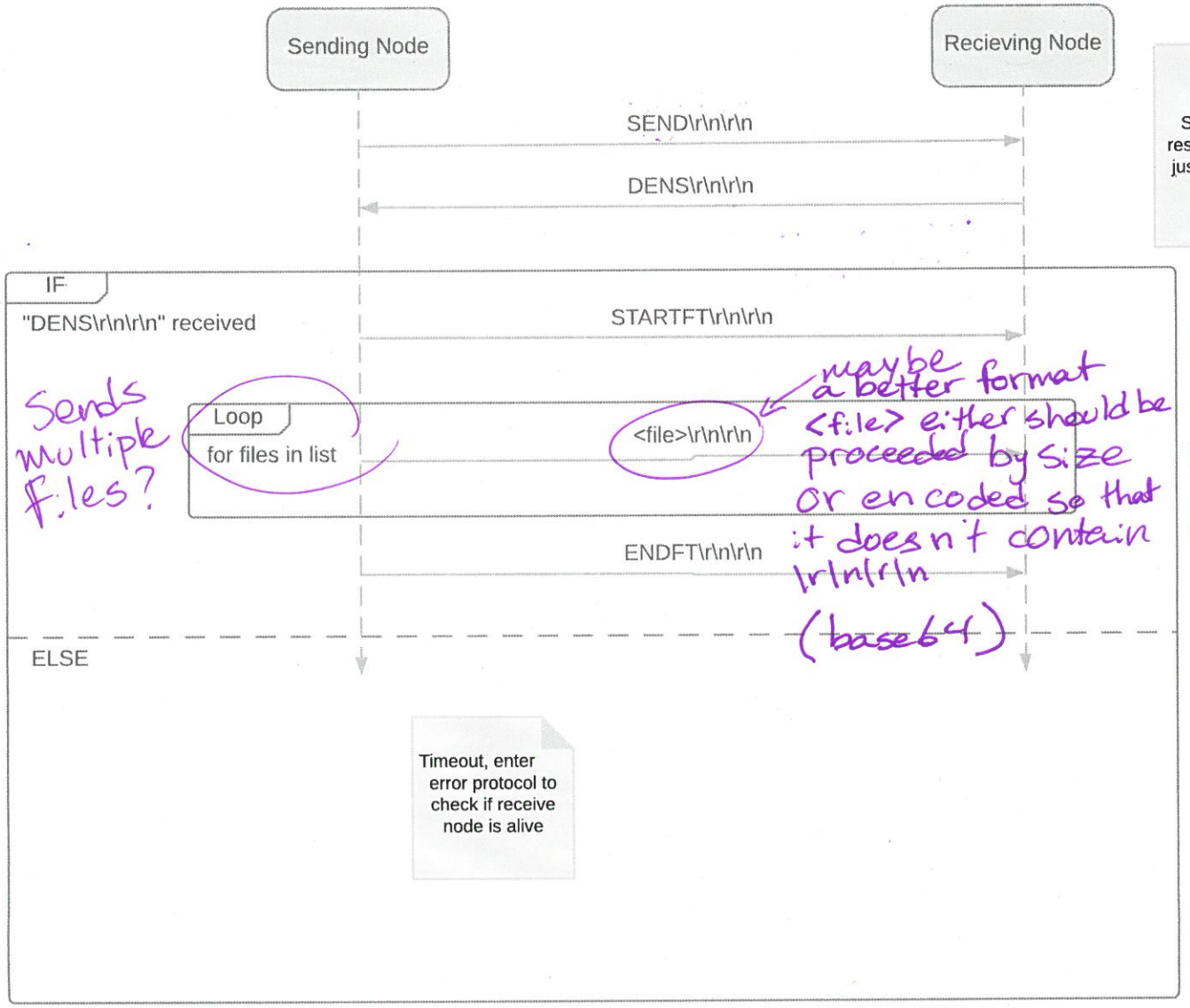
Bradley Wellman, Zachary McCleary | April 18, 2018

You are allowed to send pkt lens if it makes things easier

Join Protocol



File Transfer Protocol

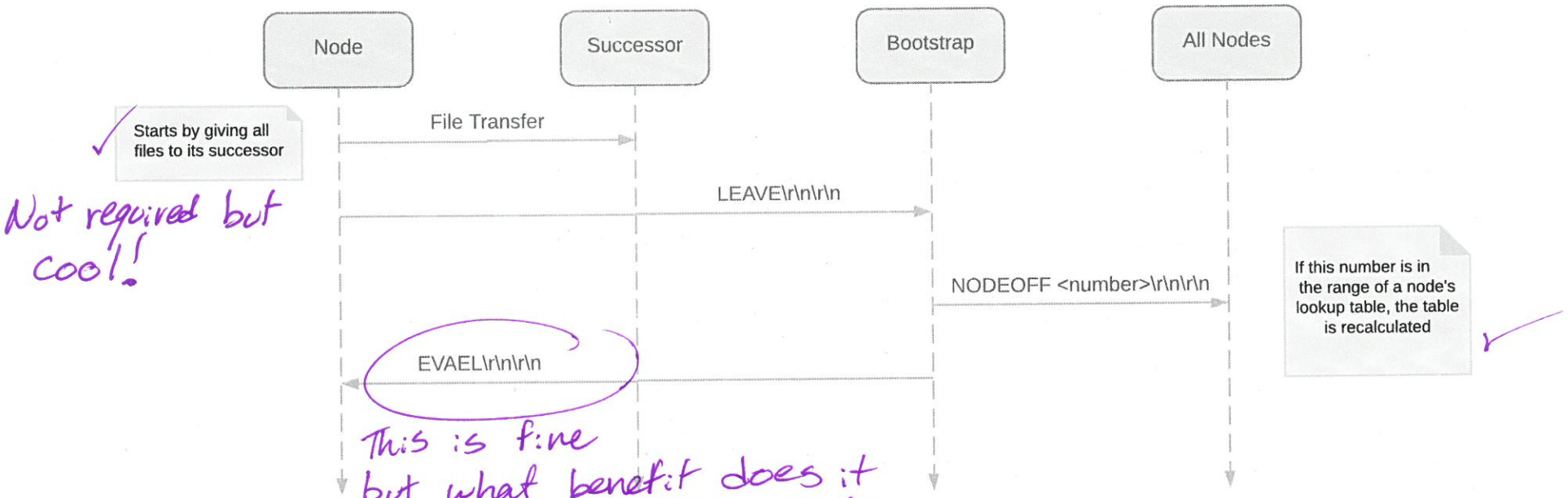


Sends multiple files?

maybe a better format <file> either should be proceeded by size or encoded so that it doesn't contain \r\n\r\n (~~base64~~)

TCP is good what's the benefit of ACKs they imply syncing up state of two programs but how necessary is that?

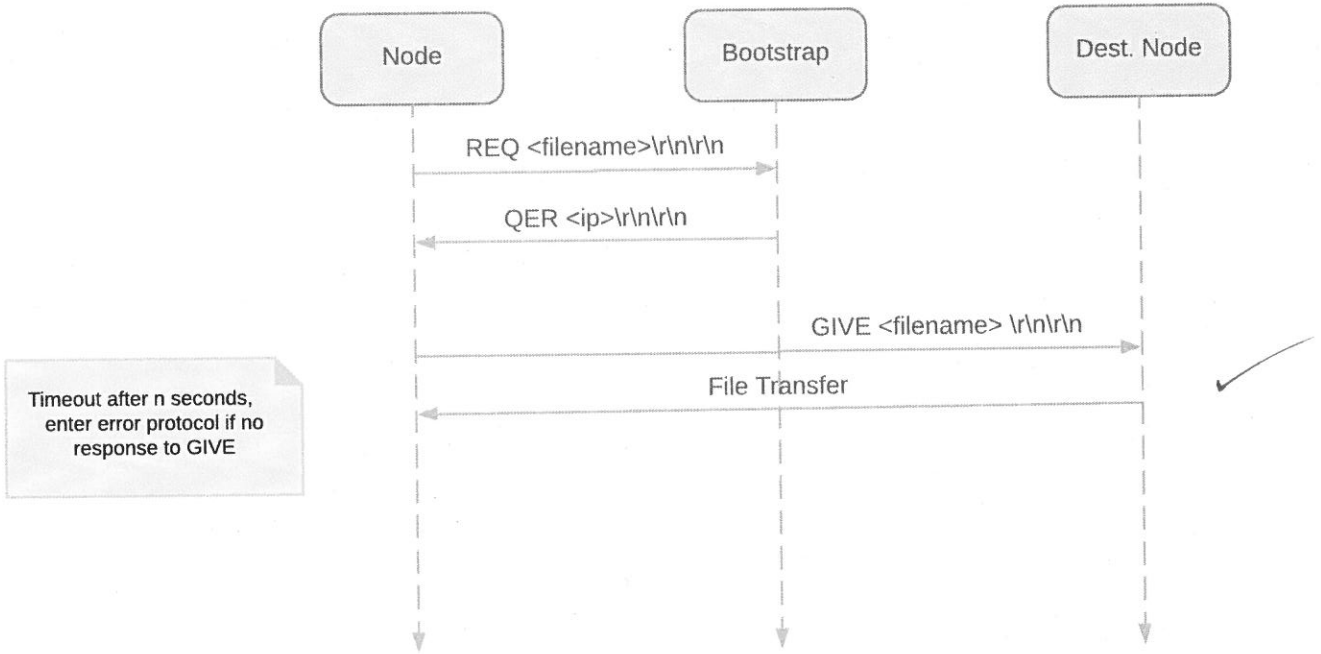
Leave Network Protocol



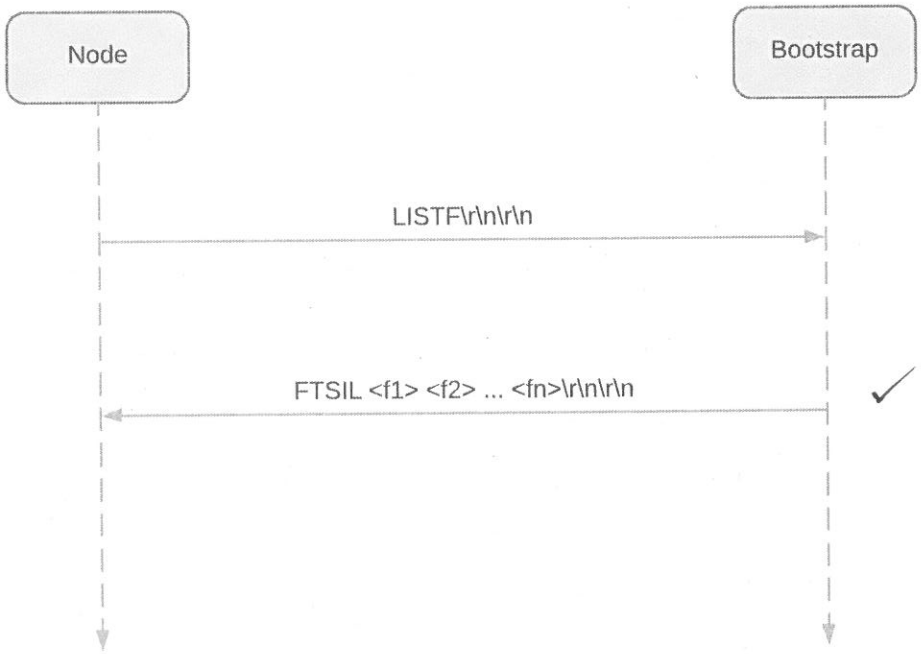
Not required but cool!

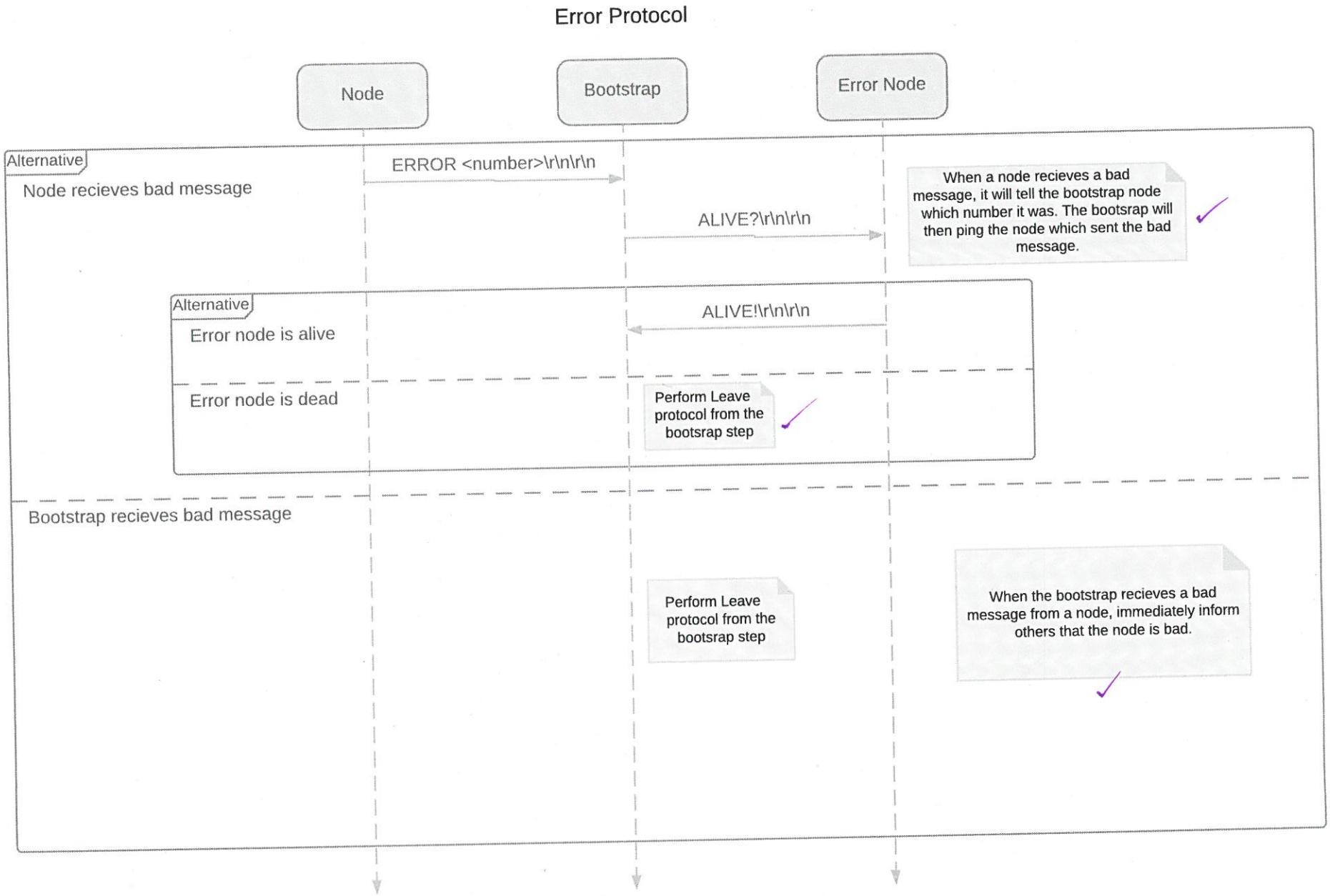
This is fine but what benefit does it have to keep the node waiting on BS? just confirmation?

Request File Protocol
(Base assignment)



List Files Protocol
(Base Assignment)



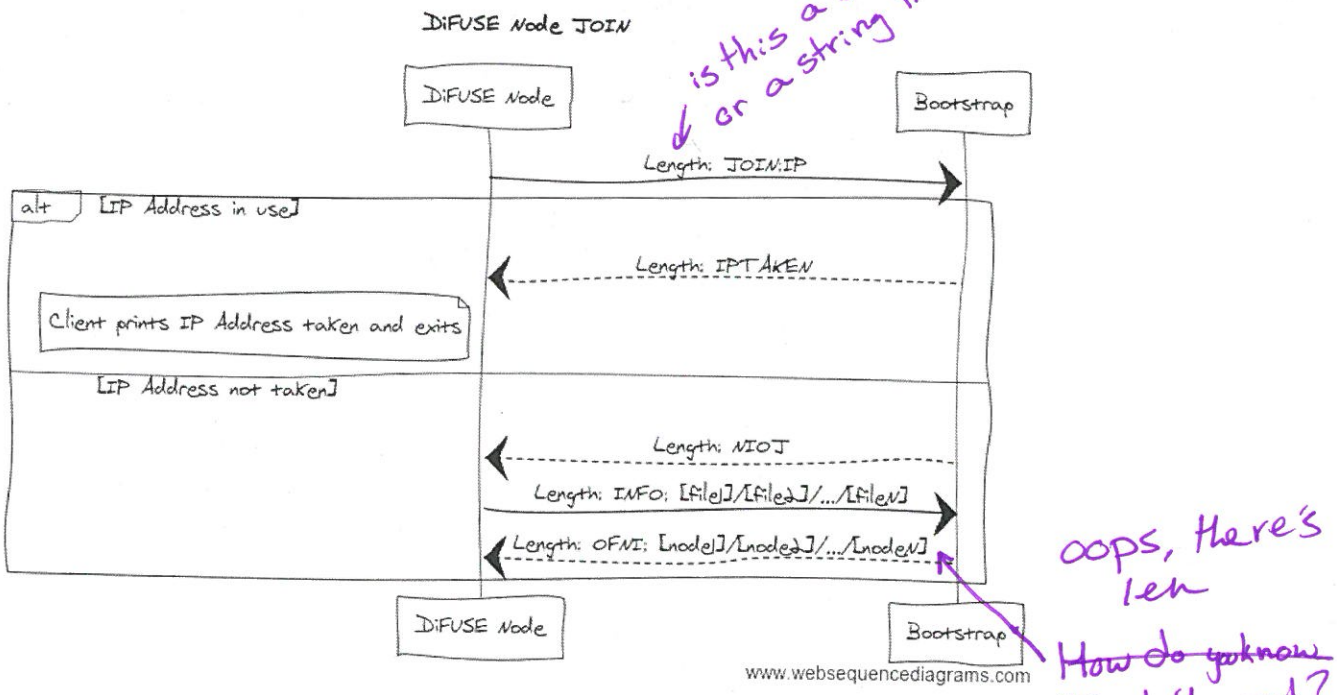

Error Protocol
Node
Bootstrap
Error Node
Alternative
Node recieves bad message
ERROR <number>\r\n\r\n
When a node recieves a bad message, it will tell the bootstrap node which number it was. The bootsrap will then ping the node which sent the bad message.
ALIVE?\r\n\r\n
Alternative
Error node is alive
ALIVE!\r\n\r\n
Error node is dead
Perform Leave protocol from the bootsrap step
Bootstrap recieves bad message
Perform Leave protocol from the bootsrap step
When the bootstrap recieves a bad message from a node, immediately inform others that the node is bad.

CSE 392: Protocol Design Document

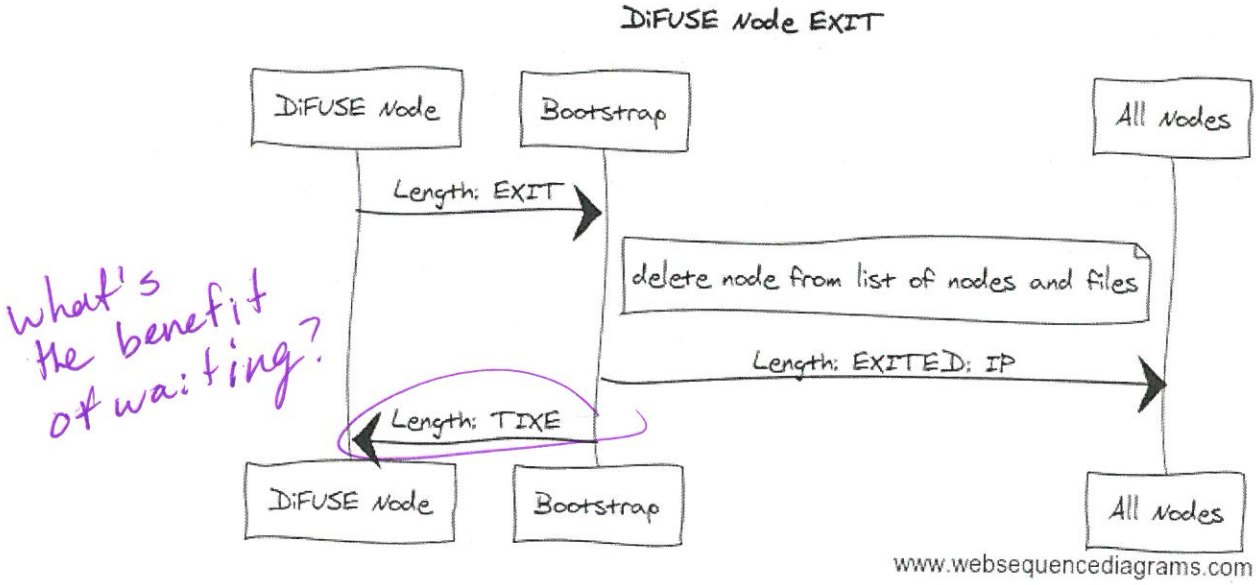
Jamie Kunzmann 110509190
Mendy Wu 110483971

DiFUSE Node to Bootstrap Protocol (baseline)

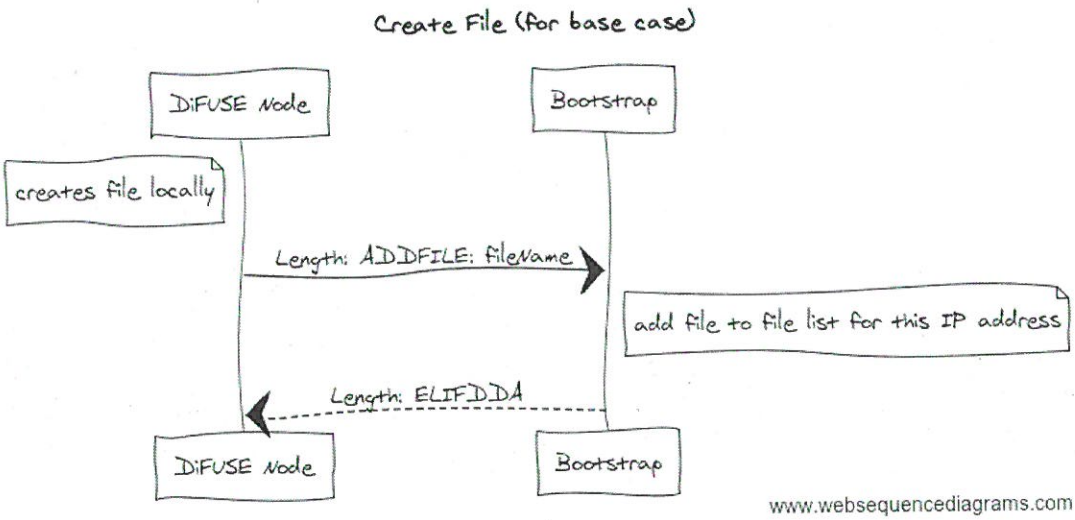
Join



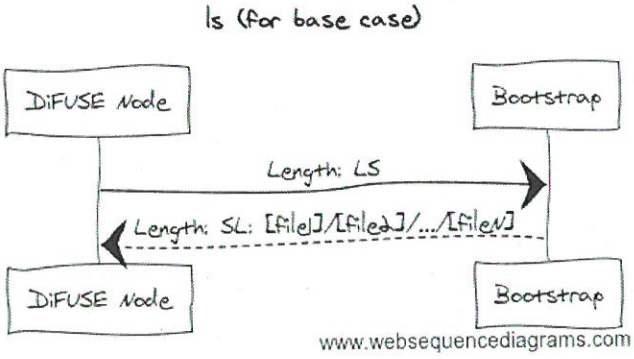
Exit



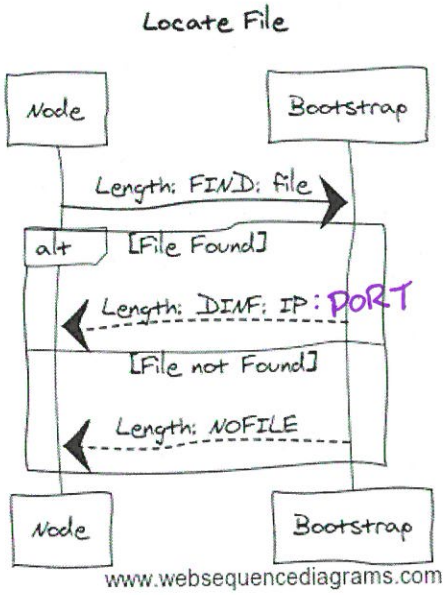
Create File



ls

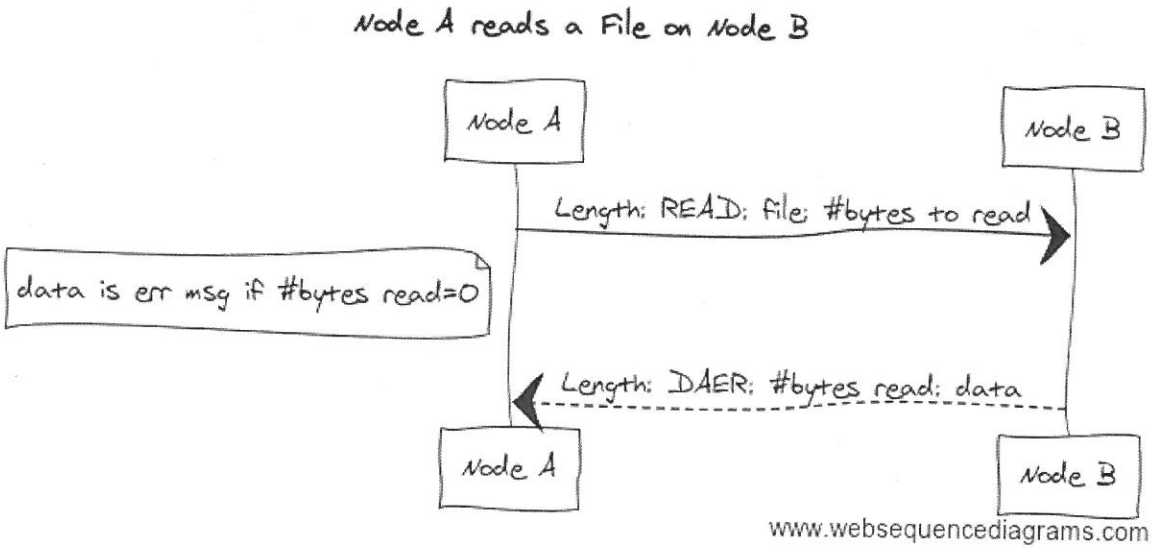


Locate File

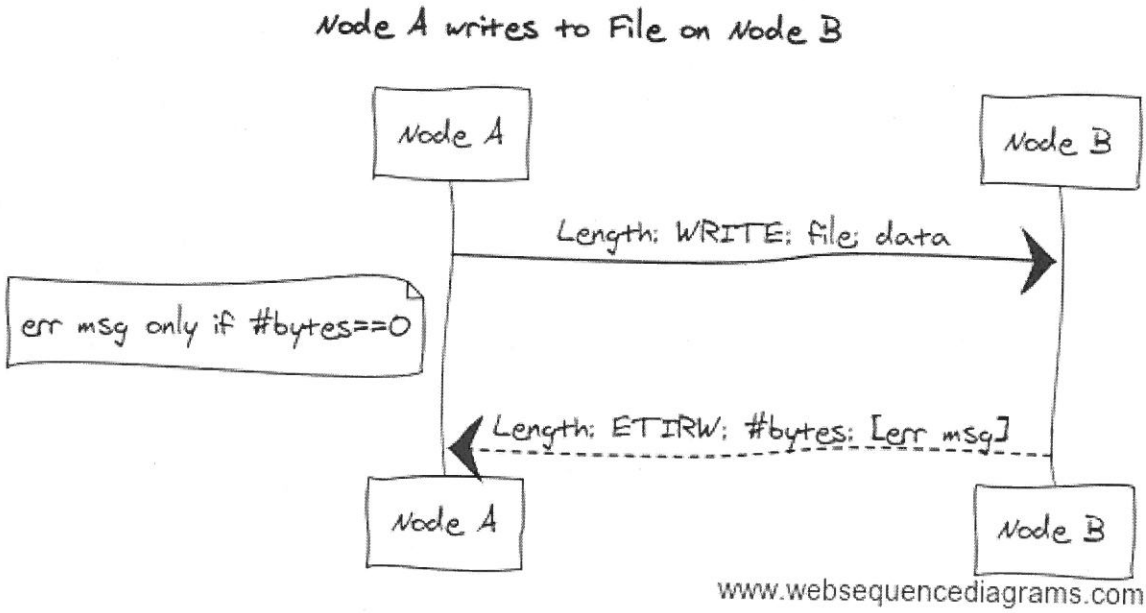


Node to Node Protocol (baseline)

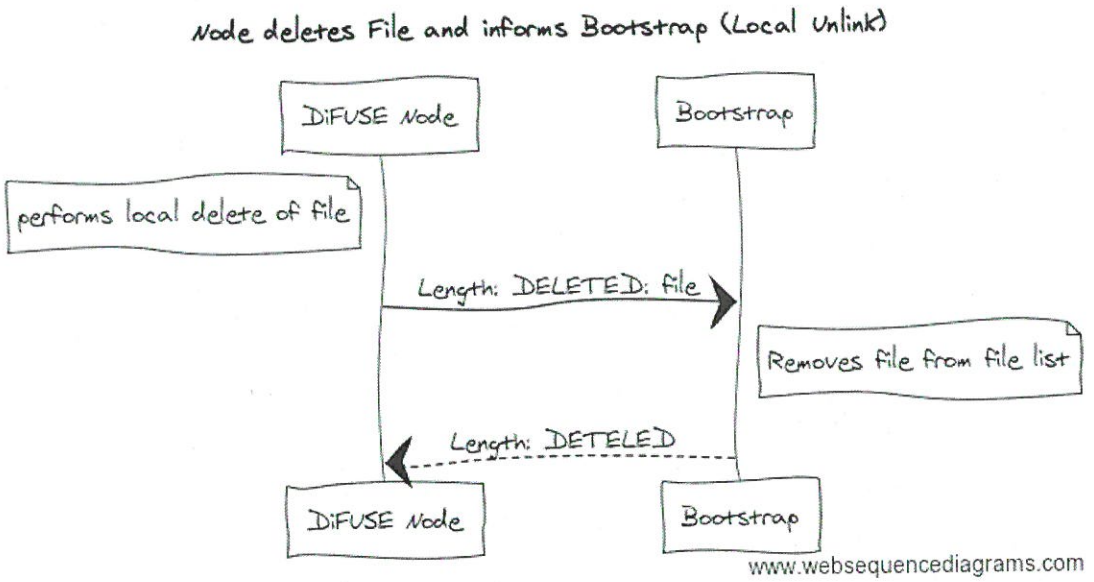
Read

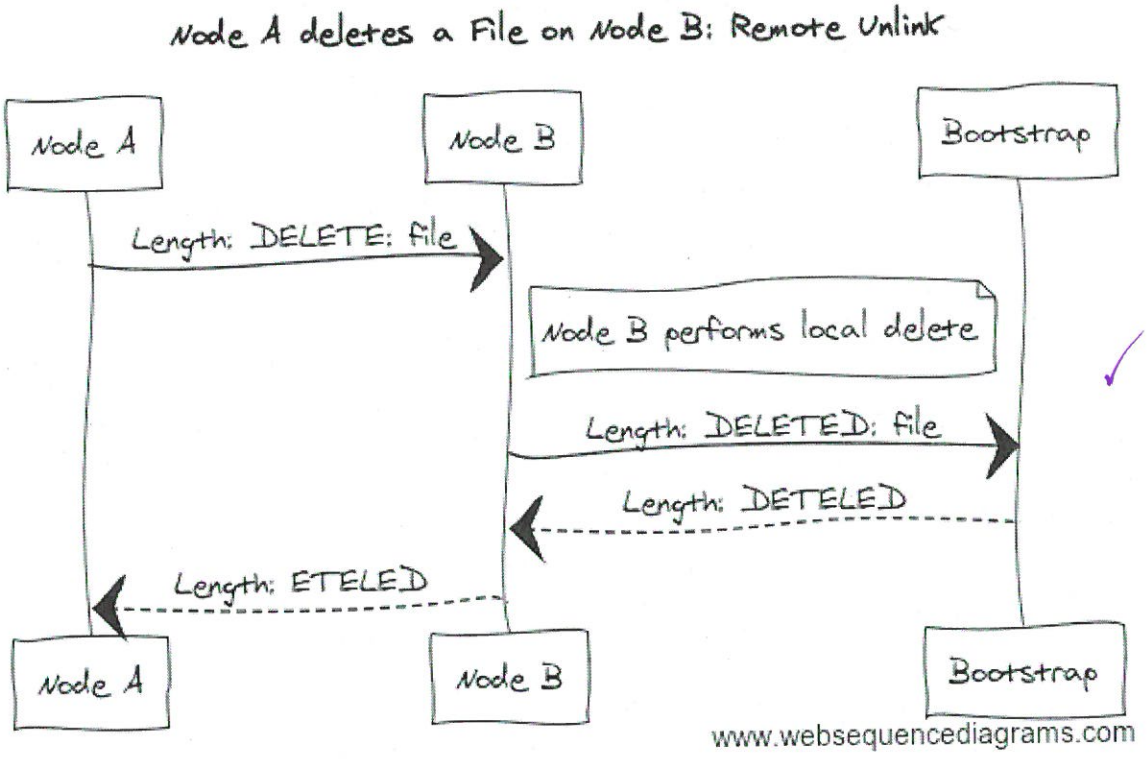


Write



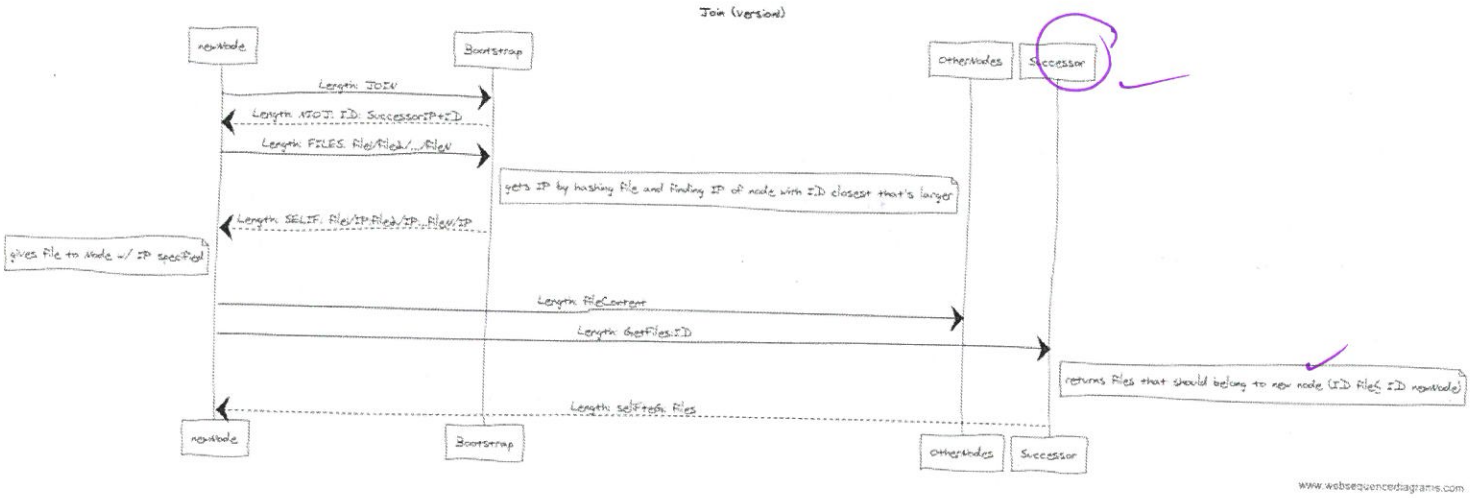
Delete

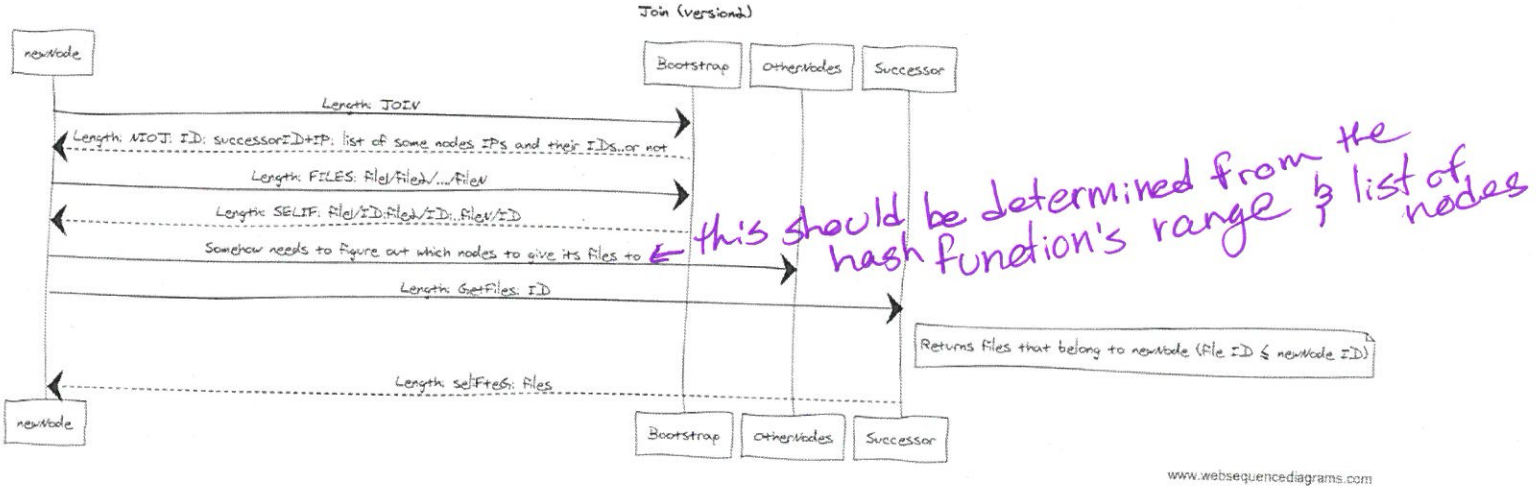

Node deletes File and informs Bootstrap (Local Unlink)
DiFUSE Node
Bootstrap
performs local delete of file
Length: DELETED: file
Removes file from file list
Length: DETELED
DiFUSE Node
Bootstrap
www.websequencediagrams.com


Node A deletes a File on Node B: Remote Unlink
Node A
Node B
Bootstrap
Length: DELETE: file
Node B performs local delete
Length: DELETED: file
Length: DETELED
Length: ETELED
Node A
Node B
Bootstrap
www.websequencediagrams.com

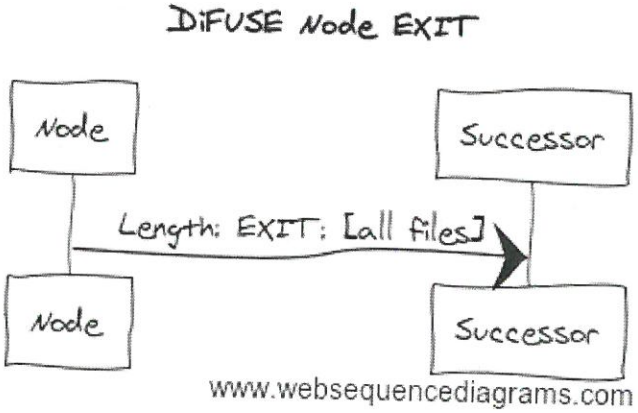
Consistent Hashing Protocol Attempt

Join





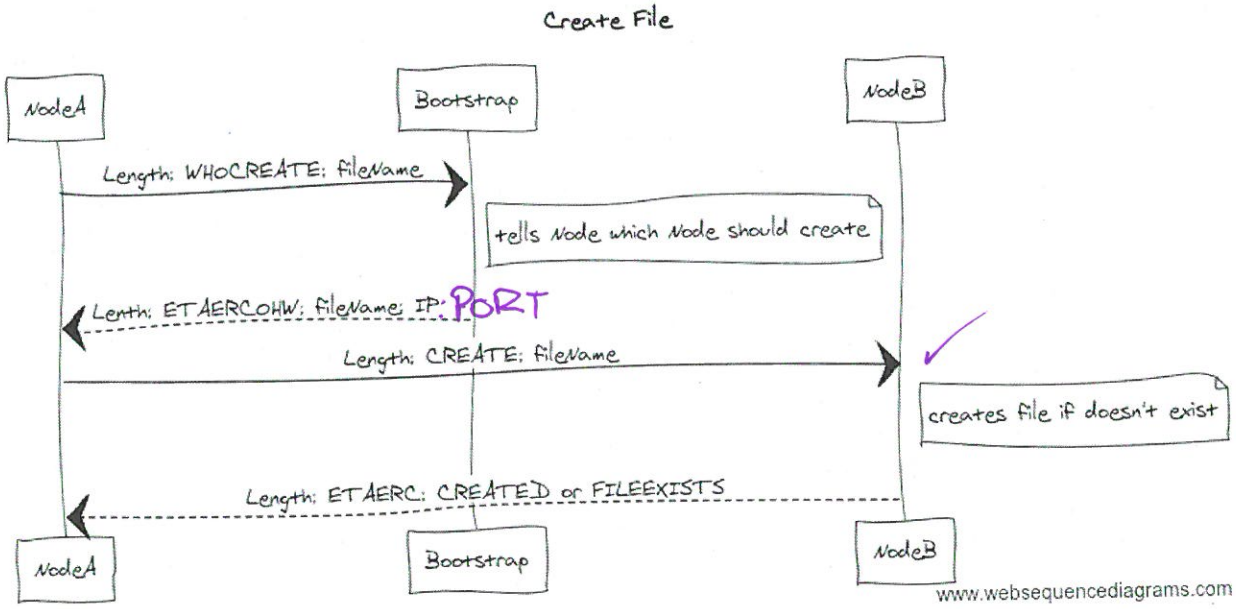
Exit



Not required but cool!

It's okay if files are lost on exit and its okay if FUSE always boots w/ no files

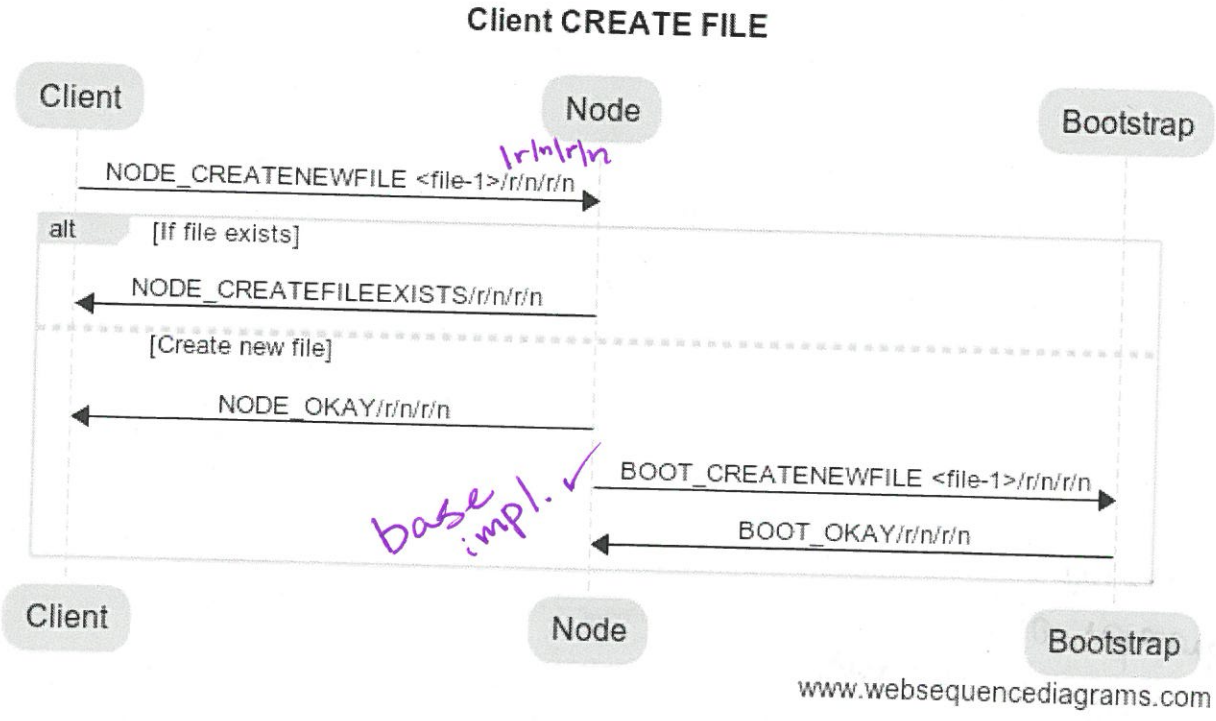
Create

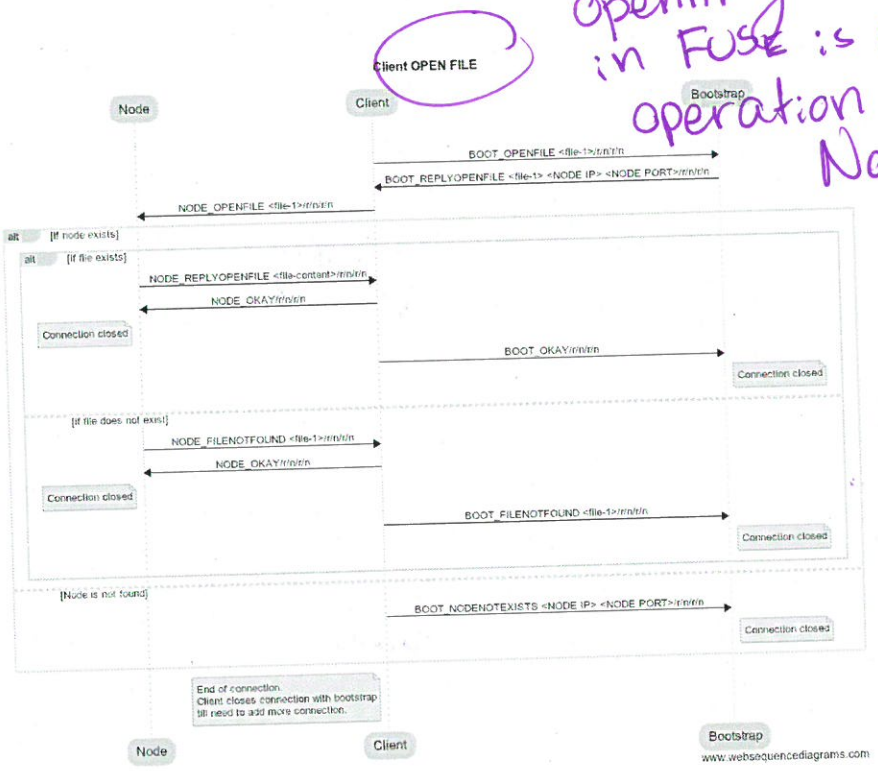

Create File
NodeA
Bootstrap
NodeB
Length: WHOCREATE: fileName
tells Node which Node should create
Length: ETAERCOHW: fileName: IP: PORT
Length: CREATE: fileName
creates file if doesn't exist
Length: ETAERC: CREATED or FILEEXISTS
NodeA
Bootstrap
NodeB
www.websequencediagrams.com

DiFUSE Protocol

by Manjur H Khan

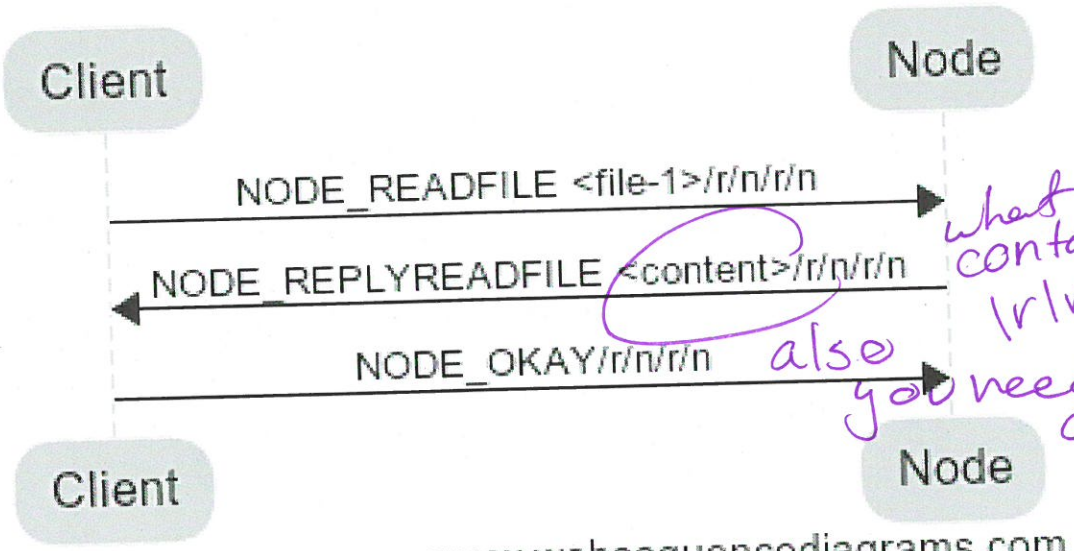
ps: I messed up printing it 3 times. sorry.





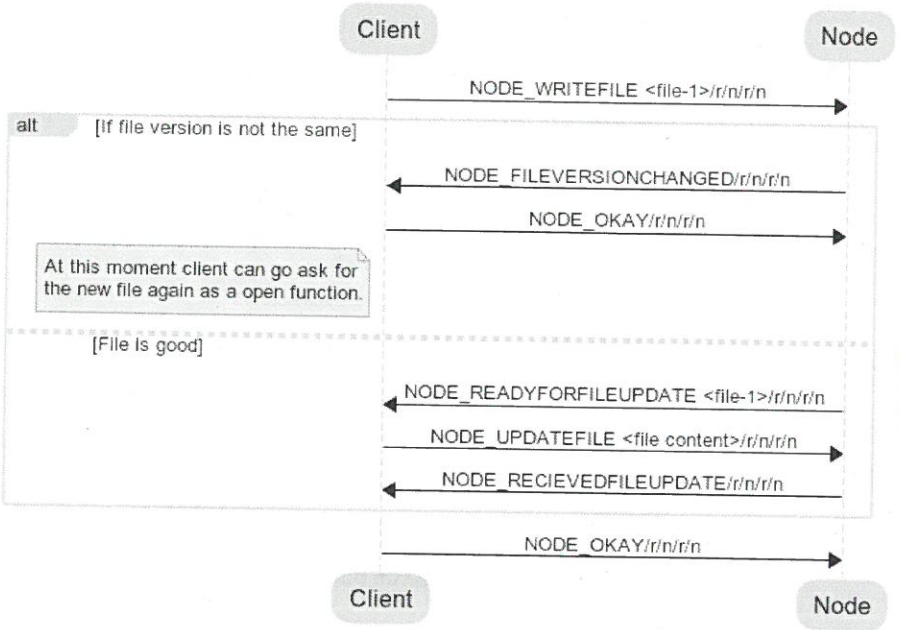
Openning a file in FUSE is a local operation only
No networking necessary

Client READ FILE



what if contains \r\n\r\n
also you need len and offset

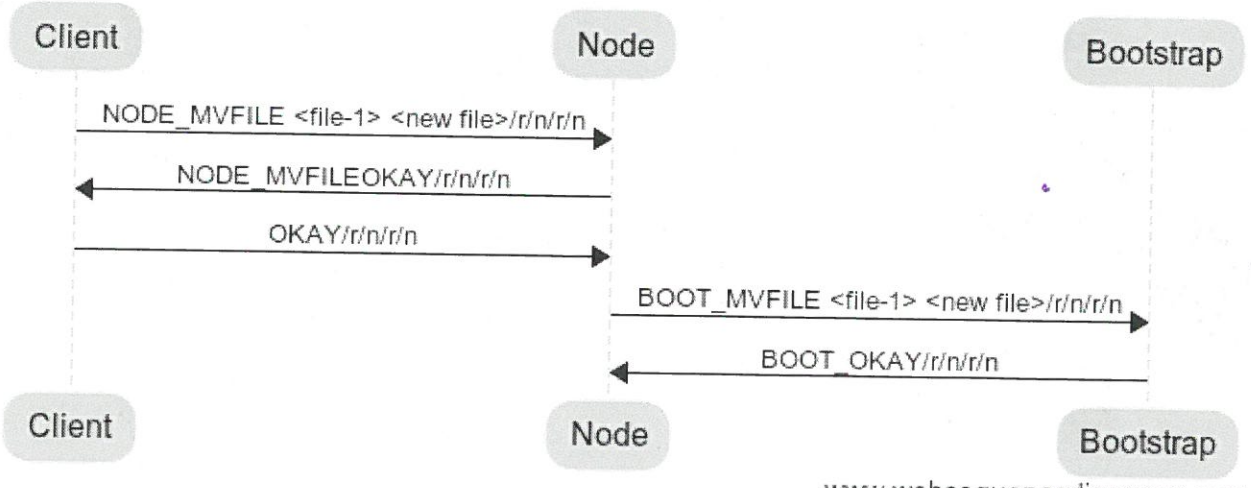
Client WRITE FILE



data
offset
len

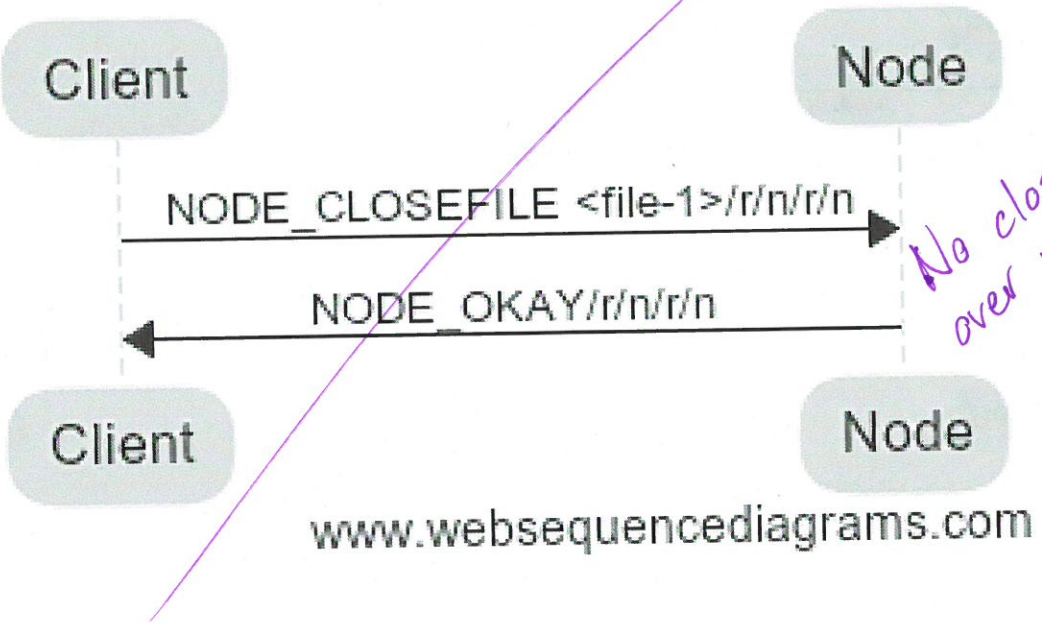
www.websequencediagrams.com

Client MV FILE



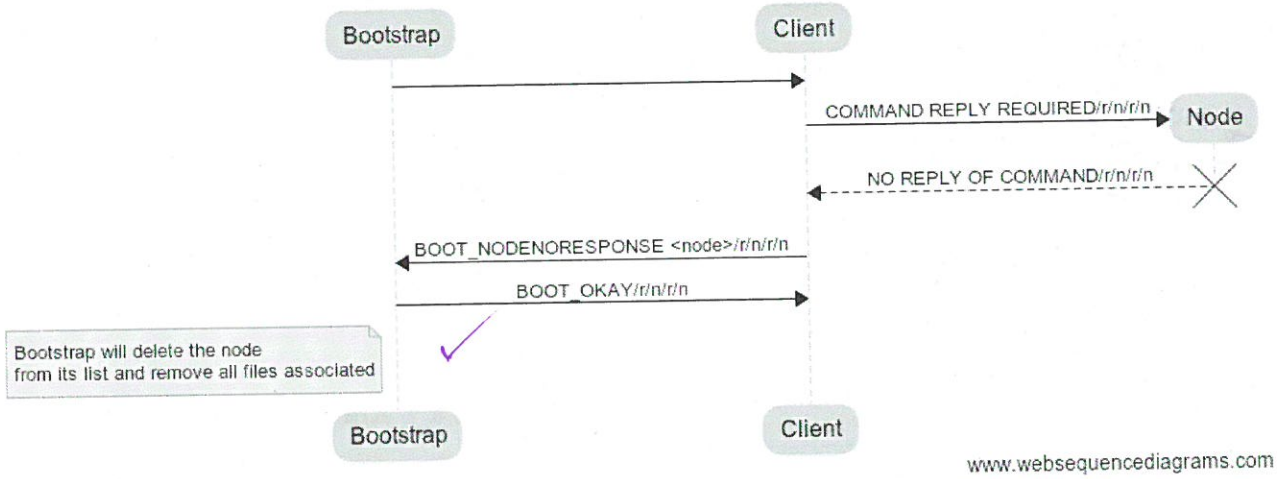
www.websequencediagrams.com

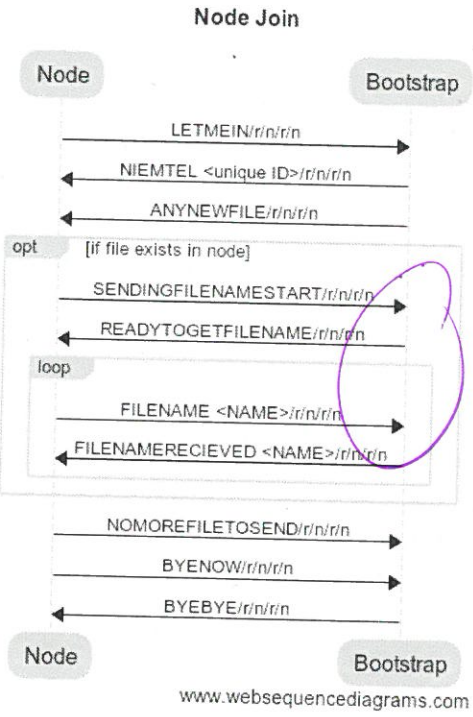
Client CLOSE FILE



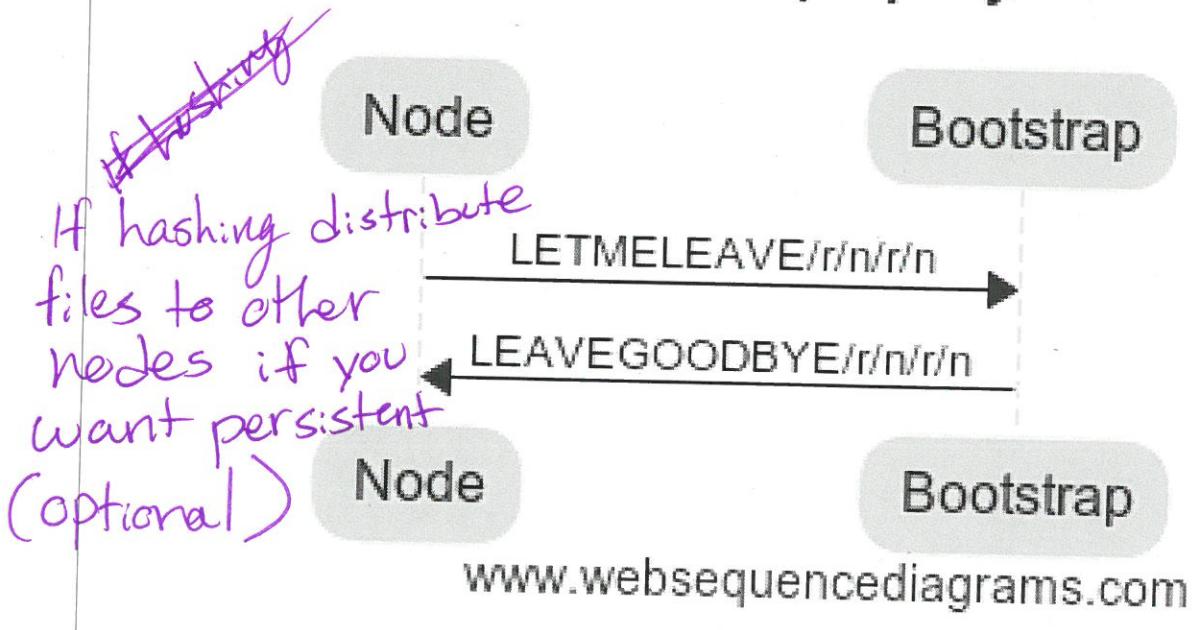
No closing over network

Error while doing anything with nodes from client





The node doesn't need to share files on startup.
like it can always just start empty

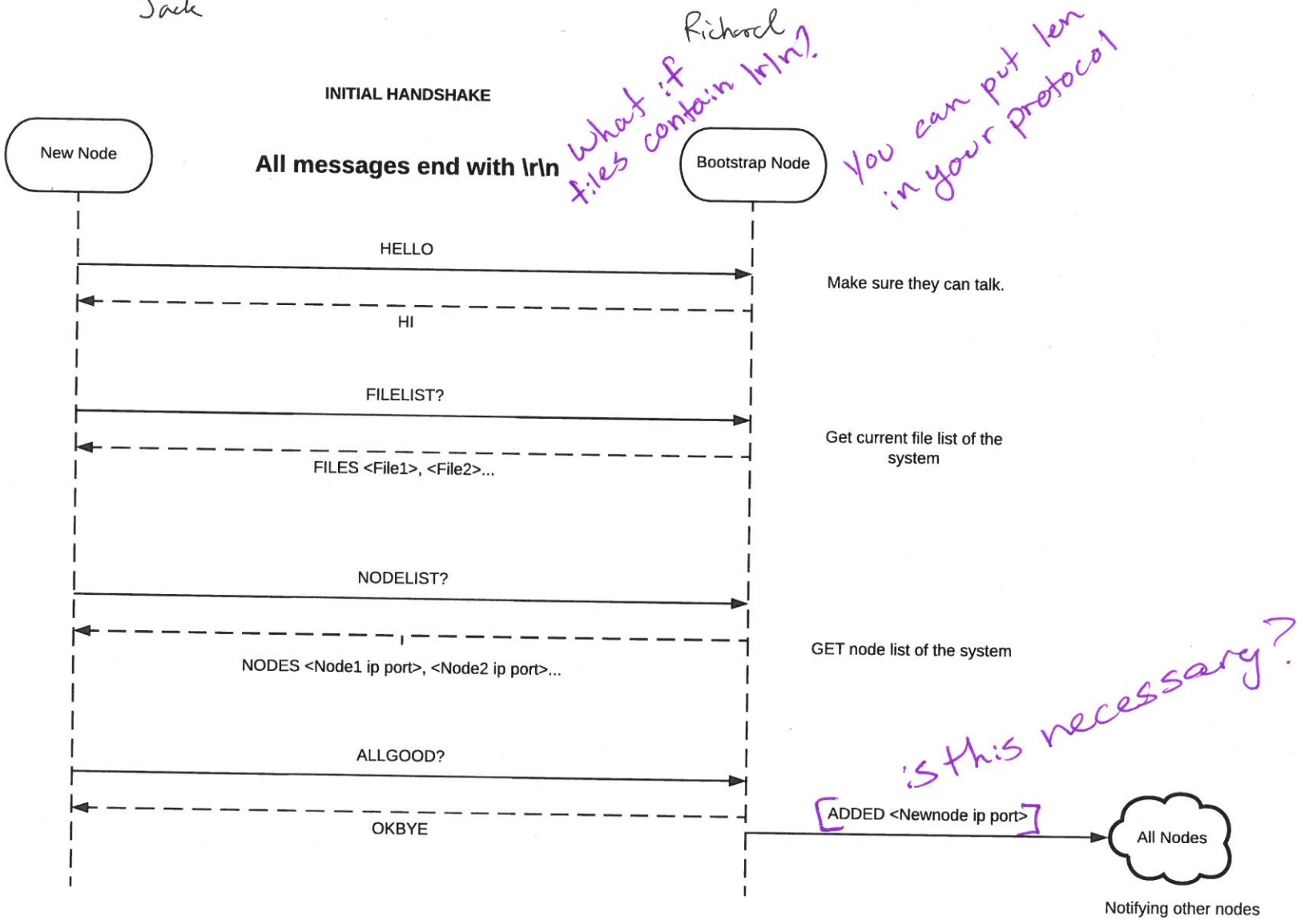


~~if hashing~~

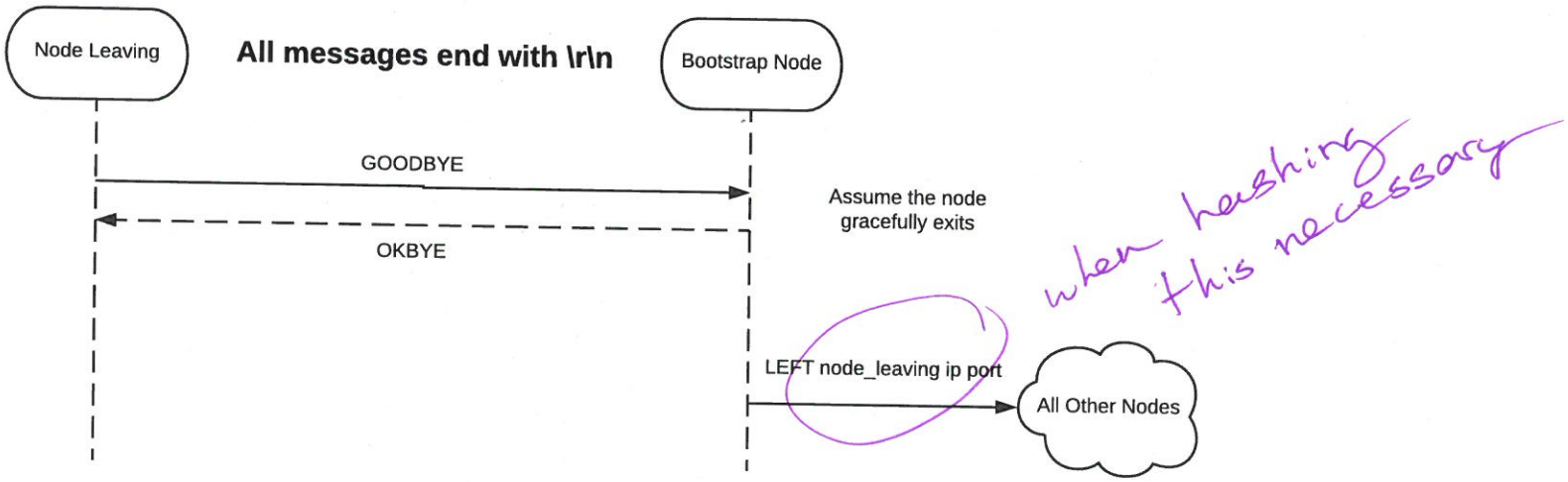
If hashing distribute files to other nodes if you want persistent (optional)

Jianneng Wu & Yueyang Huang

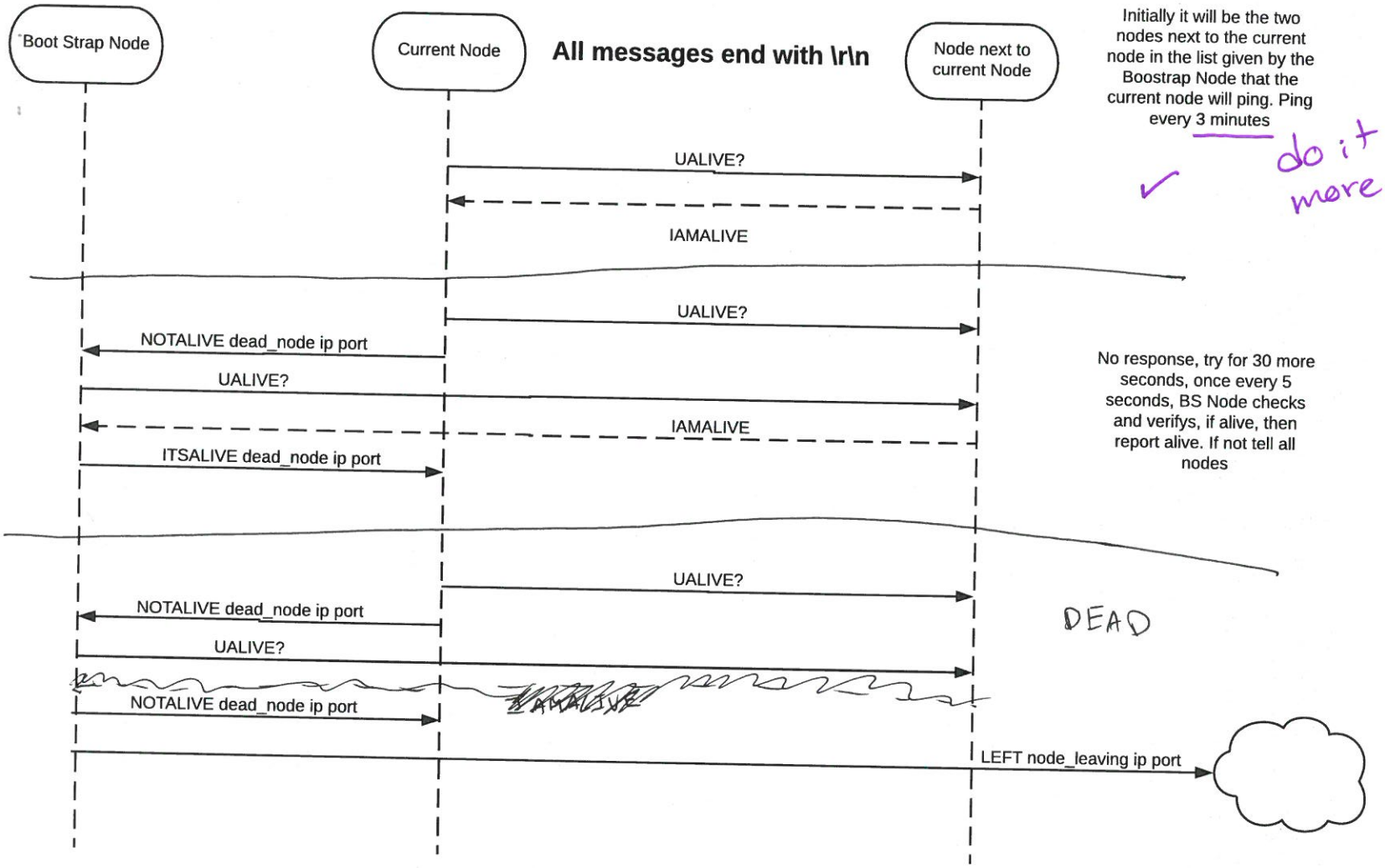
Jack Richard



Node Leaving Gracefully

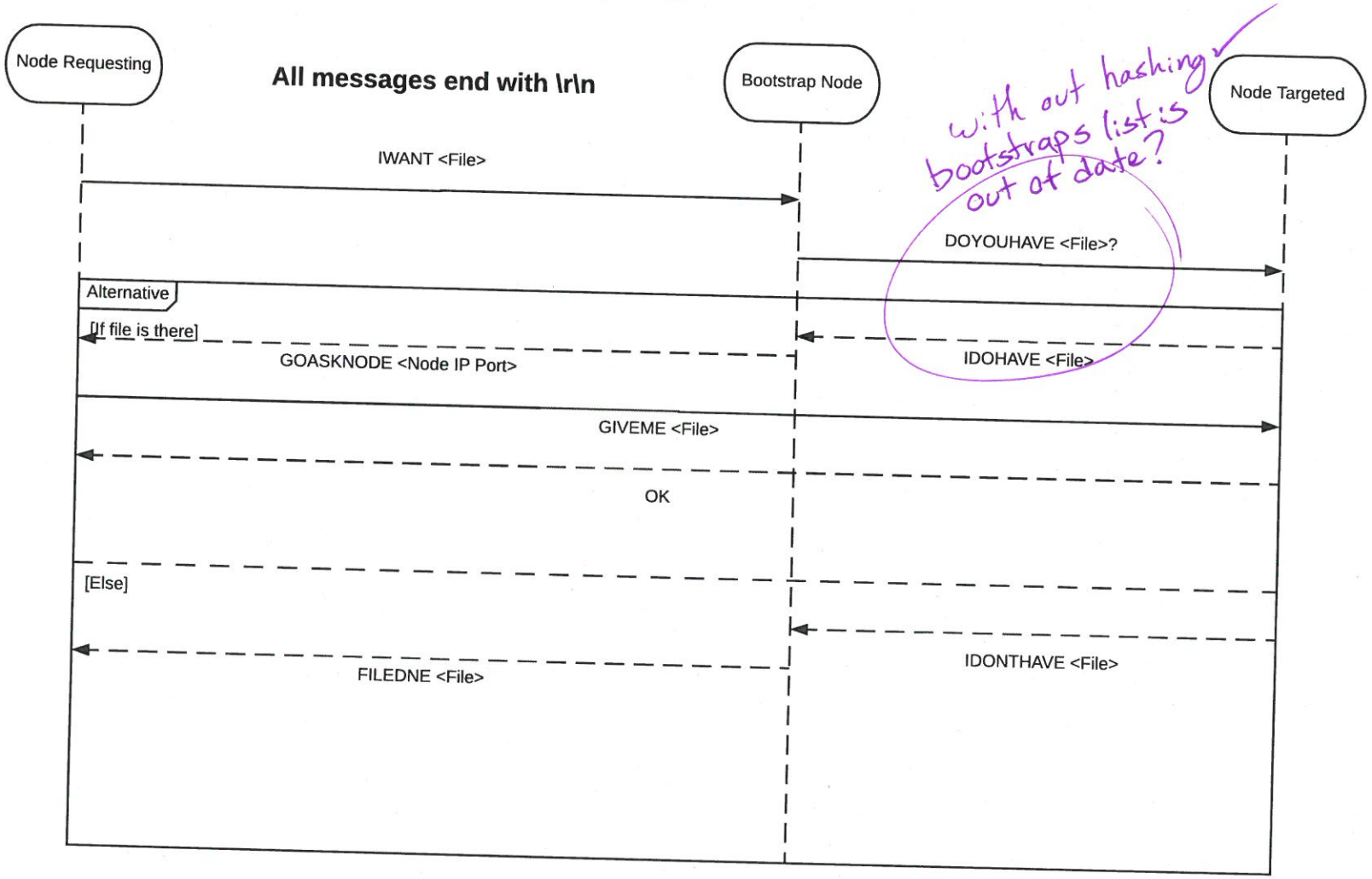


Heart Beat from Other Nodes
(NON HASING VERSION)



do it a little more often for debugging

Request Data



Brian Gavin
Ngan Nguyen

Known Port: 10920 ✓

Packet Format
(Row represents 32 bits) It's 2018, 64 bit is cooler

Verb
Payload Size (in bytes)
Status
// // Payload (payload size bytes) // //

✓

Payload is JSON string of payload size bytes whos format depends on the verb.

Status
Field is either OK or ERROR (0 or 1)

Utilized to indicate the success of an operation during a specific situation; when expecting a certain response, or when the ERROR or OK verbs are not appropriate

Verbs

All verbs are a 32-bit Enumerated types

All good

0: ERROR
```
{
        msg: string
}
```

1: OK (no payload)

2: JOIN
```
{
        files: [string]
}
```

if you attempt hashing you'll need a unique id for nodes and be able to broadcast to all nodes the hashtable updates.

3: EXIT (no payload)

4: LOOKUP_REQ

{
    filename: string
}

5: LOOKUP_RES

{
    ip: string
}

6: STAT_REQ

{
    filename: string
}

7: STAT_RES

{
    stat: [int] *
}

*the stat field is an array of integers that corresponds to the fields of a **struct stat****

In Python, this can be passed to the constructor of a **os.stat object and be reconstructed.

8. READ_REQ

{
    filename: string
    cnt: int
    offset: int
}

9. READ_RES

{
    bytes : [int]
    cnt : int
}

10. WRITE_REQ

{
    filename: string
    cnt: int
    offset: int
    bytes: [int]
}

11. WRITE_RES

```
{
        cnt: int
}
```

12. RENAME_REQ

```
{
        old: string
        new: string
}
```

12. RENAME_RES (no payload)

13. TRUNC_REQ

```
{
        filename: string
        len: int
}
```

14. TRUNC_RES (no payload)

15. UNLINK_REQ

```
{
        filename: string
}
```

16. UNLINK_RES (no payload)

17. READDIR_REQ (no payload)

18. READDIR_RES

```
{
        entries: [string]
}
```

NAUMAN SHAHZAD
Raymond Lam

CSE 392

Protocol

Try using more standardized headers

Try keeping data more isolated?

BS = bootstrap, N = node

Node joins Cluster

BS → N: HI \r\n\r\n

N → BS: IH \r\n\r\n

BS → N: FILES, ... \r\n\r\n { list of file names

N → BS: DATA, filename .... \r\n\r\n

Why does bootstrap have all files?

File Name Conflict

BS → N: CON, FILENAME, <filename> \r\n\r\n

New File

BS → N: FILES, ... \r\n\r\n { list of file names

N → BS: DATA, filename .... \r\n\r\n

Find a File

BS → N: FIND, filename \r\n\r\n

N (file is on) → BS: ALIVE \r\n\r\n

BS → N: OK \r\n\r\n

N → BS: LOCATION .... \r\n\r\n

Get File(s) Node to Node

N1 → N2: GIVEME, ... \r\n\r\n

N2 → N1: DATA, filename, ... \r\n\r\n

Get File(s) node to node

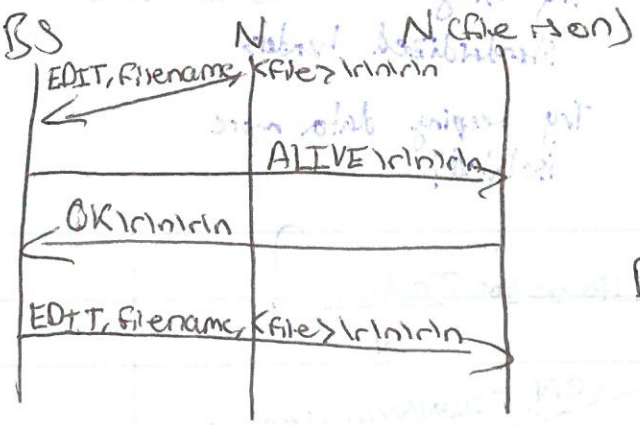
N1 → N2: GIVEME, ... \r\n\r\n

N2 → N1: NO, filename \r\n\r\n

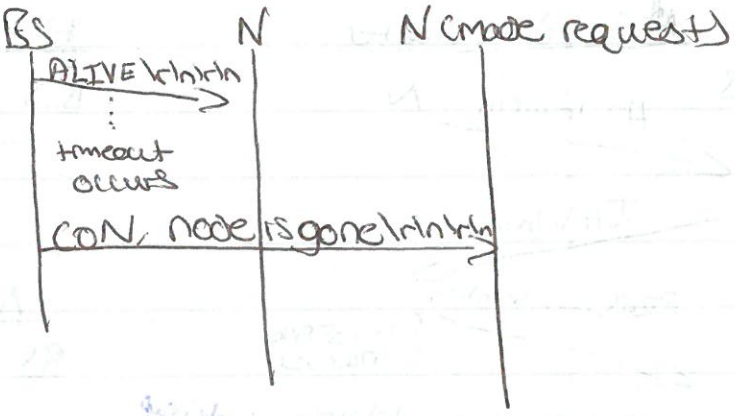
Node deletes file

BS → N: RM, filename \r\n\r\n

Edit File



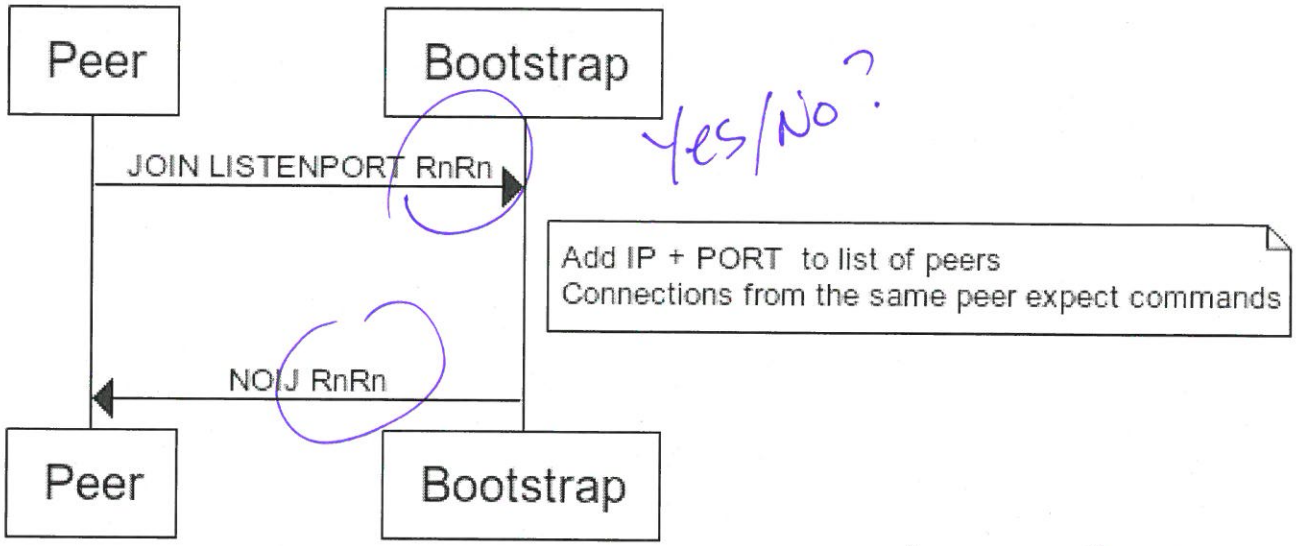
No Response Alive



Think about more error cases once you get a working implementation.

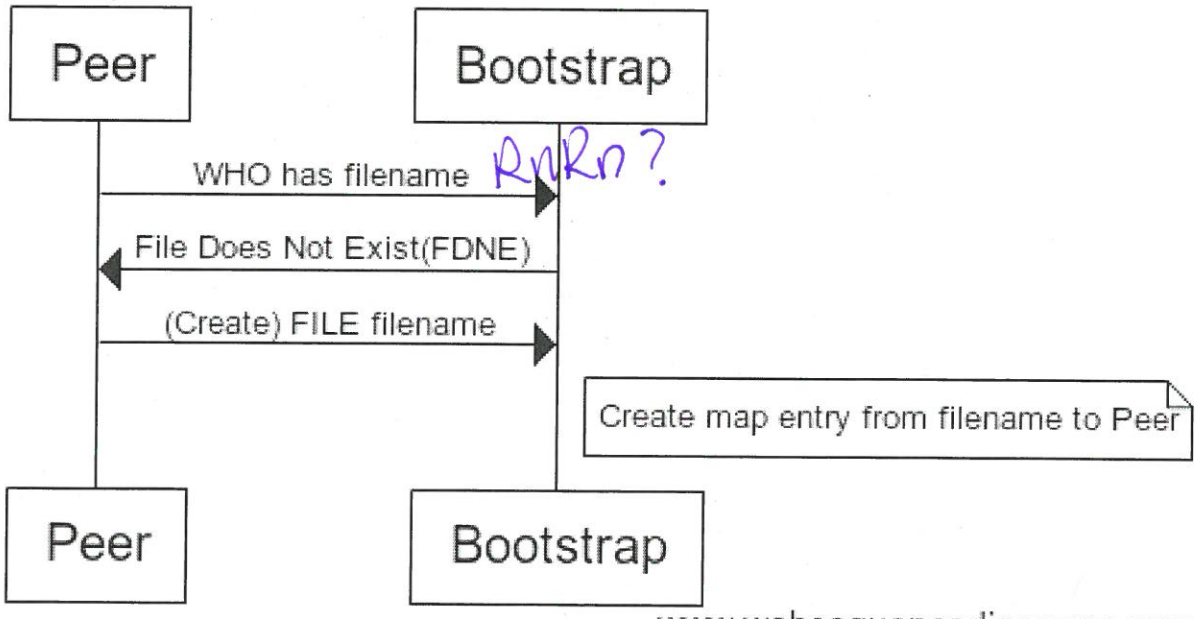
Eniola Abdul
11032426 1

Peer joins network



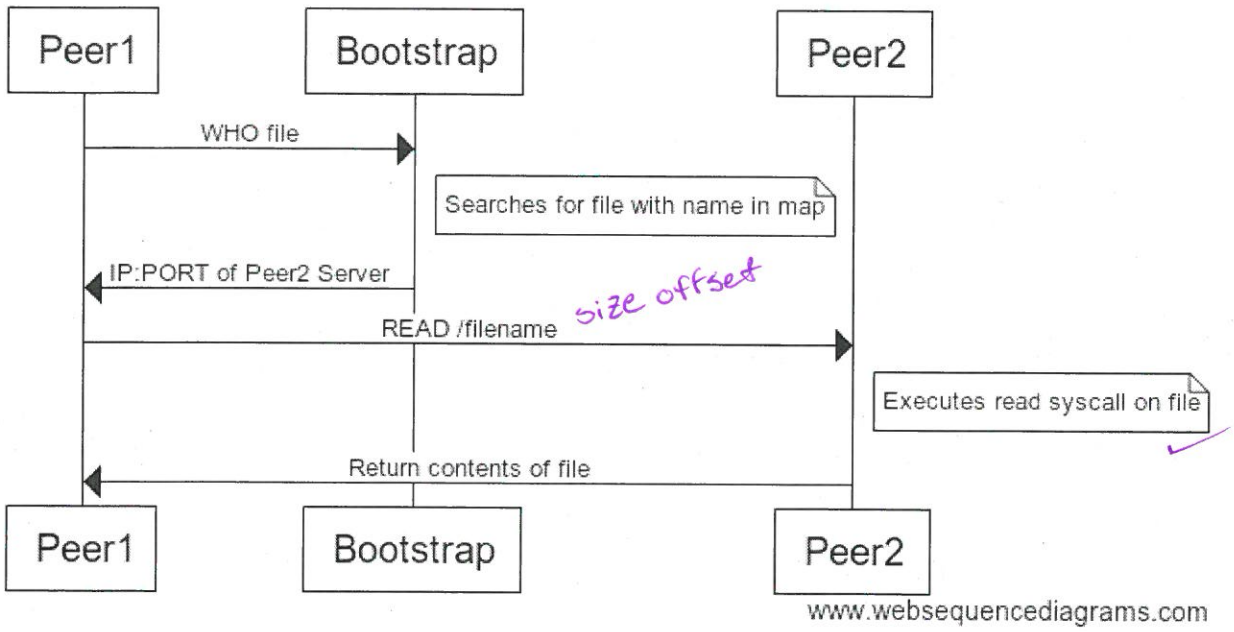
www.websequencediagrams.com

Create a file (touch /filename)

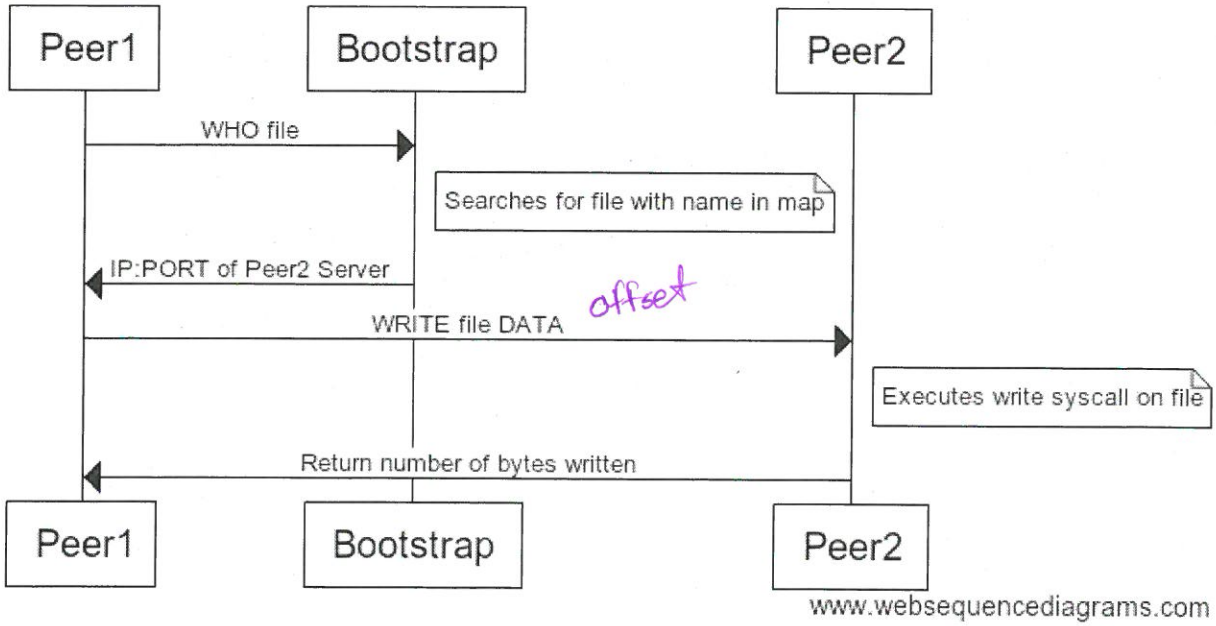


www.websequencediagrams.com

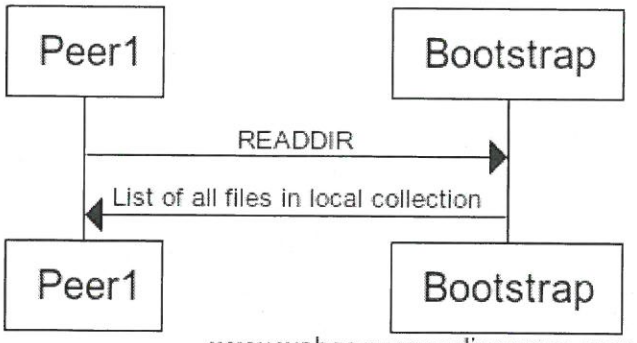
Read a file



Write a file

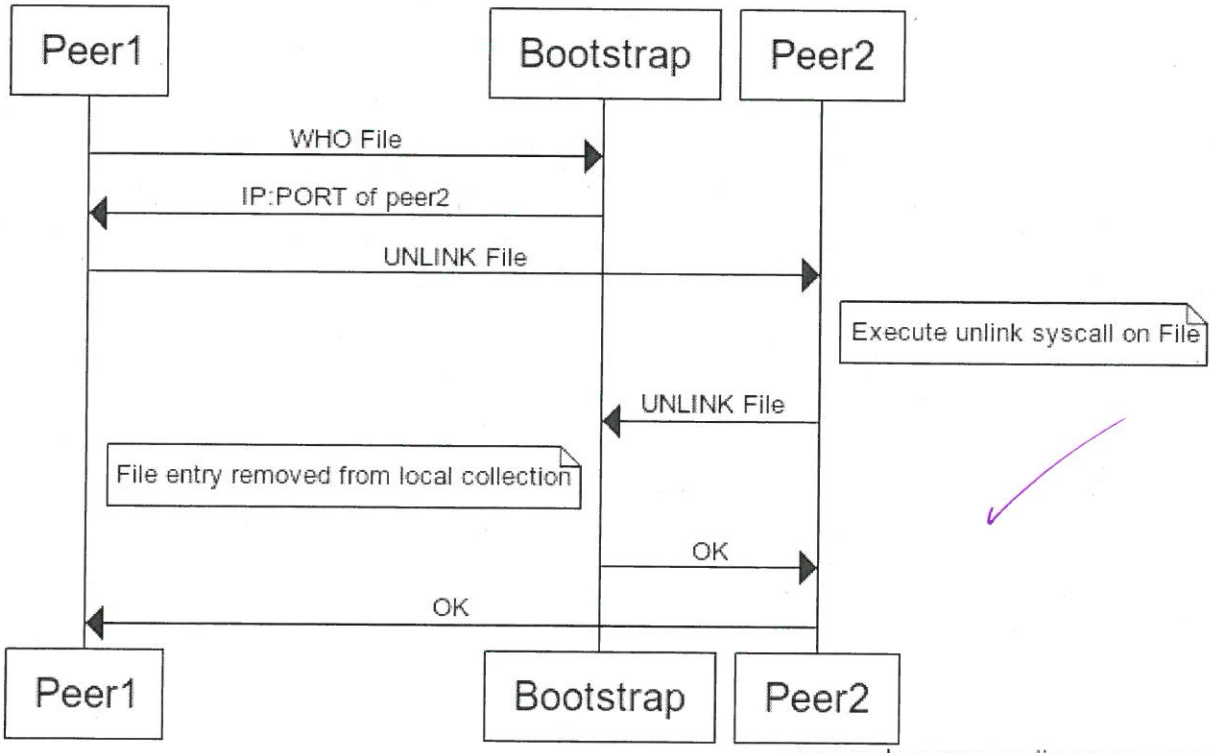


Read/List Directory



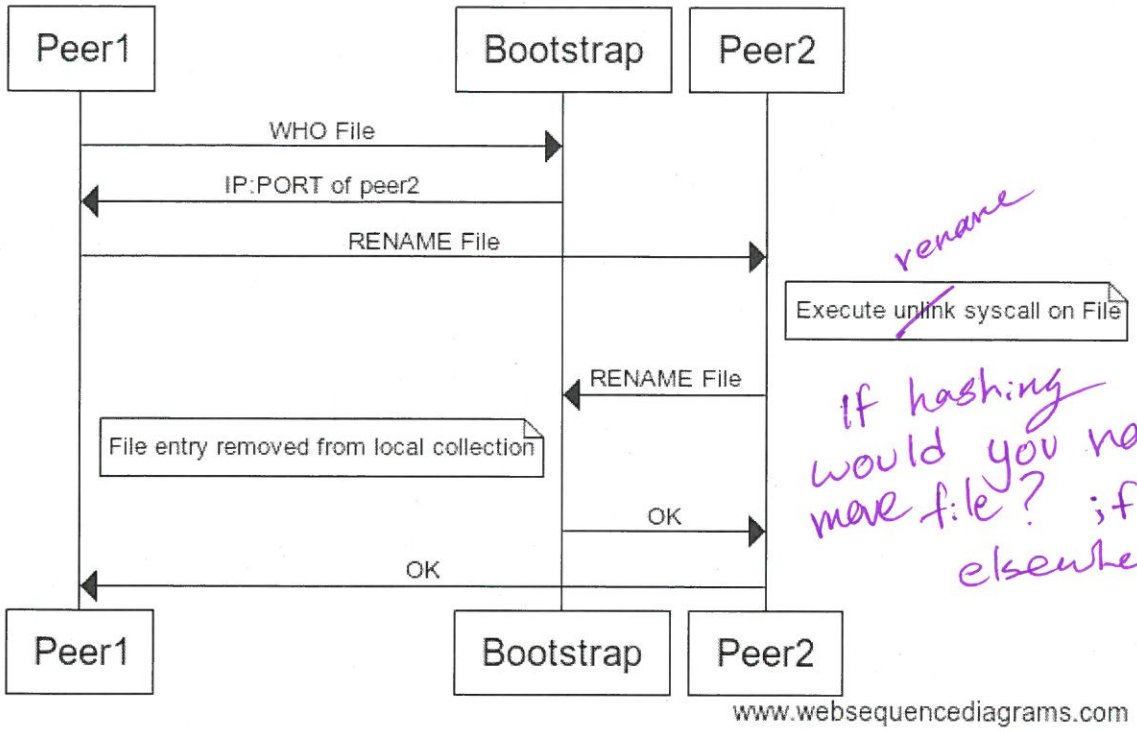
www.websequencediagrams.com

Delete a File



www.websequencediagrams.com

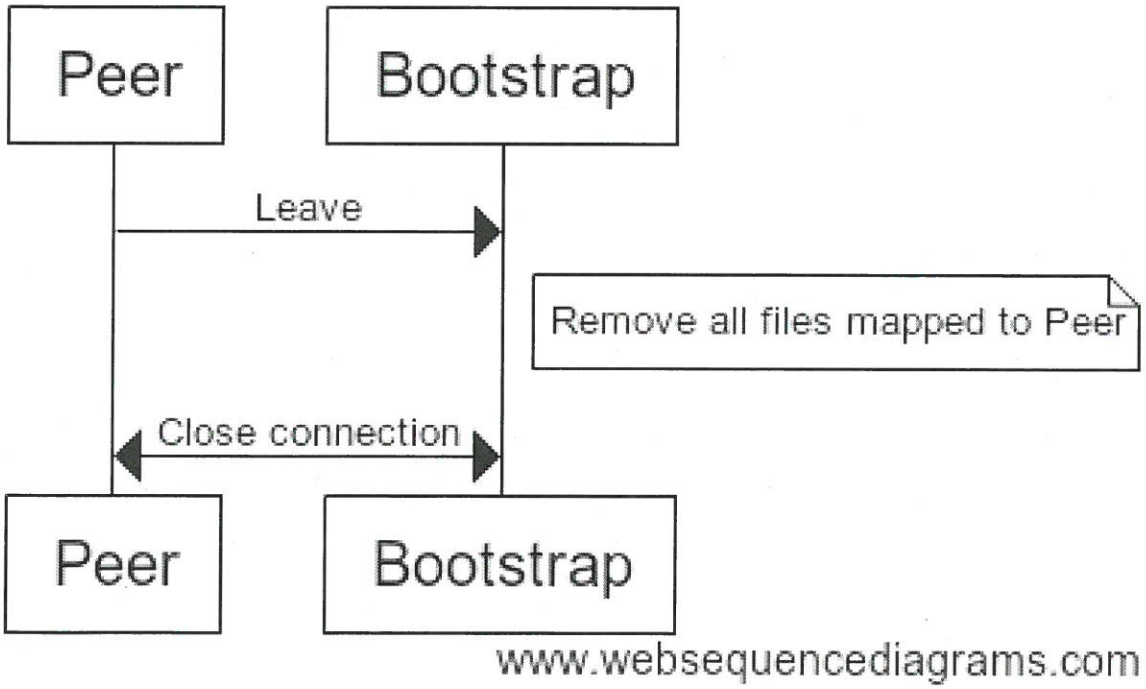
Rename a File



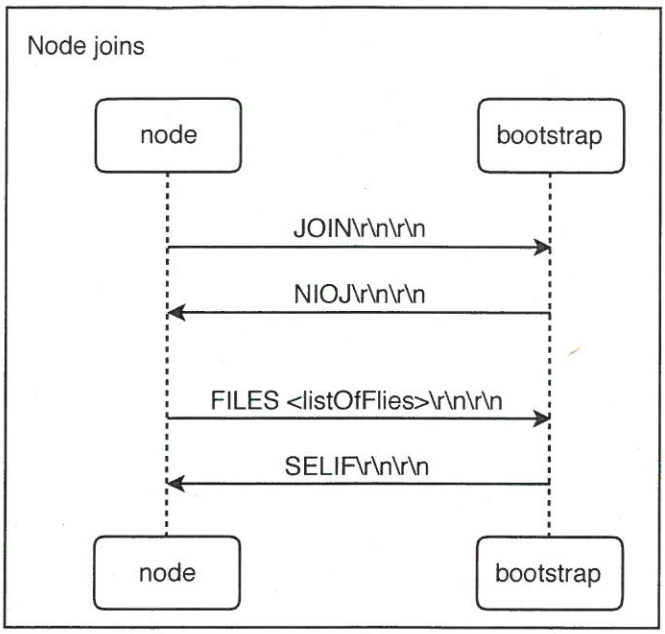
rename

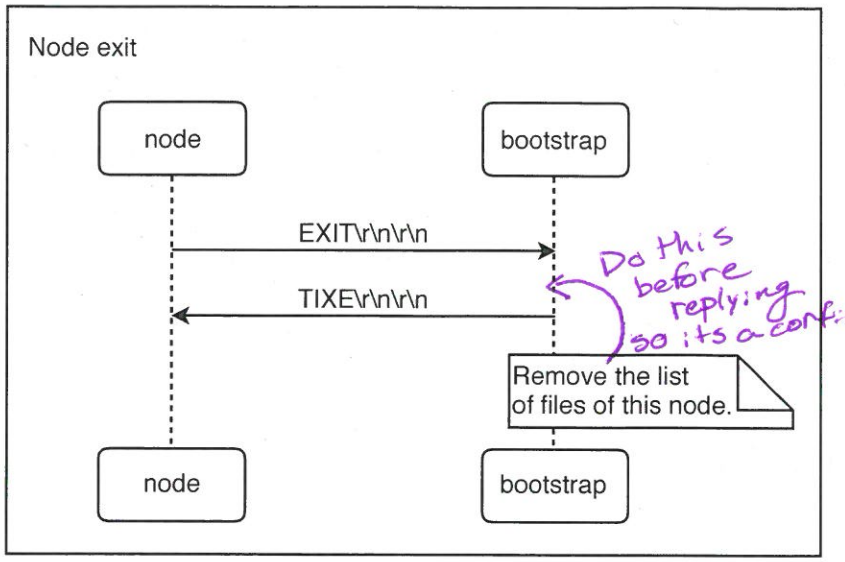
If hashing would you now move file? if hashes elsewhere

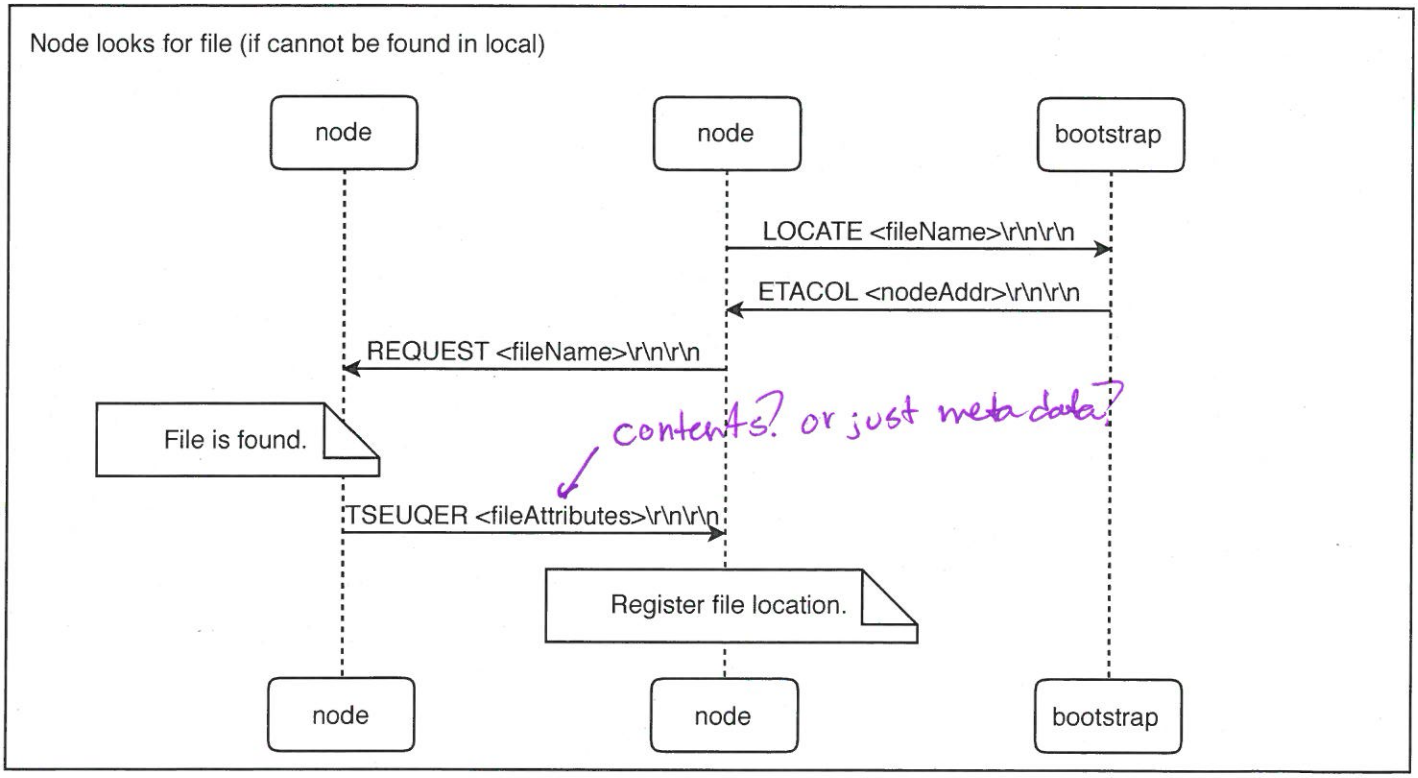
Leave the network

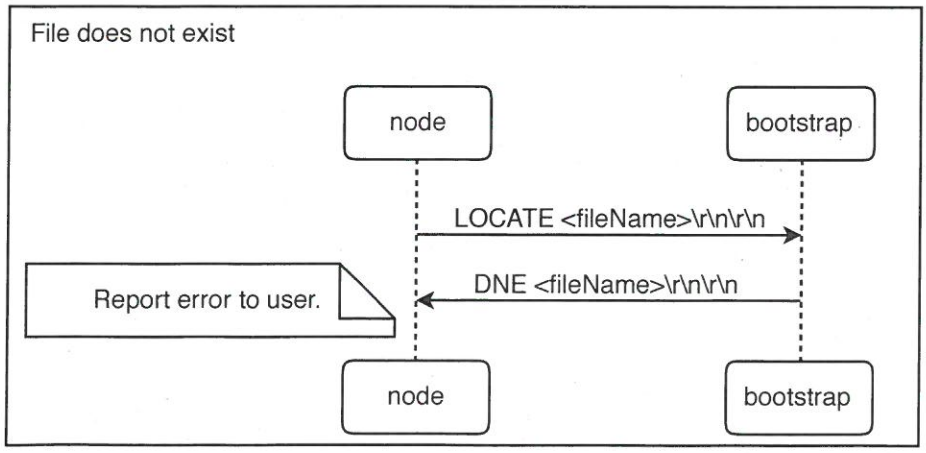


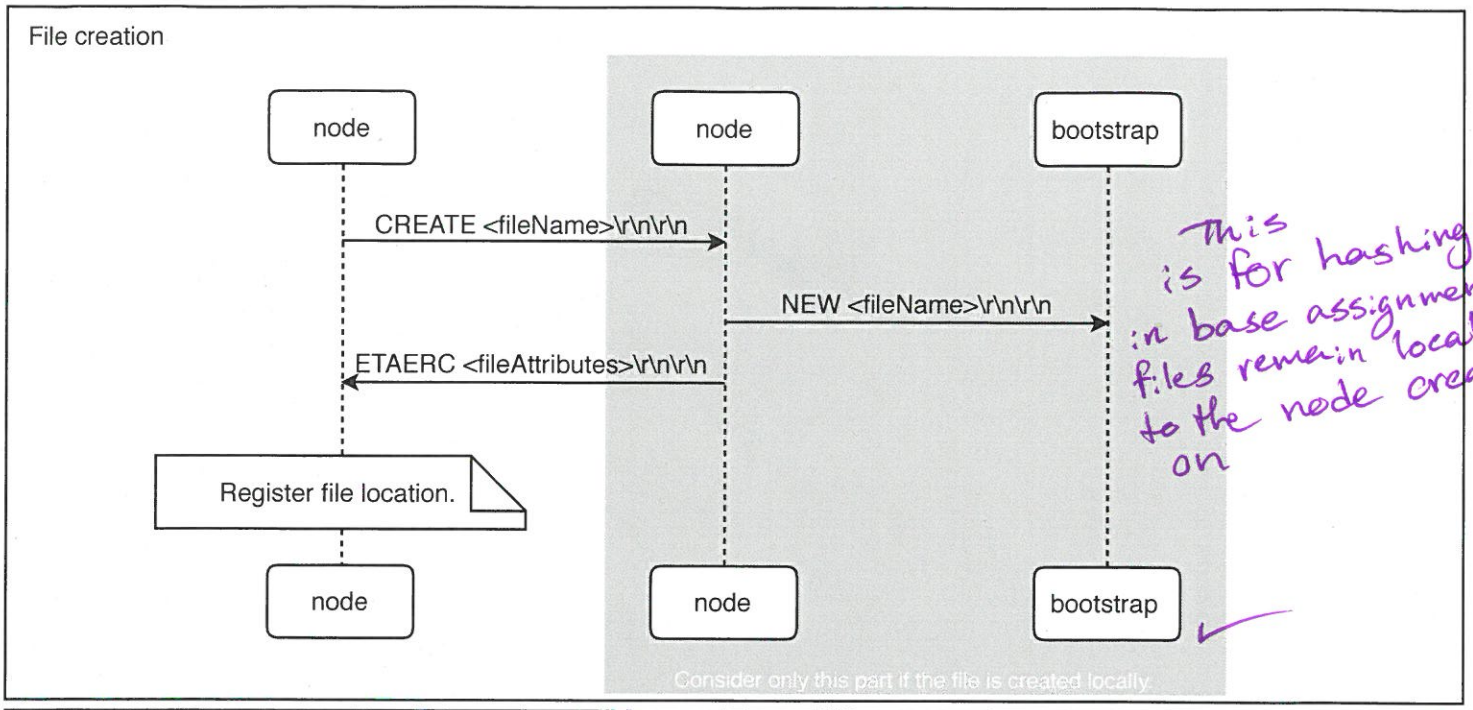
CSE392 hw3 Protocol Design Document
Aritra Nirmal, Tsz Hin Leung

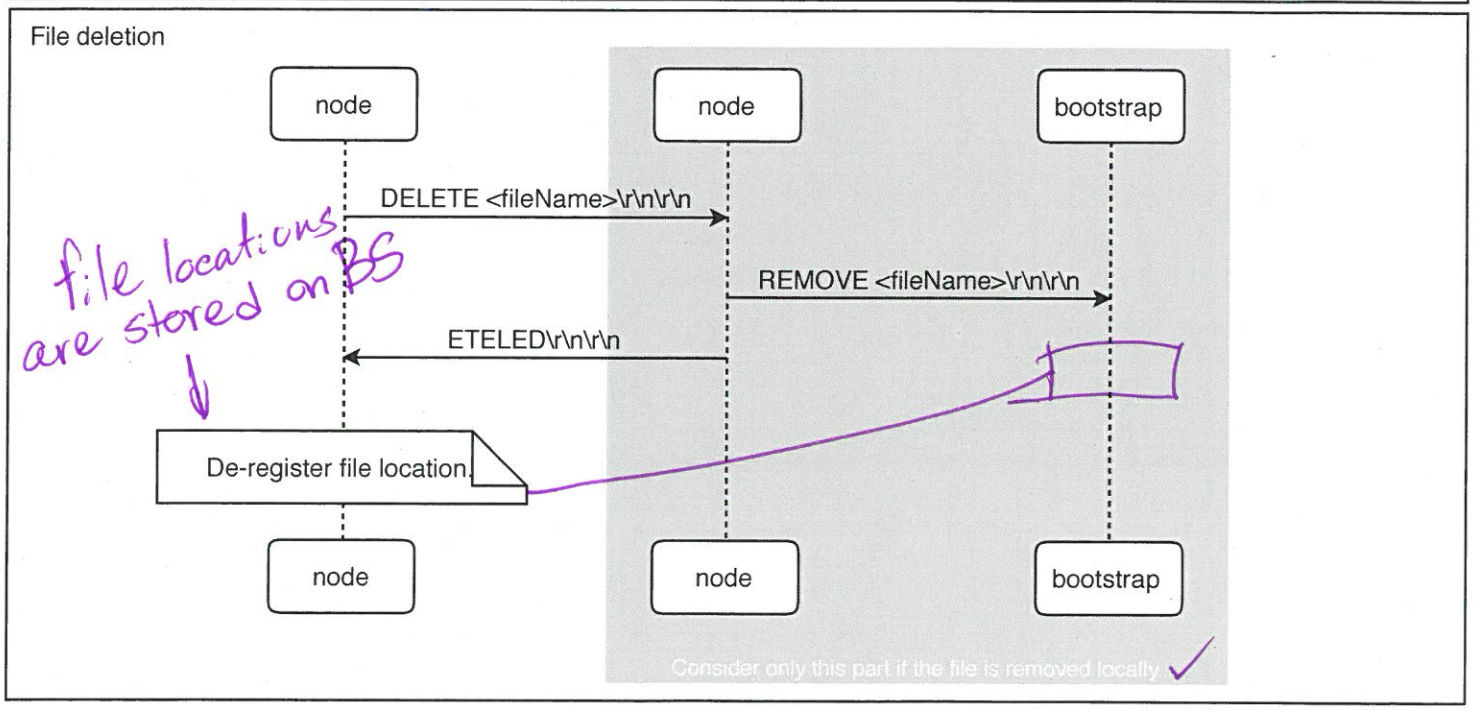


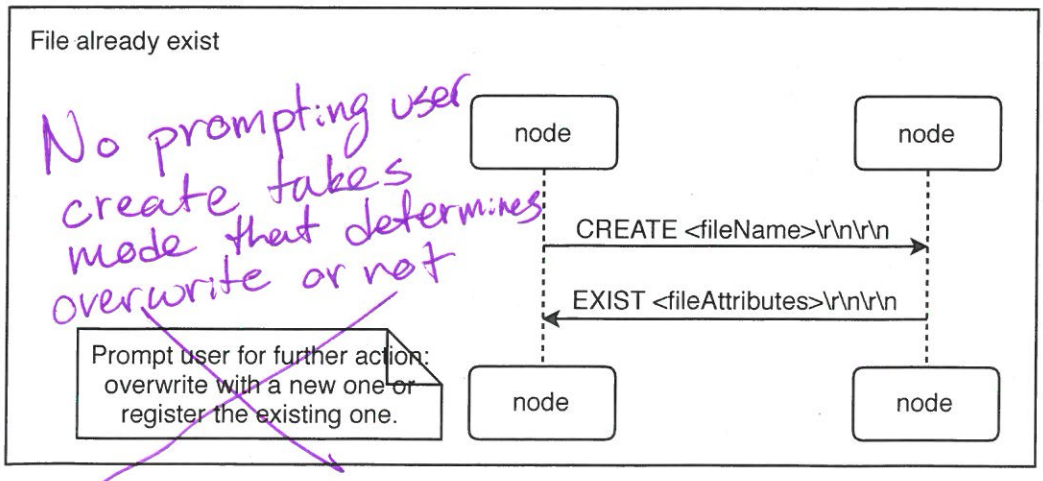


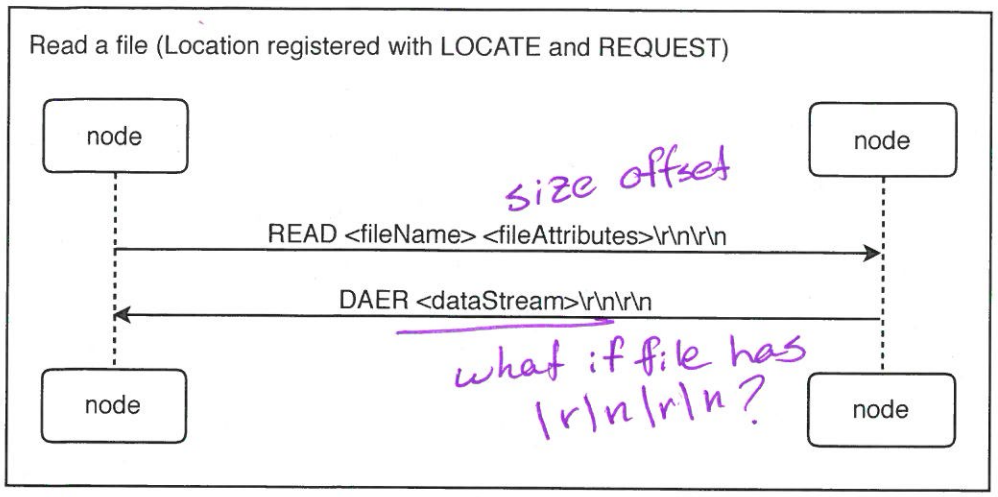




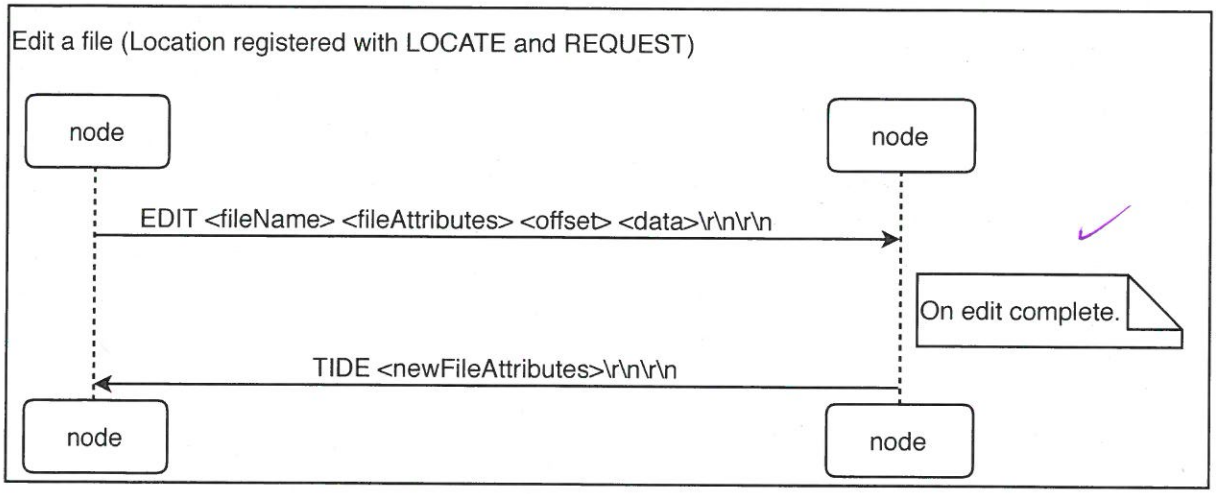

File creation
node
node
bootstrap
CREATE <fileName>\r\n\r\n
NEW <fileName>\r\n\r\n
ETAERC <fileAttributes>\r\n\r\n
Register file location.
node
node
bootstrap
Consider only this part if the file is created locally.
This is for hashing part in base assignment files remain local to the node created on

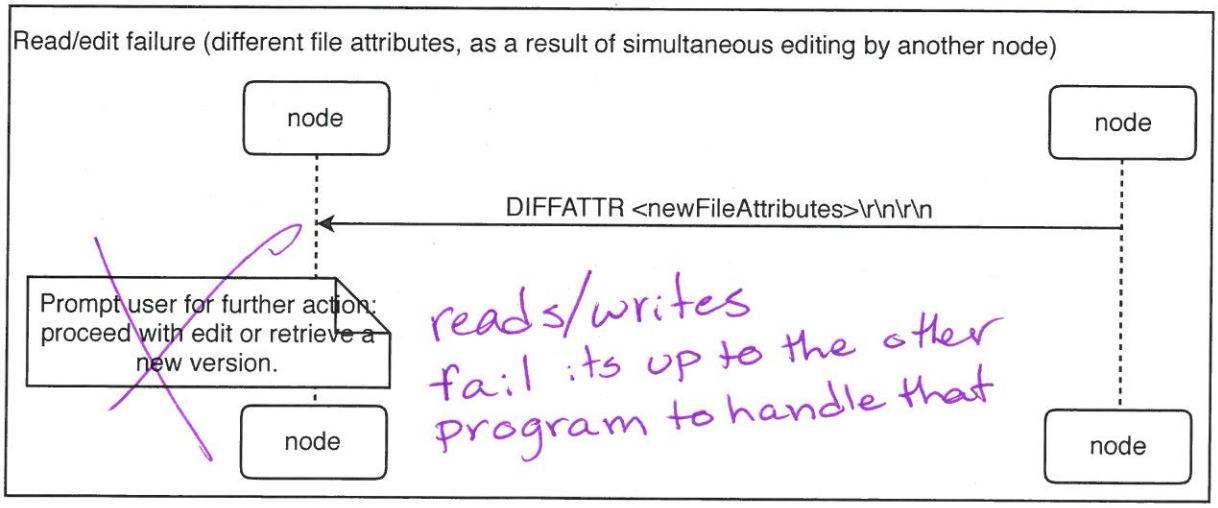

File deletion
node
node
bootstrap
DELETE <fileName>\r\n\r\n
REMOVE <fileName>\r\n\r\n
ETELED\r\n\r\n
file locations are stored on BS
De-register file location.
node
node
bootstrap
Consider only this part if the file is removed locally.


File already exist
No prompting user create takes mode that determines overwrite or not
node
node
CREATE <fileName>\r\n\r\n
EXIST <fileAttributes>\r\n\r\n
Prompt user for further action: overwrite with a new one or register the existing one.
node
node



You can put lengths in protocol





EIFU
TOMITA
CINDY
WU

OPCODE 1 char = 1 byte = 16 different instructions

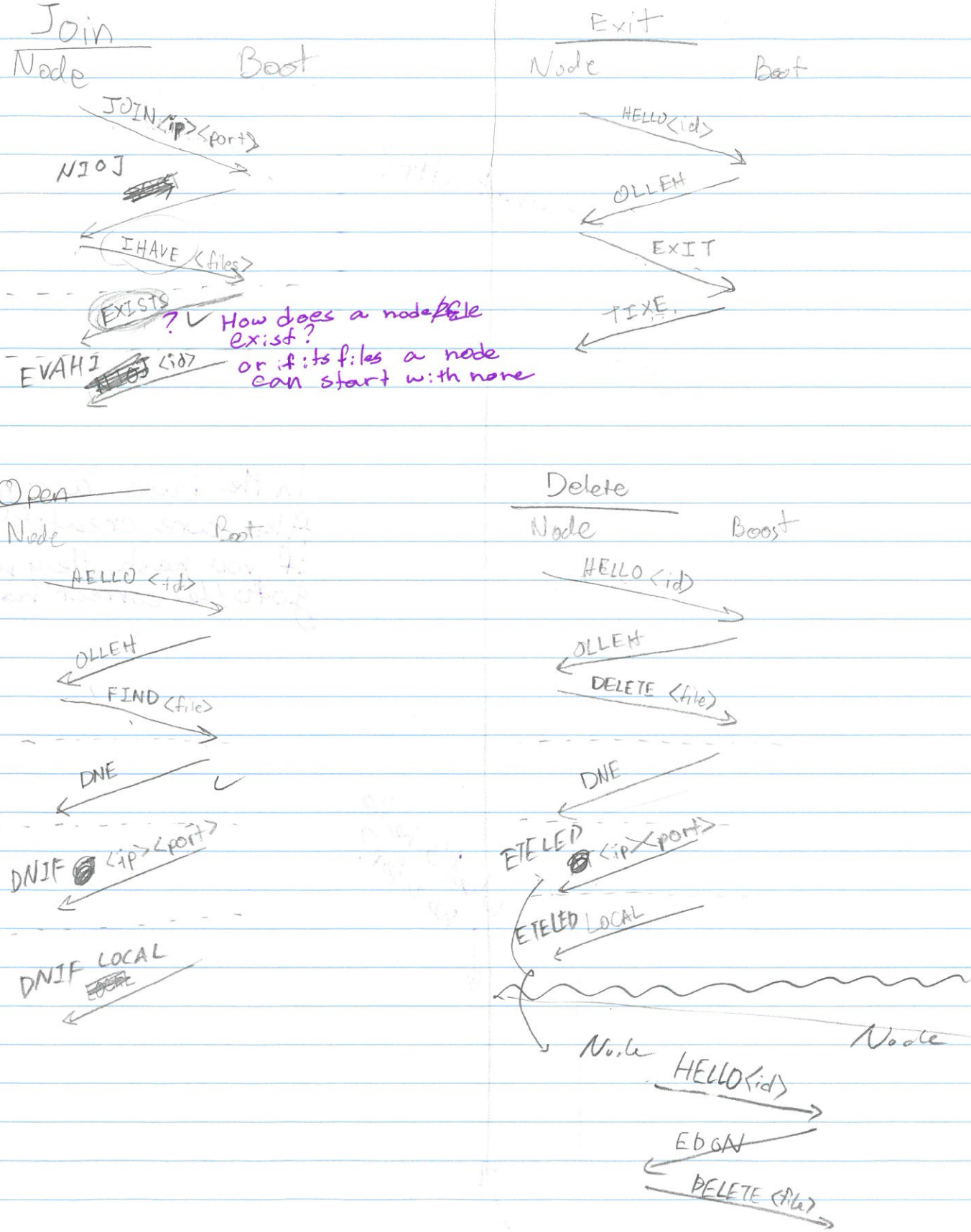
0	JOIN	NIOJ	
1	IHAVE	EVAHI	
2	FIND	DNIF	
3	EXIT	TIXE	
4	DELETE	EIELED	DELETE
5	ATTR	RTTA	
6	MAKE	EKAM	
7			
8			
9			
10			
11	KILL <node_id>		
12	HELLO		HELLO
13	ERROR <msg>		
	from node to booT	booT to node	node to node

HEADER ... I int (4bytes) \r\n\r\n

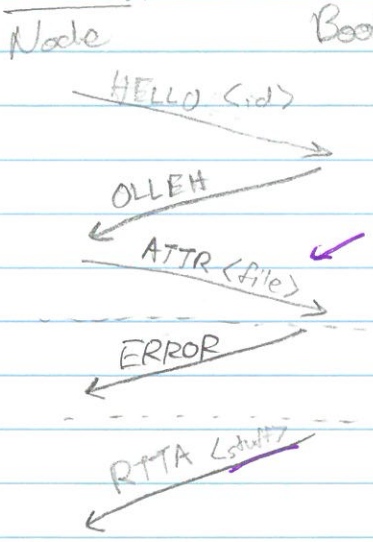
PACKETSIZE	OPCODE

↳ PACKET SIZE ... always multiple of 16

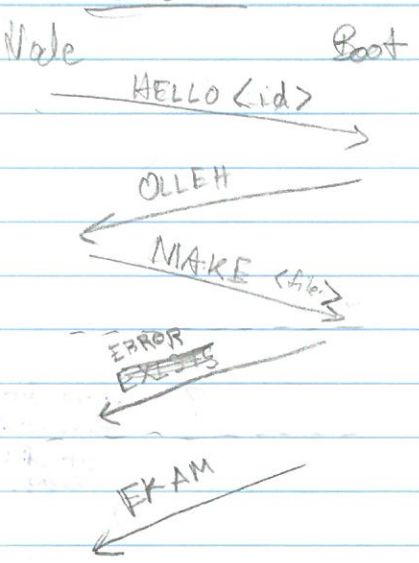
ooo... alignment


Join
Node
Boot
JOIN <ip> <port>
NIOJ
IHAVE <files>
EXISTS
?
How does a node/file exist?
or if its files a node can start with none
EVAHI <id>
Exit
Node
Boot
HELLO <id>
OLLEH
EXIT
TIXE
Open
Node
Boot
AELLO <id>
OLLEH
FIND <file>
DNE
DNIF <ip> <port>
DNIF LOCAL
Delete
Node
Boost
HELLO <id>
OLLEH
DELETE <file>
DNE
ETELED <ip> <port>
ETELED LOCAL
Node
Node
HELLO <id>
EDOM
DELETE <file>

Get attri

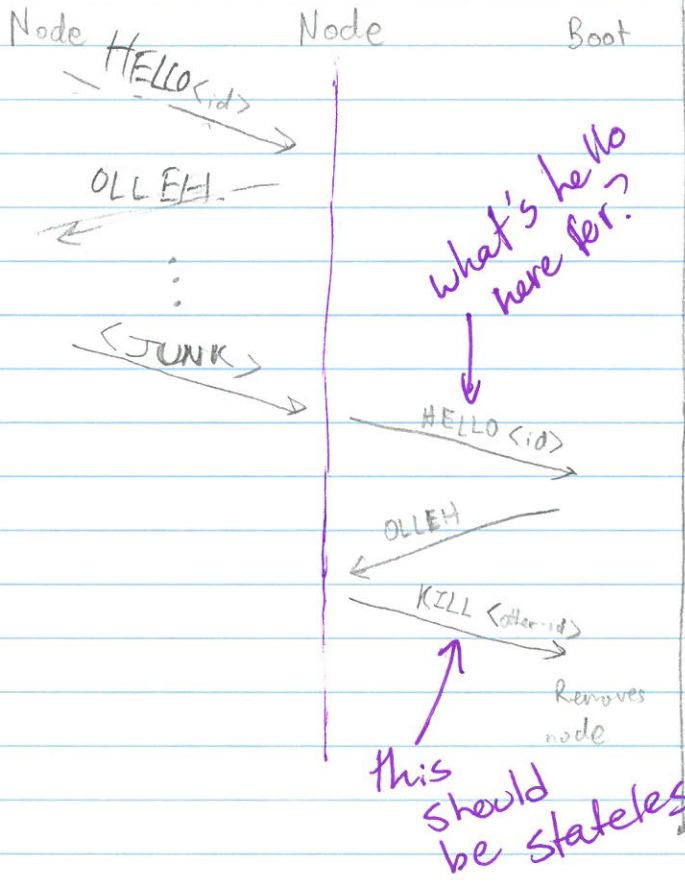


Create



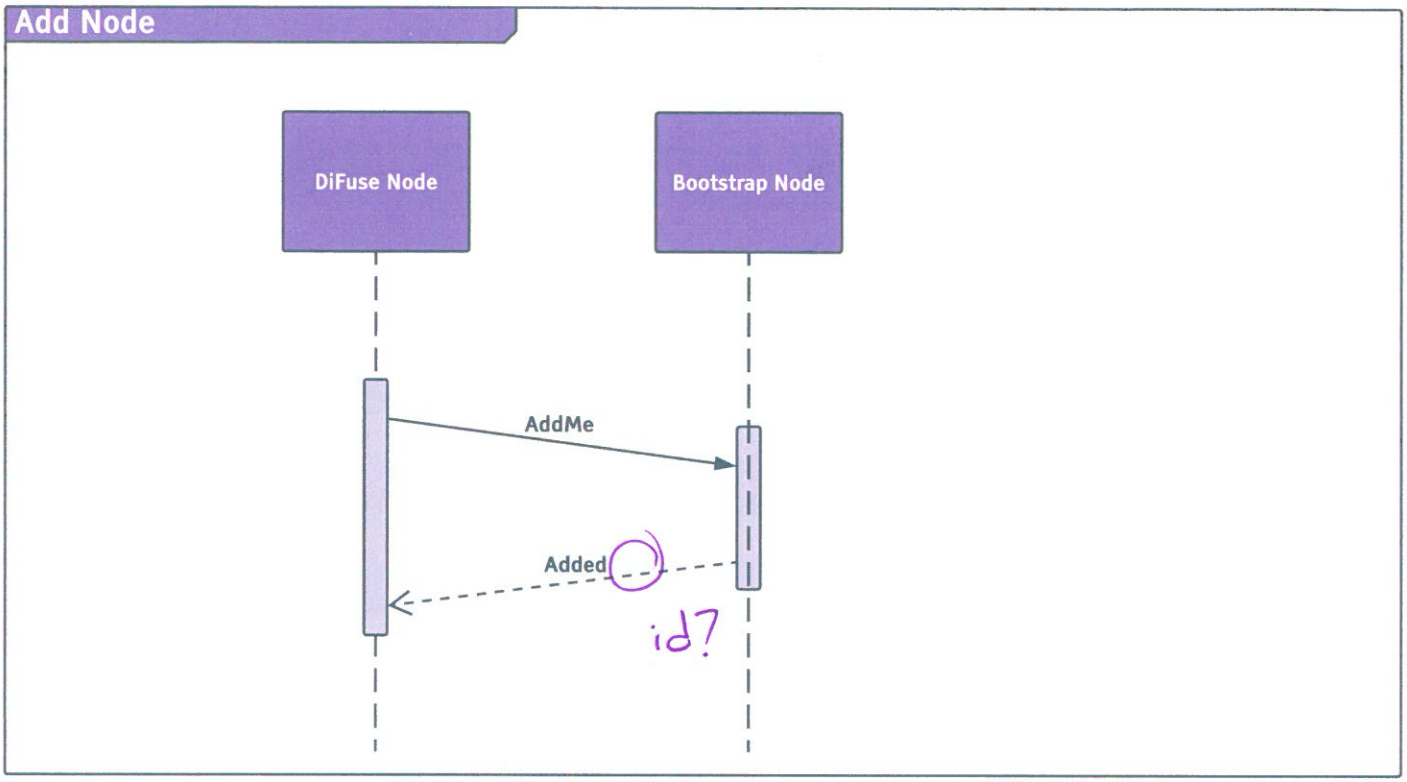
In the base assignment files are created locally if you hash, they need to go to the correct node

Junk Node - Node

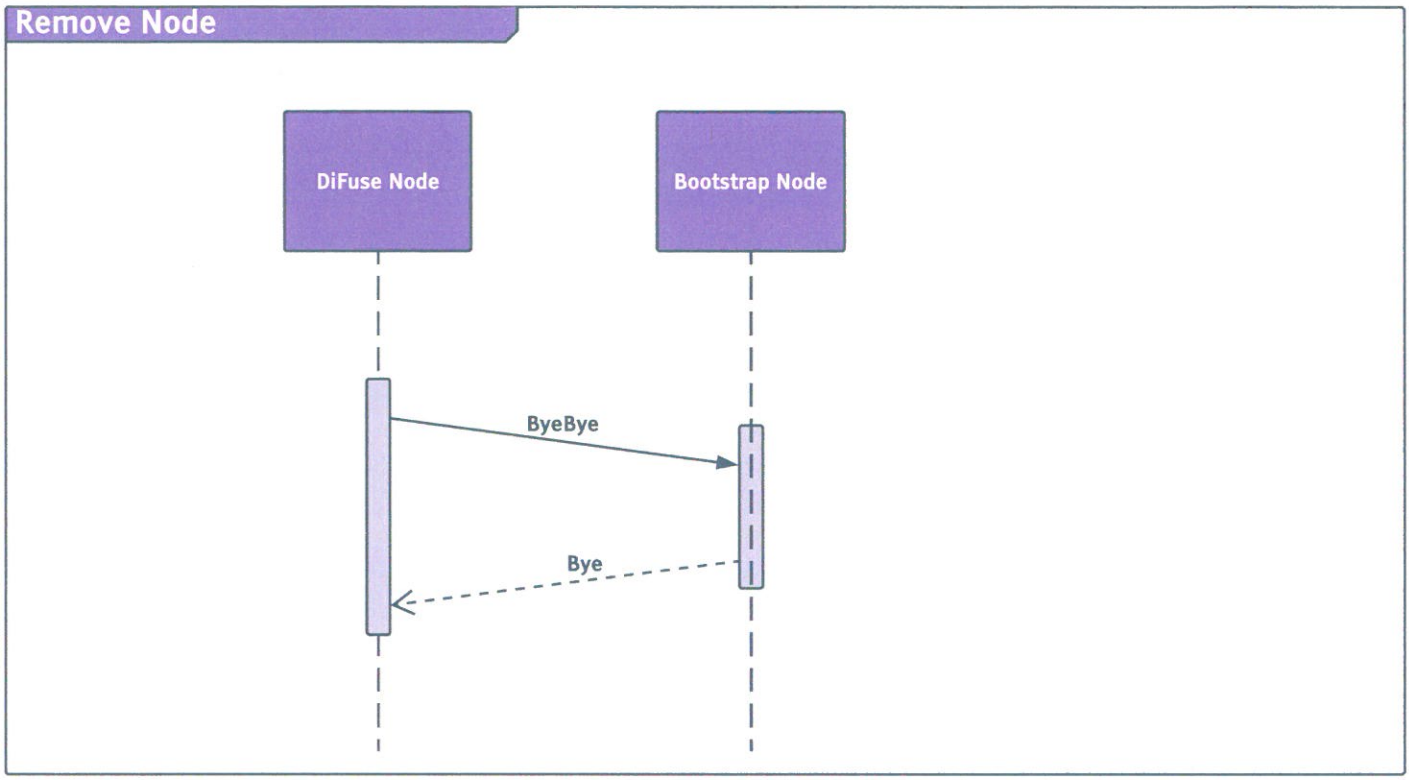


Junk Node - Boot

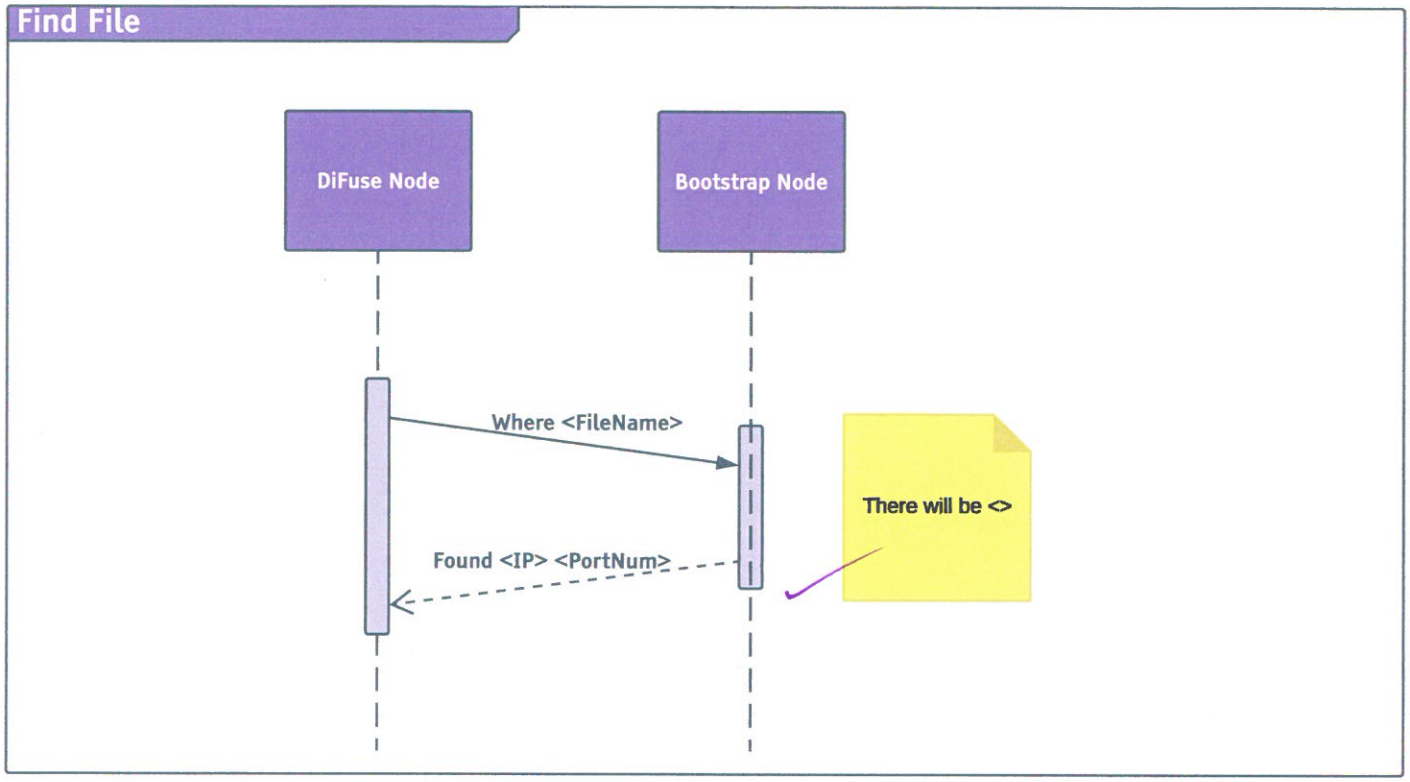
Stefan Imrit 1089961163

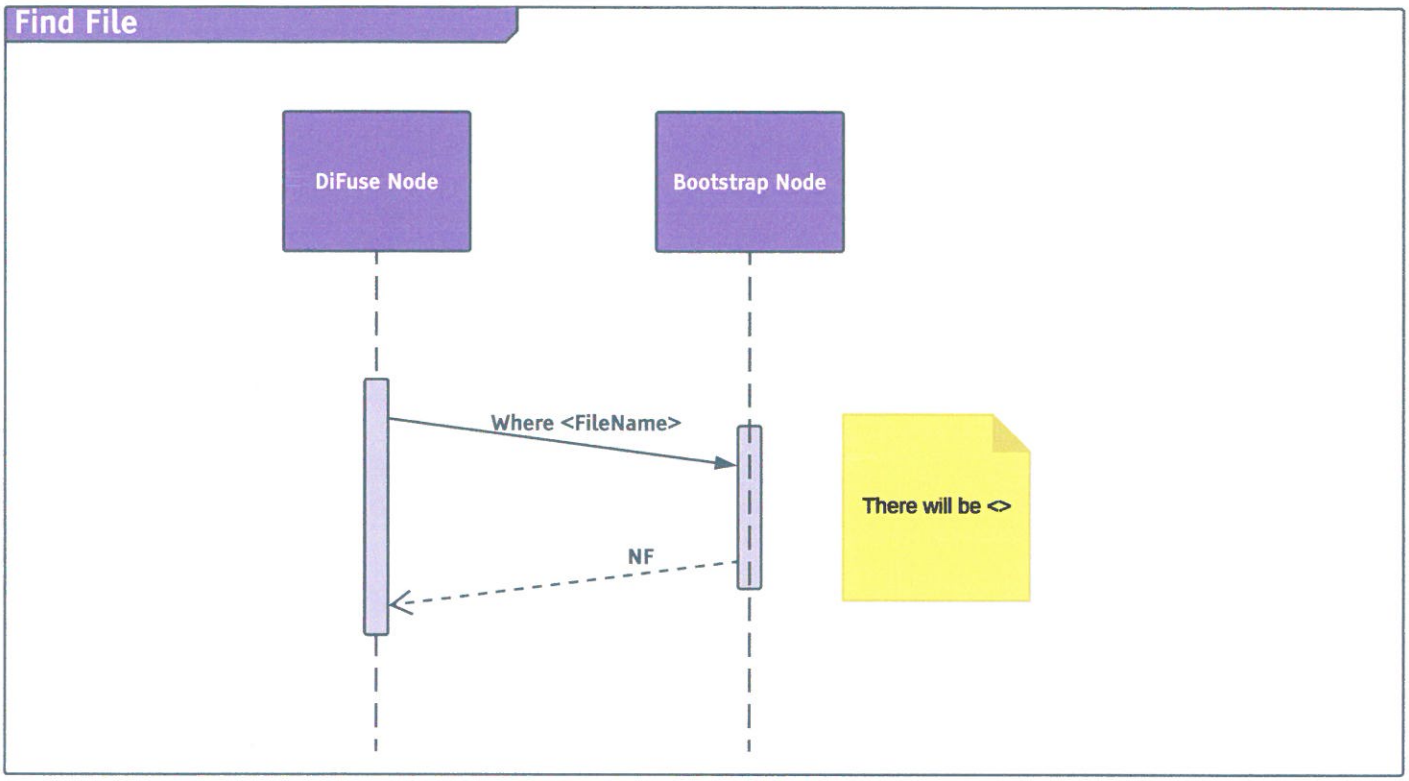


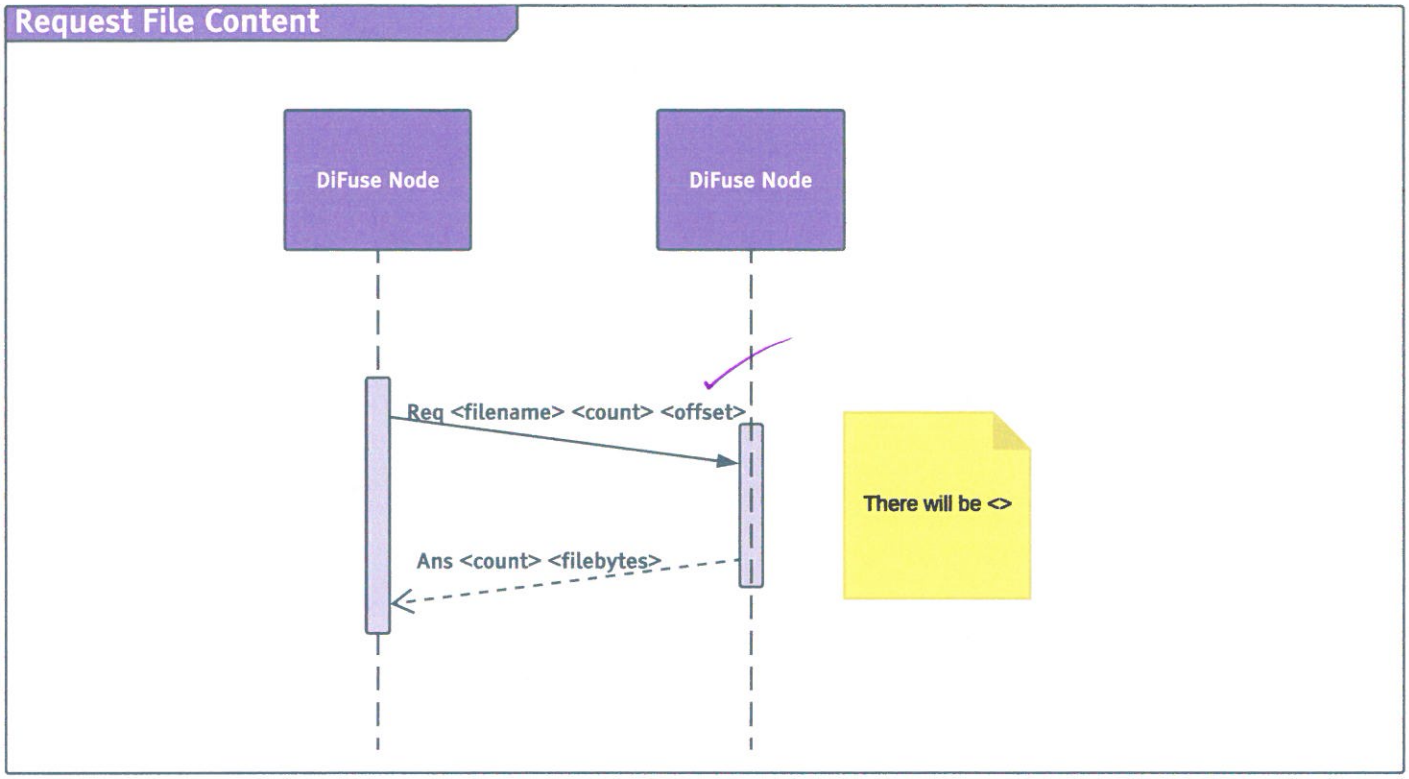
Stefan Inrit

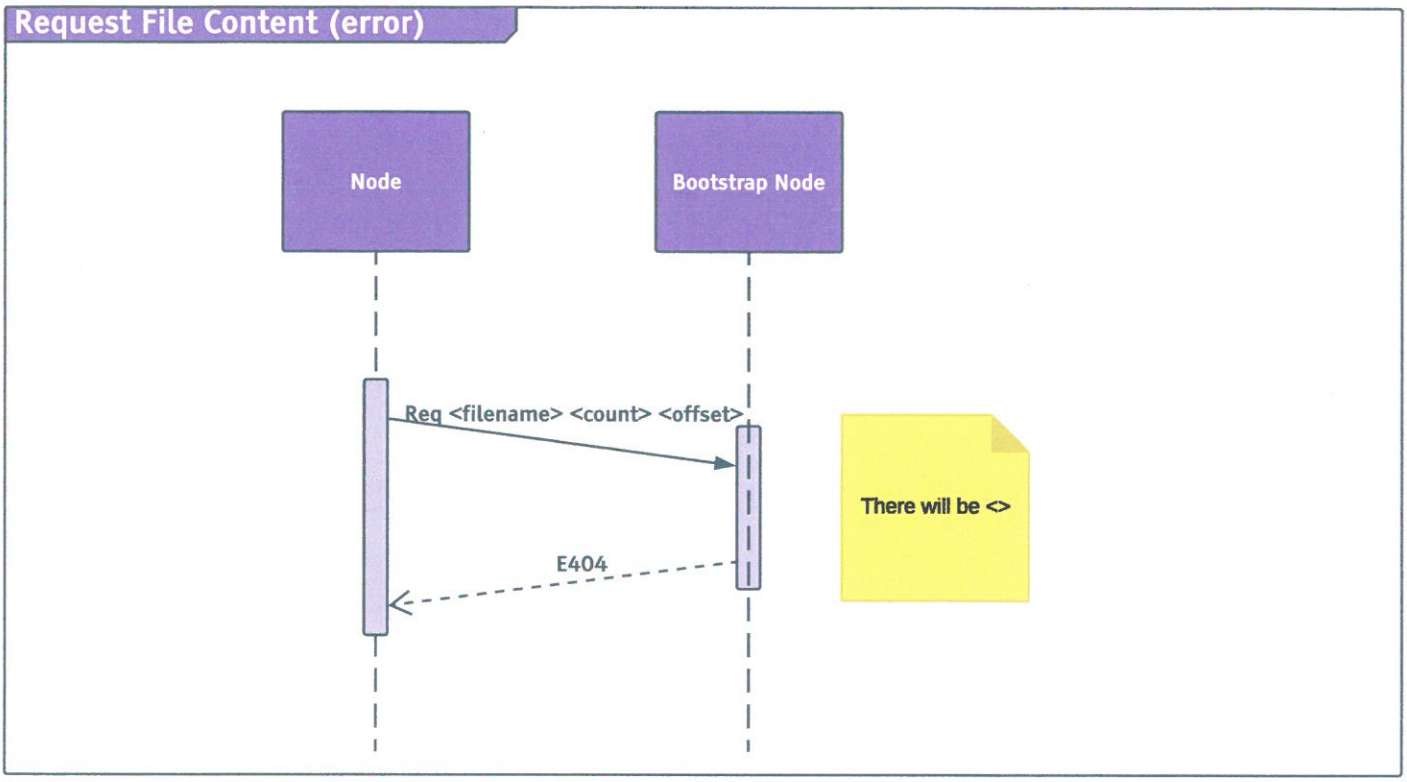


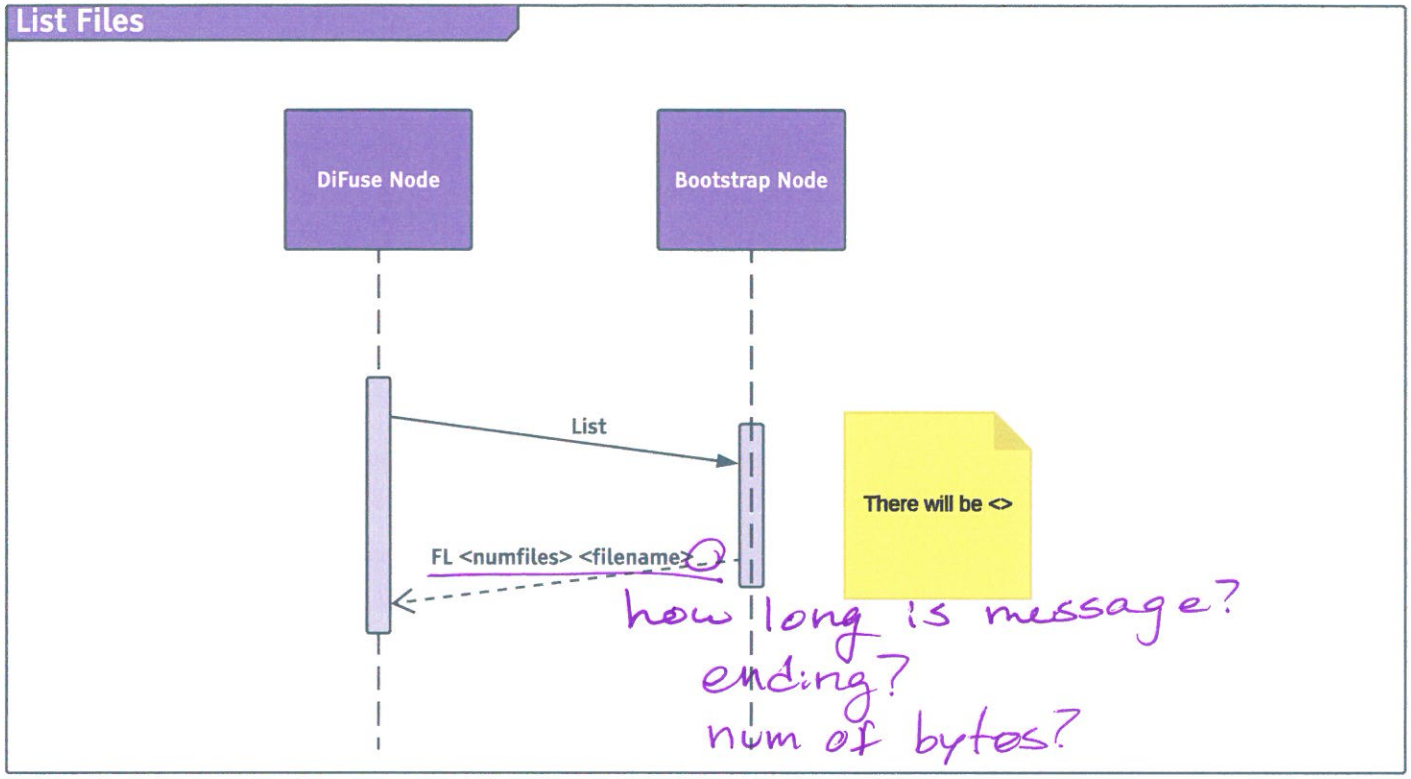
Stefan Irrit

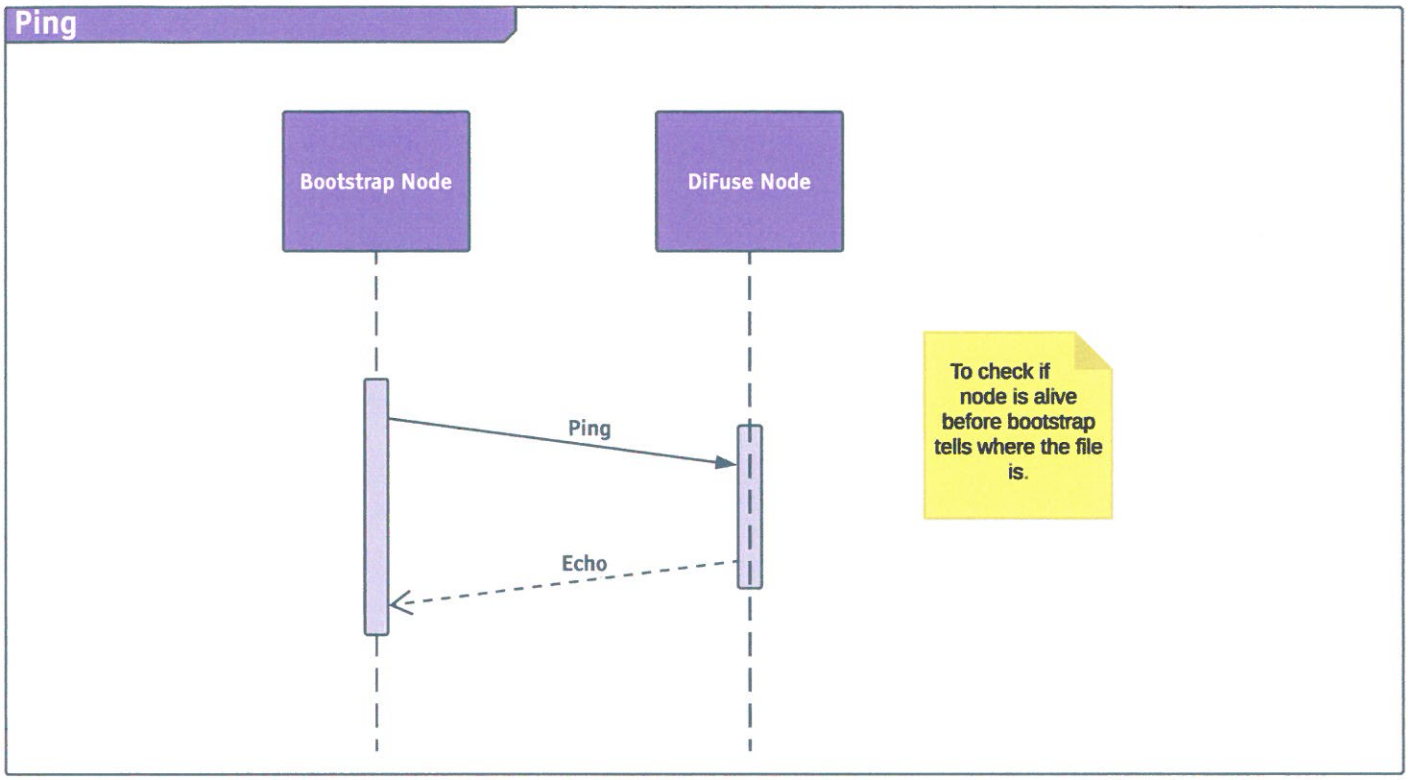



Find File
DiFuse Node
Bootstrap Node
Where <FileName>
NF
There will be <>


Request File Content
DiFuse Node
DiFuse Node
Req <filename> <count> <offset>
There will be <>
Ans <count> <filebytes>

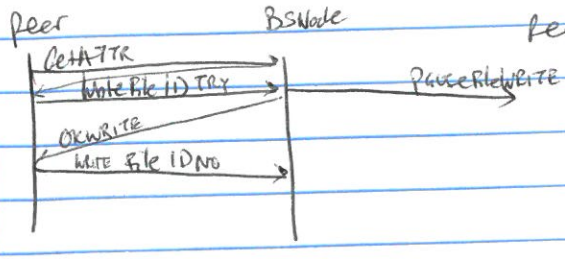

Request File Content (error)
Node
Bootstrap Node
Req <filename> <count> <offset>
There will be <>
E404


List Files
DiFuse Node
Bootstrap Node
List
There will be <>
FL <numfiles> <filename>
how long is message?
ending?
num of bytes?

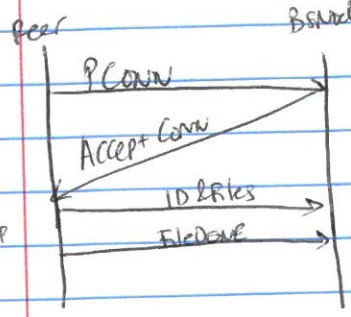

Ping
Bootstrap Node
DiFuse Node
Ping
Echo
To check if node is alive before bootstrap tells where the file is.

Kenny Lee

WRITE

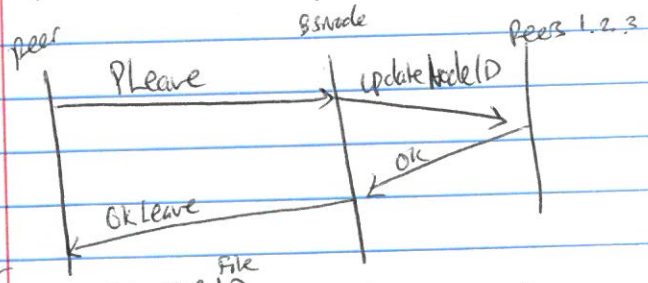


JOIN

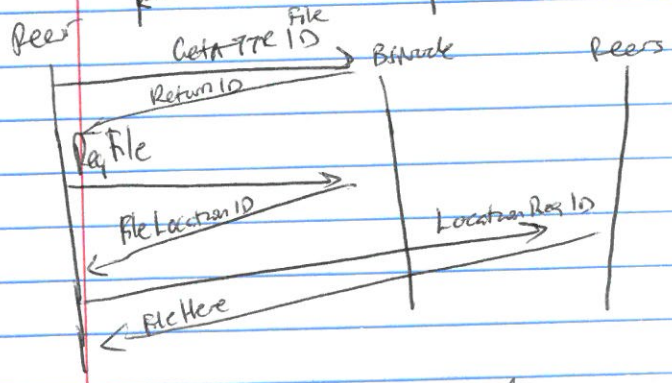


$ID = (IDI)_P$

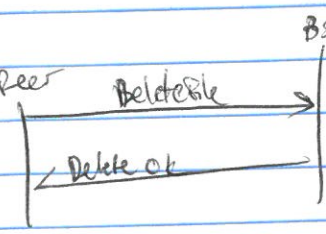
EXIT



ASK
For File

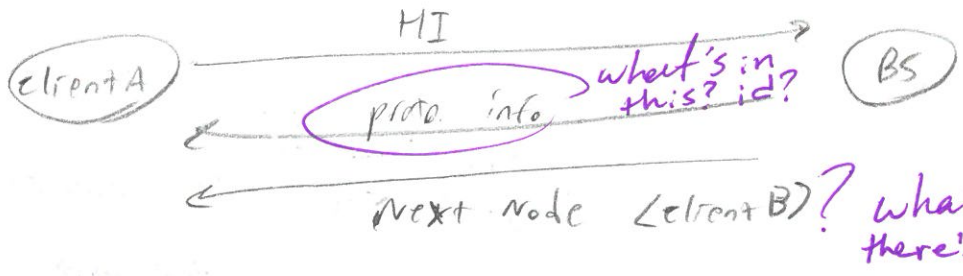


Delete



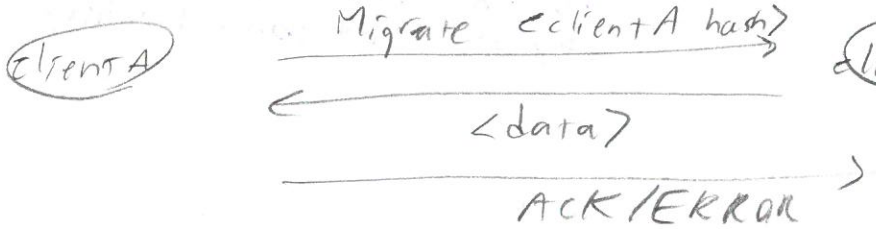
Asher Davidson, Freeman Lor

Handshake



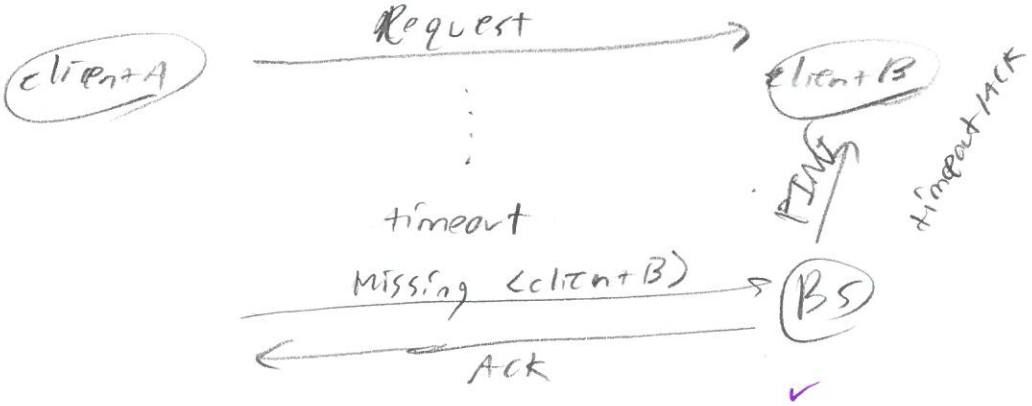
Migrate

Client B migrating its data to A

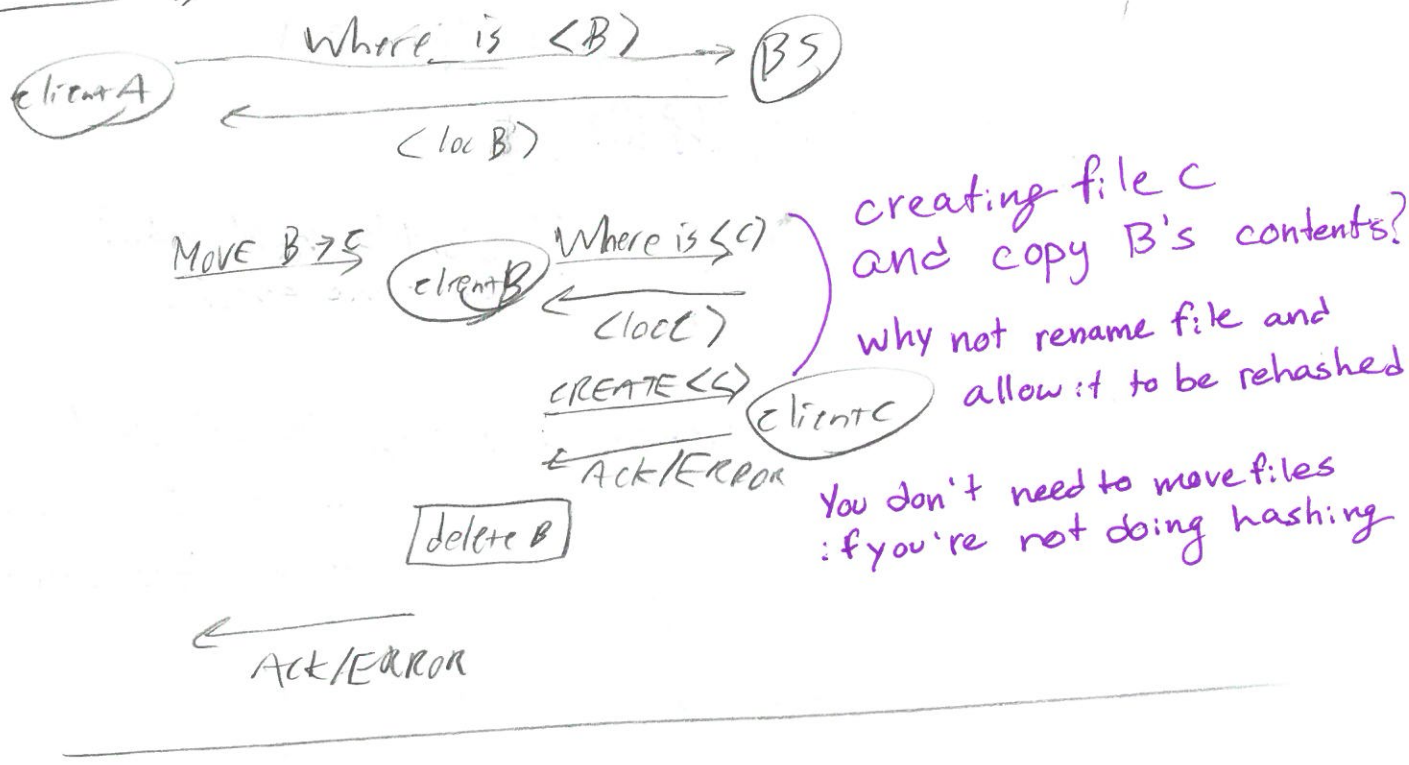


- cus B is leaving
- cus A is joining?

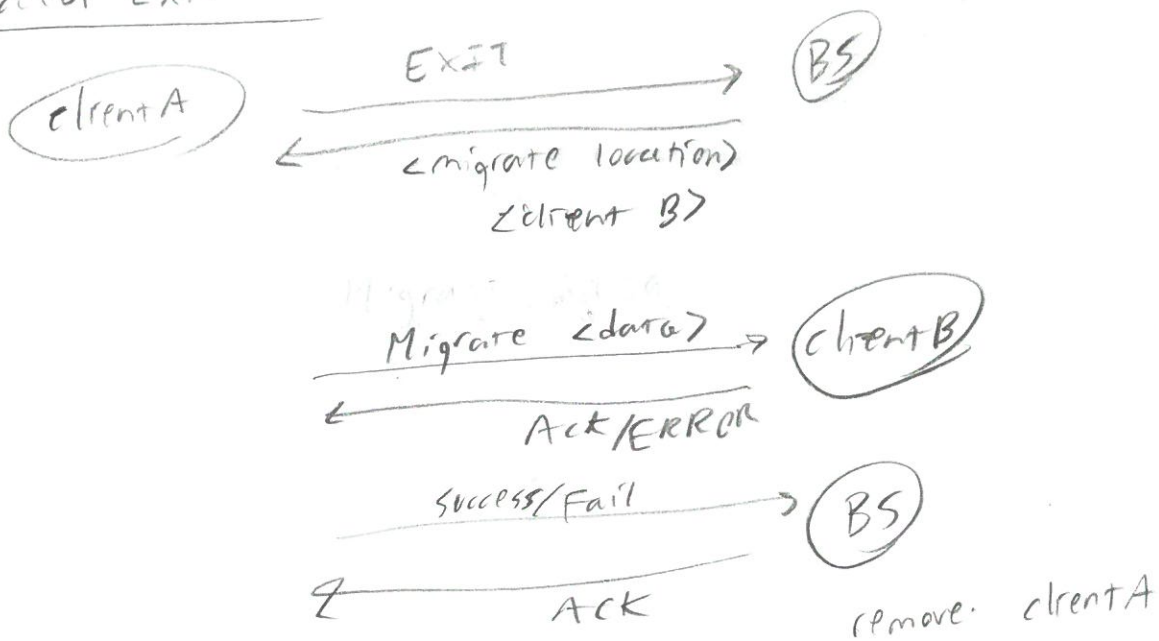
Missing Node



Rename B → C



Graceful Exit

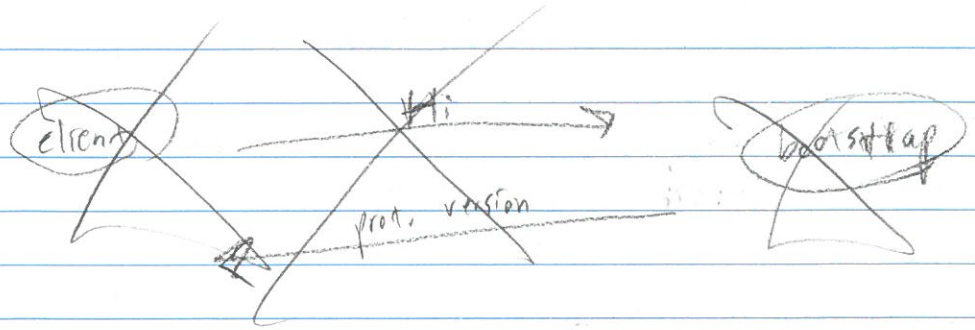


closes

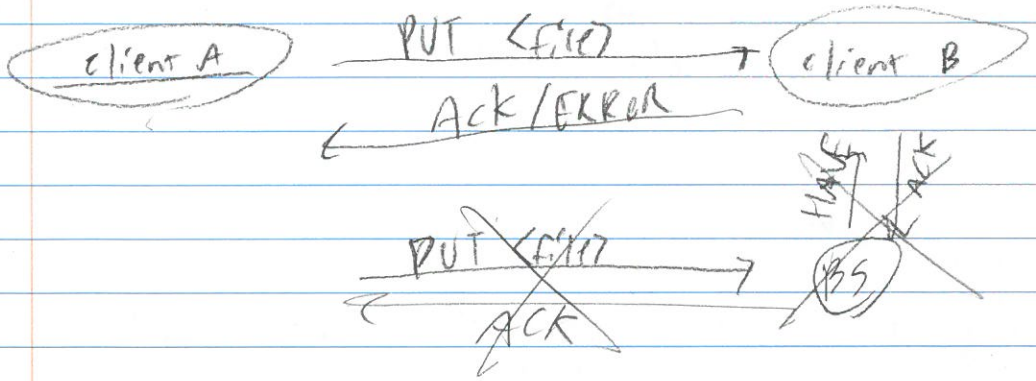
what's the format of your pkts?

Asher Davidson, Freeman Lou

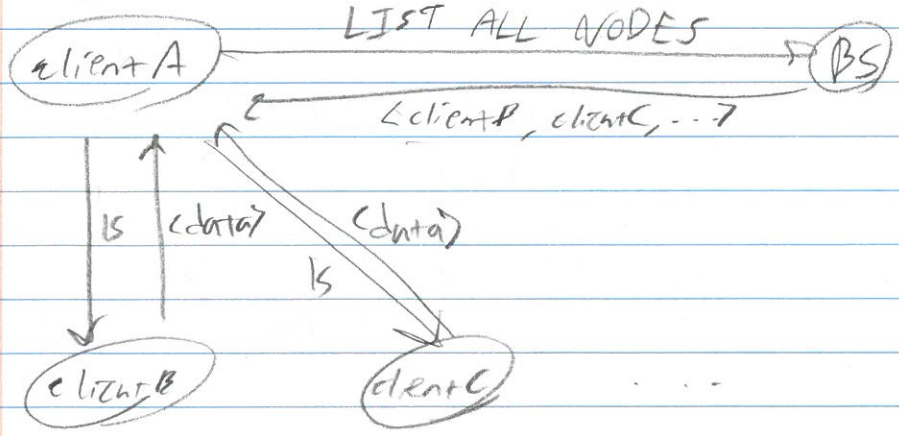
[(hash, address)]



Insertion / Deletion

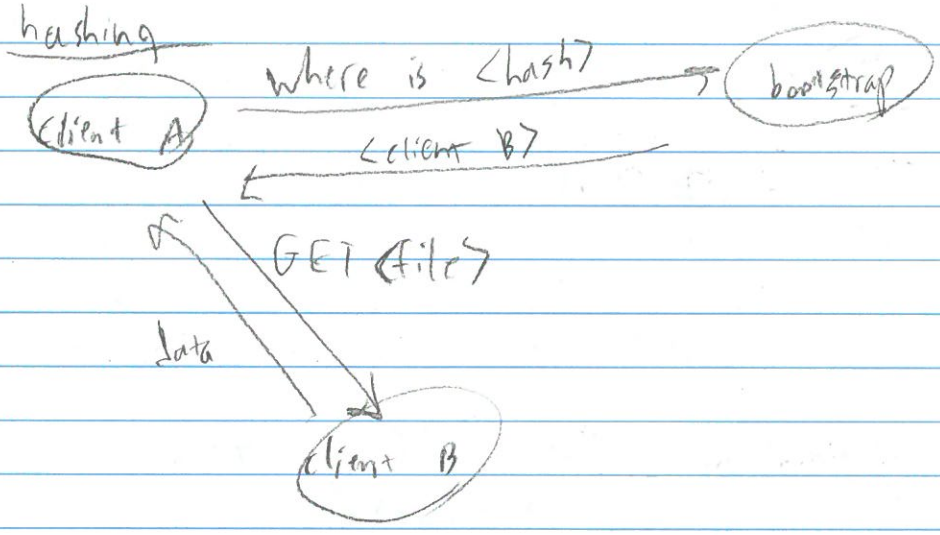


ls

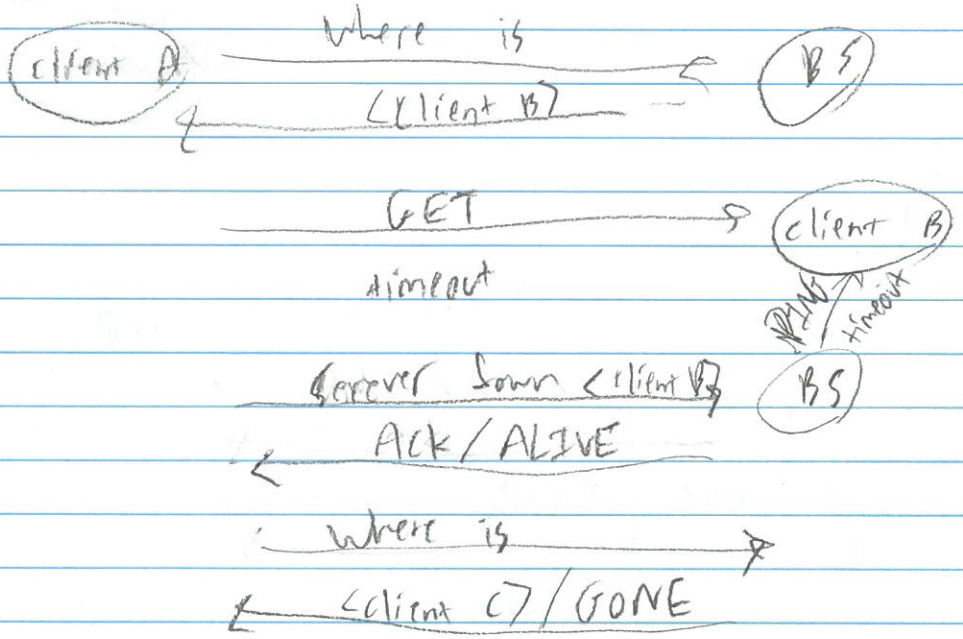


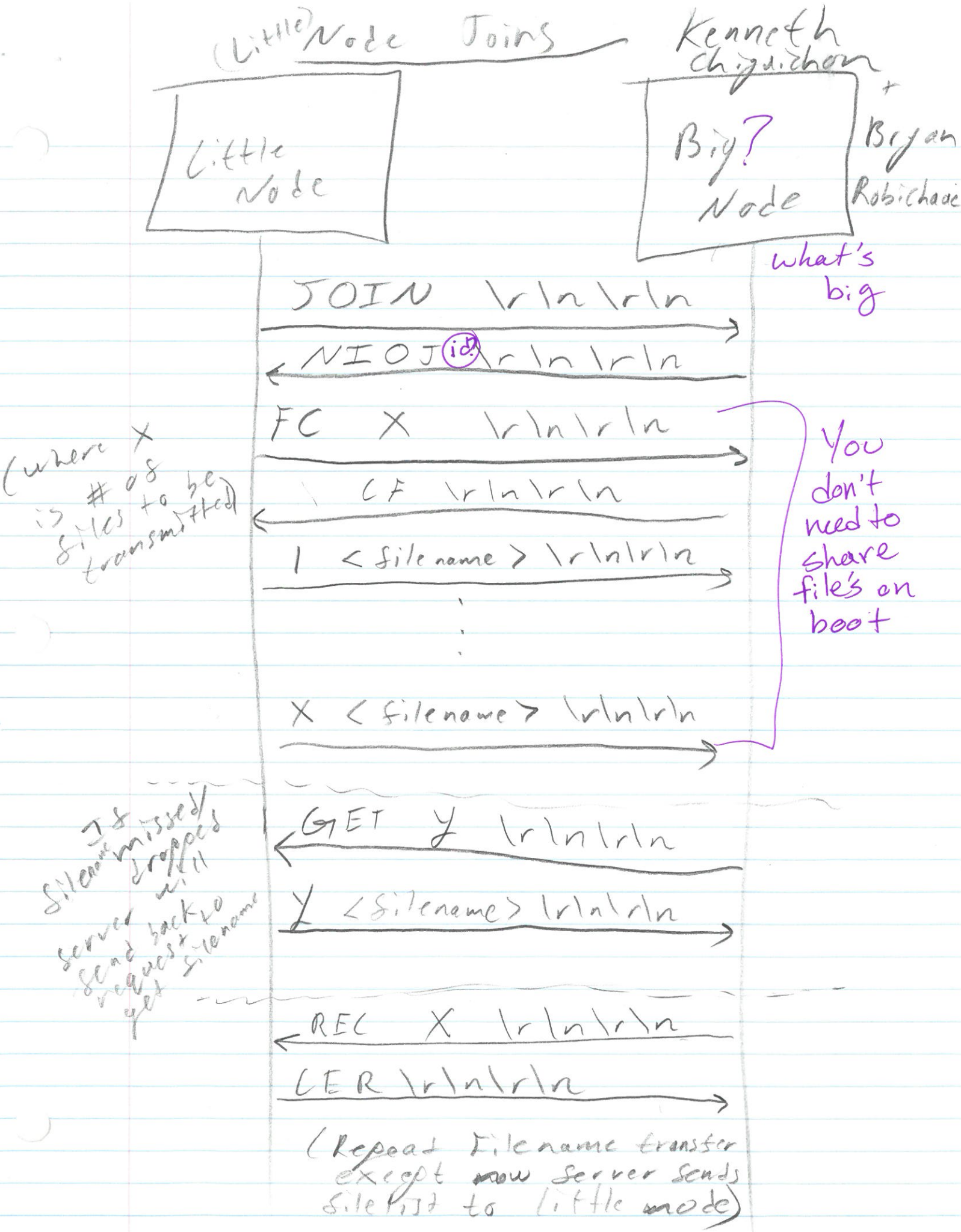
timeout

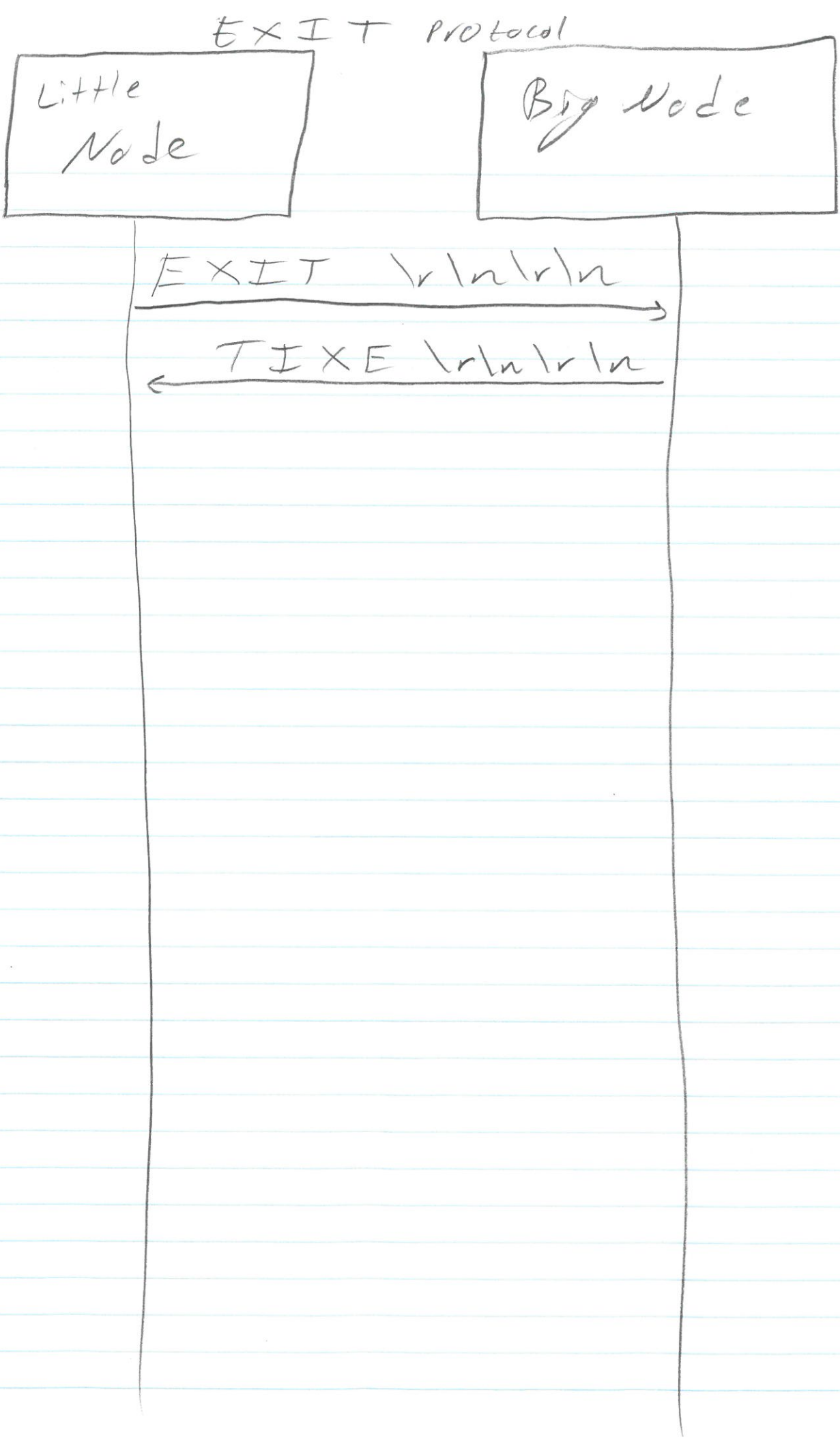
hashing



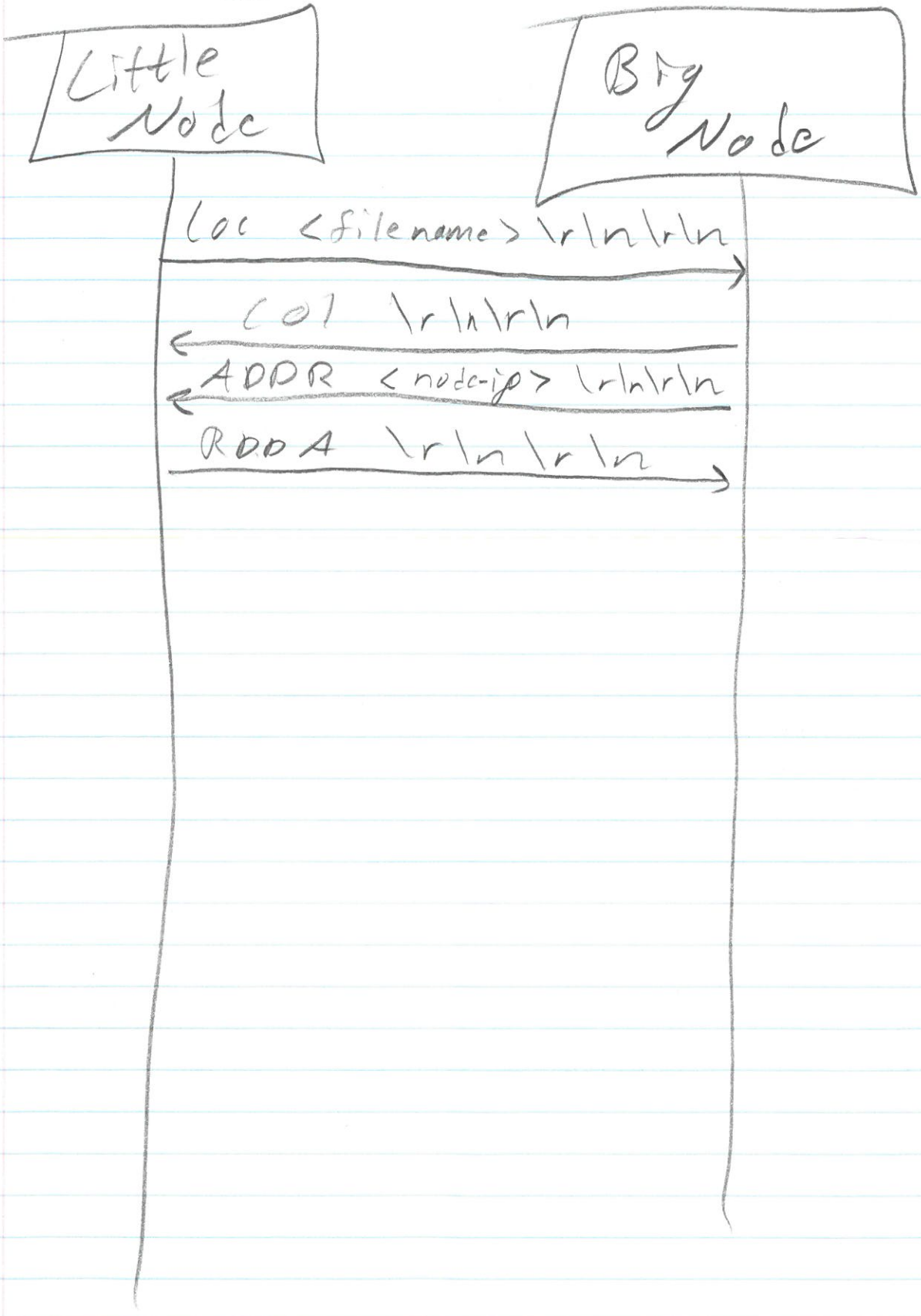
times out



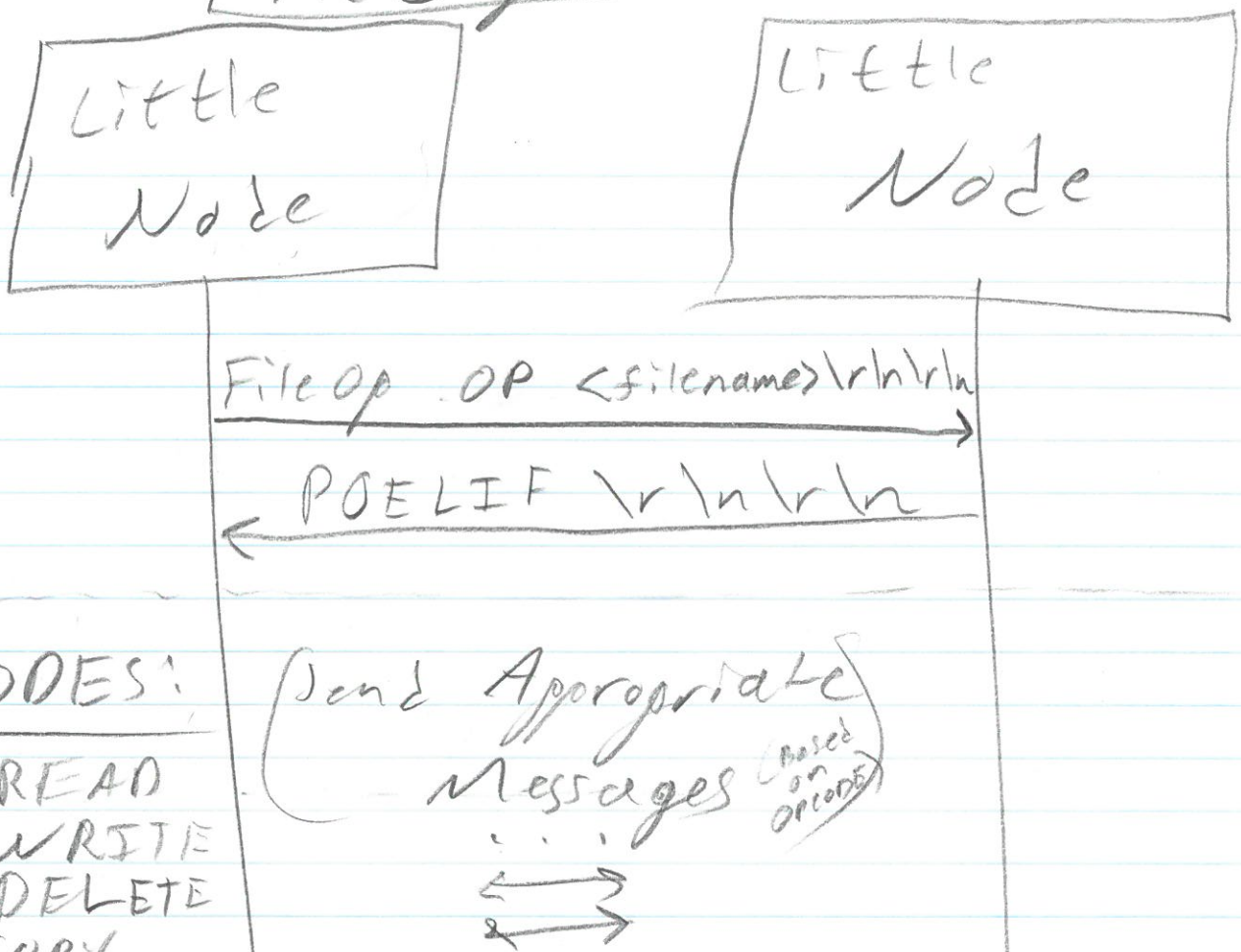

(Little) Node Joins
Kenneth Chijioke
+
Bryan Robichaud
Little Node
Big? Node
what's big
JOIN \r\n\r\n
NIOJ id \r\n\r\n
FC X \r\n\r\n
(where X # of files to be transmitted)
CF \r\n\r\n
1 <filename> \r\n\r\n
X <filename> \r\n\r\n
You don't need to share file's on boot
I missed/dropped file(s) in server send back to request filename(s) get
GET Y \r\n\r\n
Y <filename> \r\n\r\n
REC X \r\n\r\n
CER \r\n\r\n
(Repeat Filename transfer except now server sends file list to little node)


EXIT Protocol
Little Node
Big Node
EXIT \r\n\r\n
TIXE \r\n\r\n

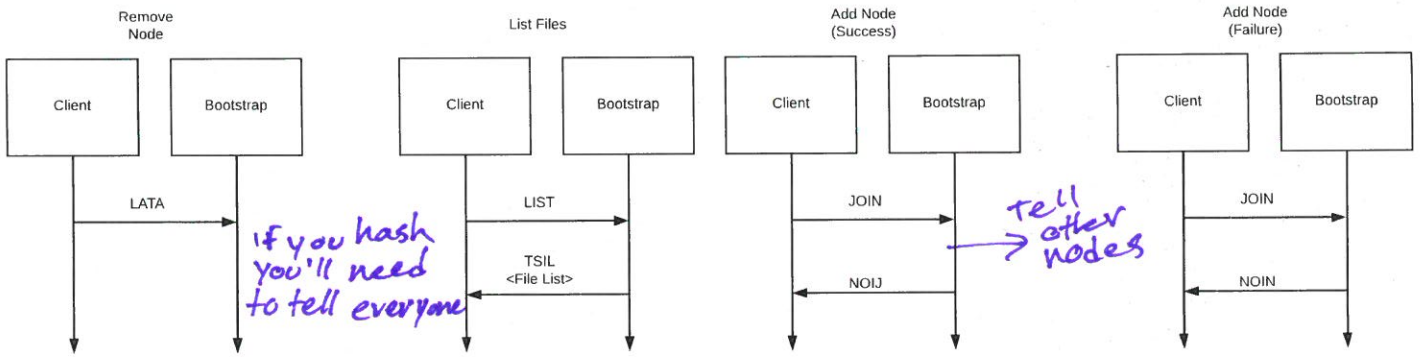
File Location Res

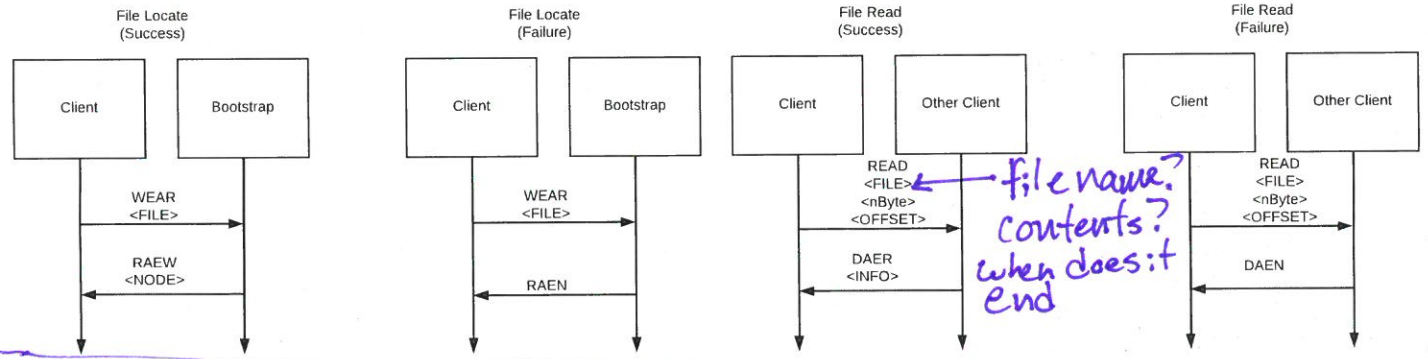


FileOp Protocol

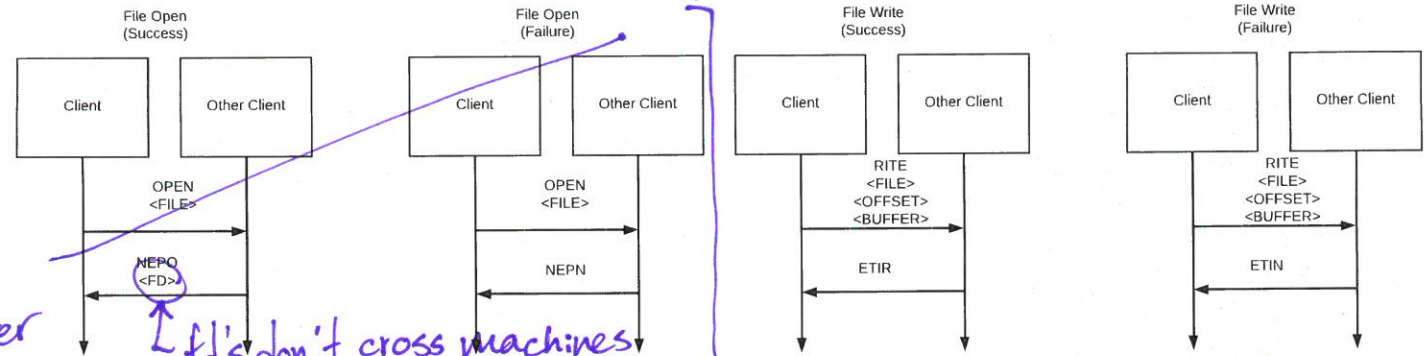


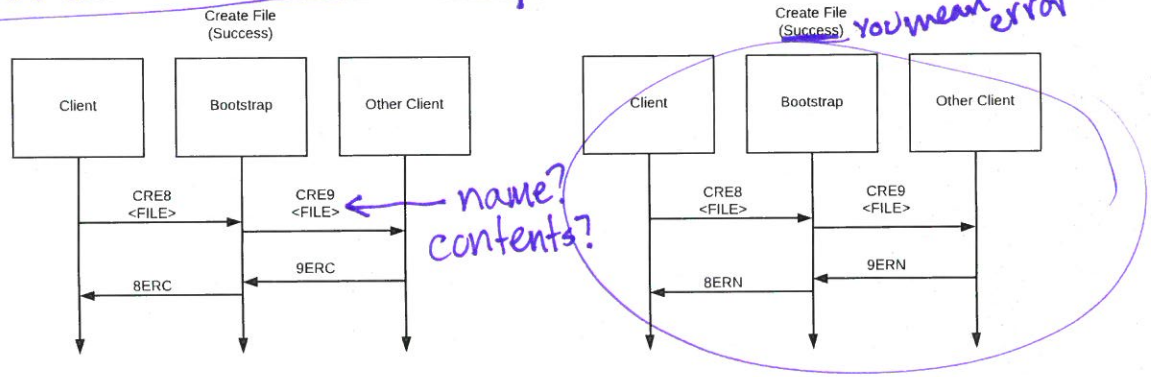
Anthony Mulica and David Shank





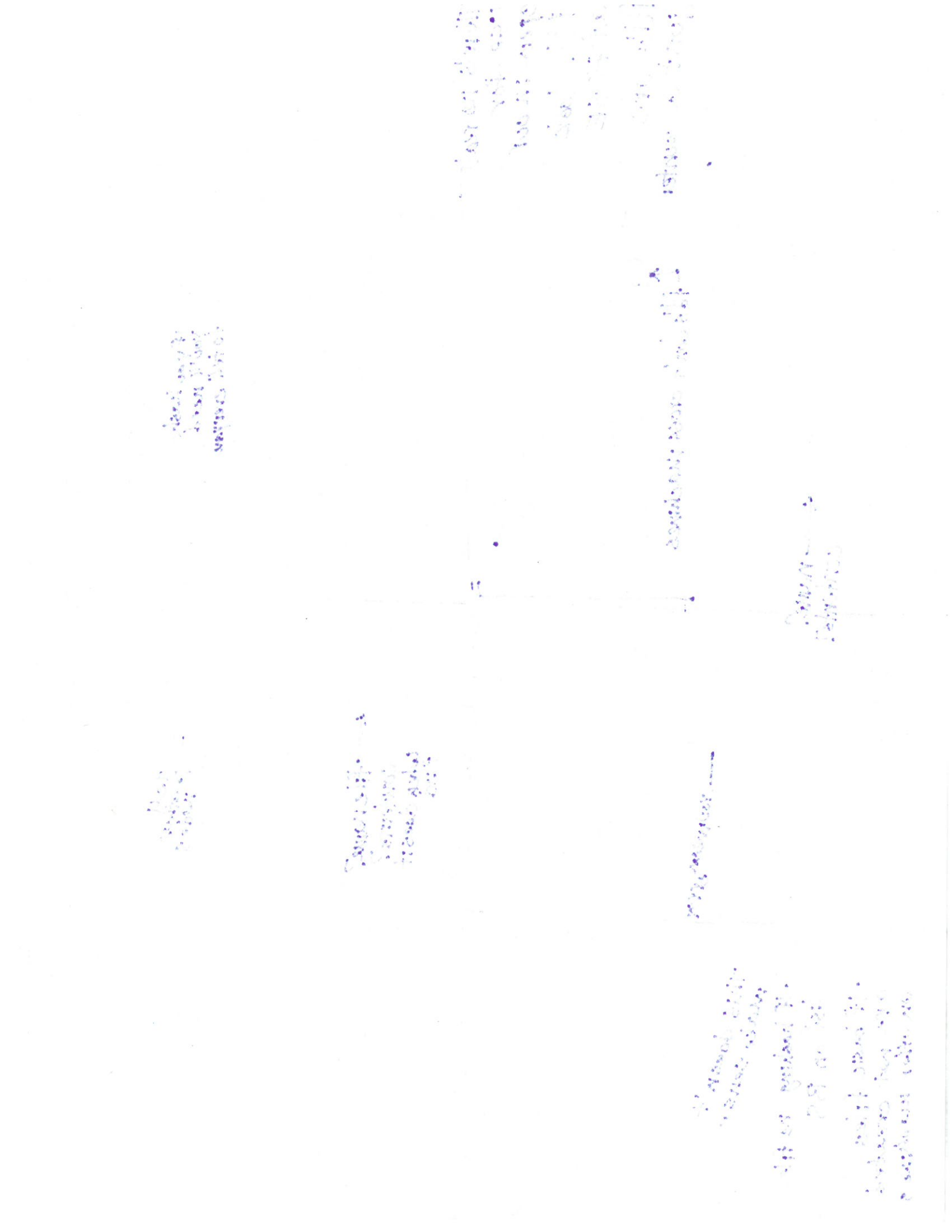
There's no real use for open in our FS. look at FUSE fd's are just a counter



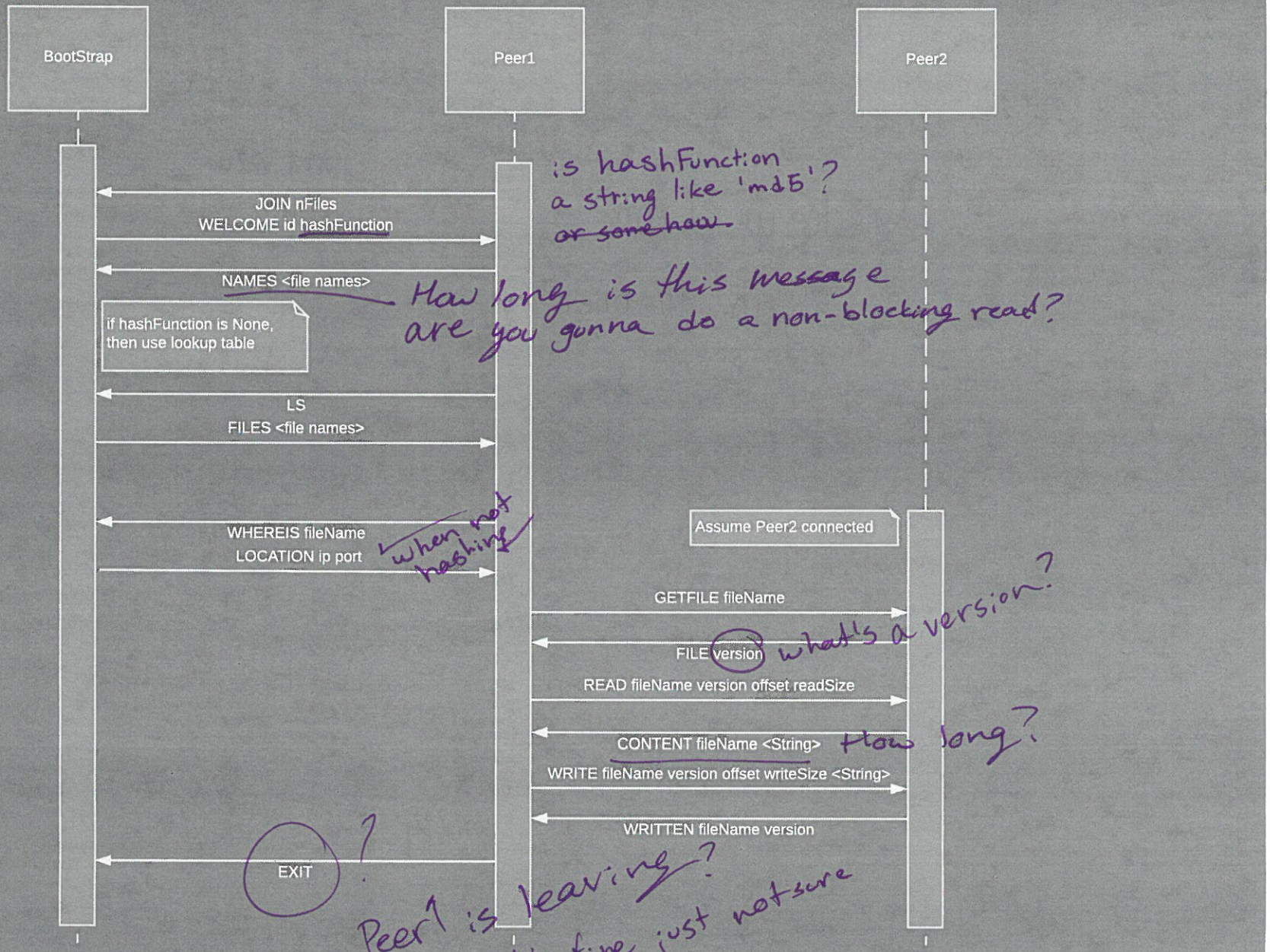


This doesn't make sense

- If hashing no file list on BS
- If basic files are not created on other machines



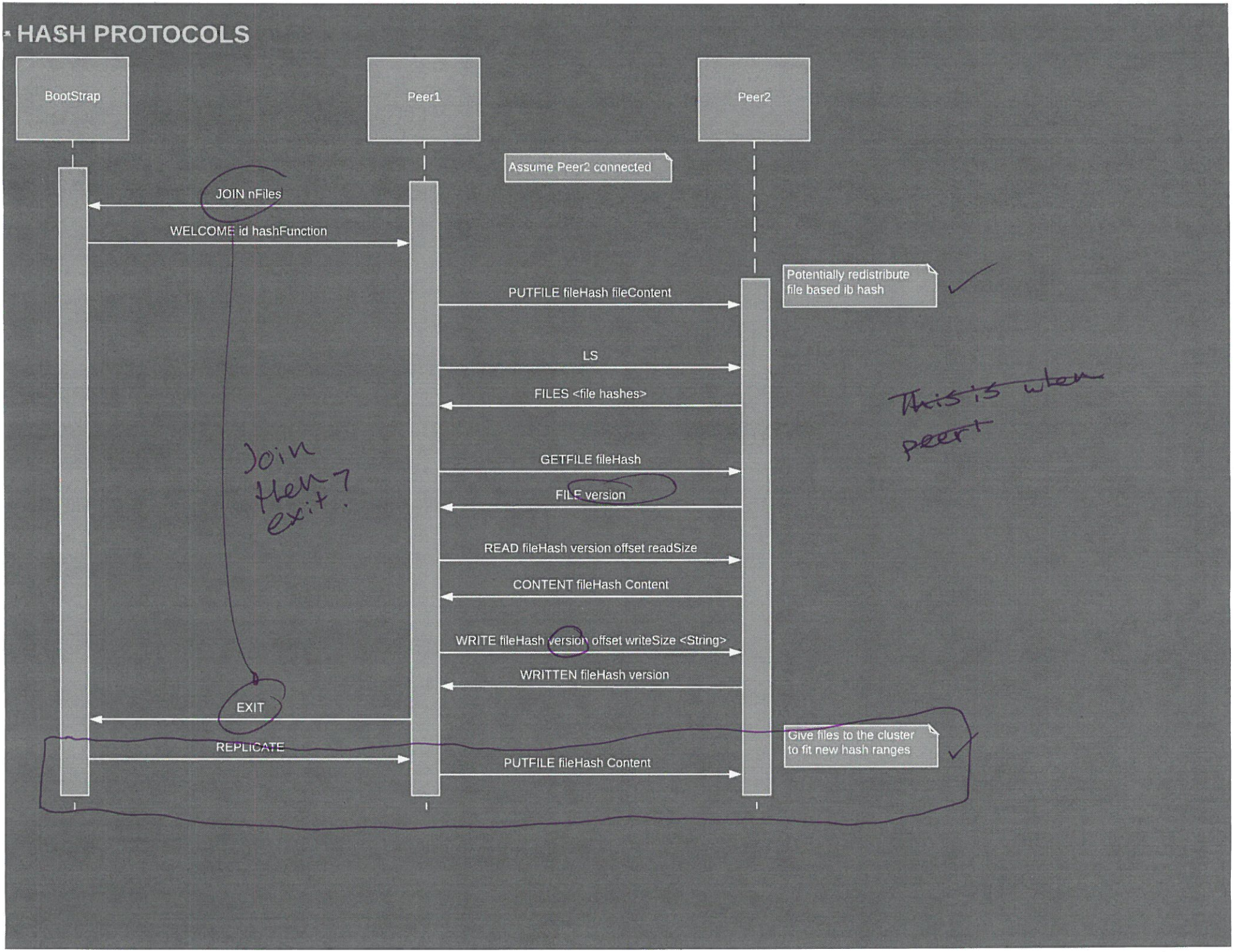
ORIGINAL PROTOCOLS



Andrew Lam

Ryan Webber

HASH PROTOCOLS



Ryan Webber

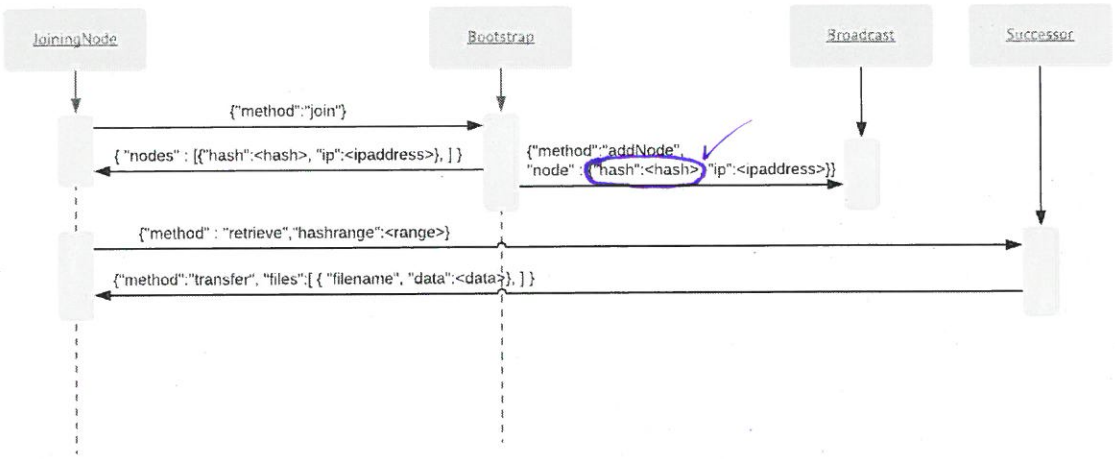
Andrew Lam

CSE392 HW3 Protocol

Rohith Rokkam and Chris Morales

These diagrams are for the consistent hashing tier of the assignment. The bootstrap node will always have an updated list of nodes on the network (this is assuming everyone exits gracefully). It will not have a list of files / file metadata (these will be on their respective nodes).

Join:



When a node wants to join the network, it contacts the bootstrap (at a well-known IP/port). The Bootstrap node will give back the map of nodes to IP addresses, including the new node. The new node will then find the range of files it needs to take from the successor, and request them from the successor.

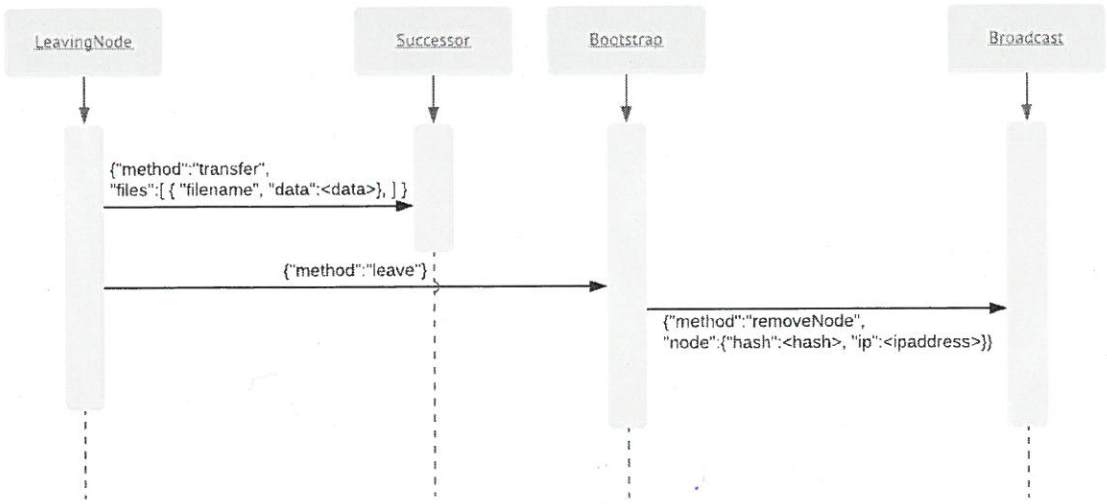
this is for balancing files I assume. Sounds good.

what's the hash? unique If it's an id then cool

When the bootstrap broadcasts the new node I suppose the "successor" waits for the retrieve files message does this make more sense than the "successor" ~~fore~~ sending over the files?

- just something to think about (not implying incorrectness)

Leave:



It's not a requirement to do this but cool beans if you do

When a node leaves the network, the node that is leaving will first transfer its files to its successor. It will then inform the bootstrap node that it is leaving. The bootstrap will remove the node from its table and broadcast the update to the other nodes.

Functions:

JSON is great but how will you know how much to read. Pascal ~~string~~ string?

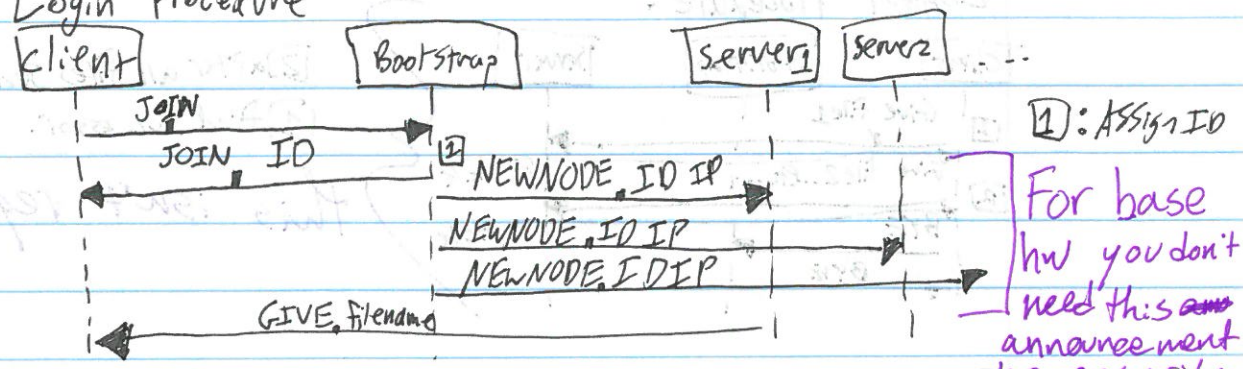
```
{
        "method" : <method_name>,
        ...
}
```

The "method" field can take on the following values: getattr, open, read, rename, *truncate*, unlink, *utimens*, write.

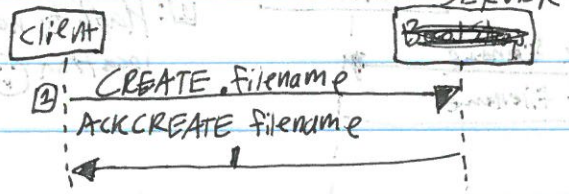
Implement each as needed

Itai Flam Benito Kestelman 4/19/18

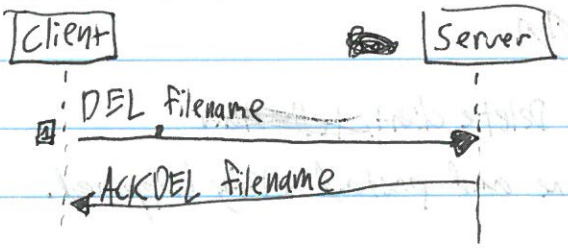
Login Procedure



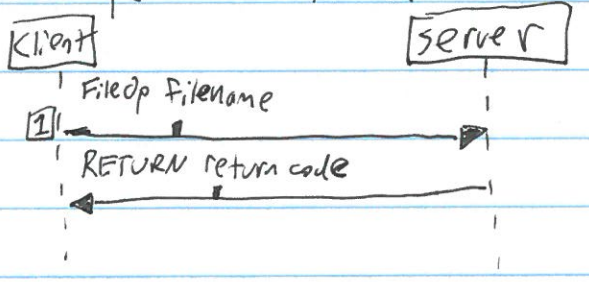
Create File Procedure:



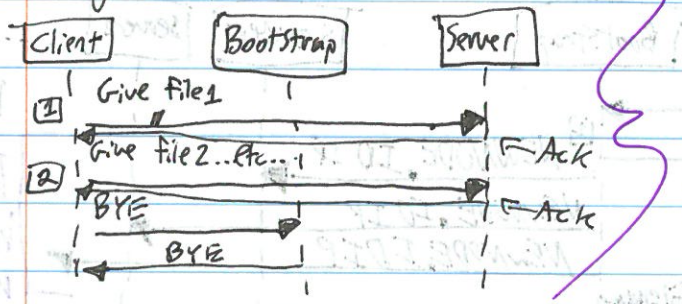
Delete File Procedure:



File Operation Procedure (~~Rename~~, write, read, etc.)



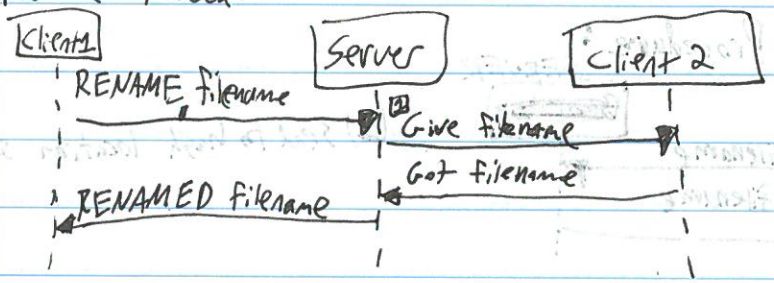
Logout Procedure:



2 after all files sent.
1 find successor.

this isn't required

Rename Procedure:



1: Hash to new file location :)

File not Found Error: Returns w/ return code of function call.
Any other Error with File op ↗↗

Recieve Garbage: Server: Delete client, ~~tell client~~
Client: ~~Move files to~~ Ignore and pretend nothing happened.