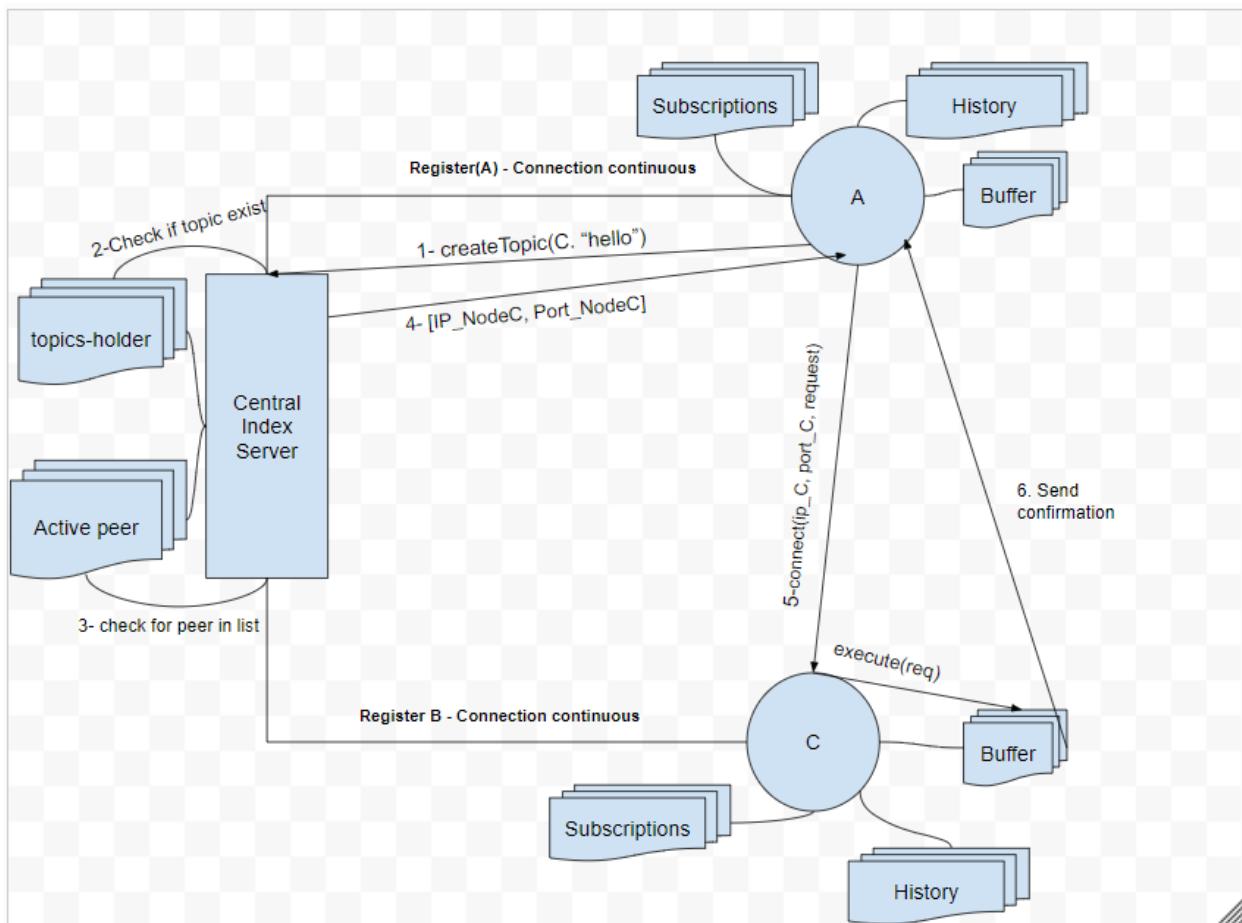


## 1. Design of the System and Discussion

The attached picture is the overview of my system design which I used. There are 2 peer nodes **A**, **B**, and a central index server. The central index server hosts a dictionary for topic-holder nodes and a dictionary of active peers. Peer- [ip, port]. It uses it to manage the changes that occur and based on the peer node operation keeps the up-to-date information about the system. Each peer node has its Buffer, where it stores the messages, History to keep track of the reading history for each peer node that subscribed to a specific topic, and a list of subscriptions of nodes.

The connection between any peer node and the Index server is **continuous**, but the connection between nodes is 1 session only. Nodes after executing the request close the connection between each other.



Let's see scenario described in the attached picture. Once server start running, node A first register itself with server. Server assigns port address to the node A. After it stores information of A in Active peer dictionary. It also initializes **topics** dictionary to empty as there's no topic yet.

Peer node C also does the same.

Peer A requests to create the topic "**hello**" in Peer C. It sends the request to the index server. The index server checks if the topic "**hello**" already exists or not. If not, it checks if there is **peer C** in the active peer dictionary. If there is no such topic and peer C in the list of active peer dictionaries, then it fetches the credentials of the peer node C and sends it back to the requestor which is peer A. After peer A

**connects to** peer C. Peer C deserializes the packet and checks the request. After it executes the request, it sends back the reply as confirmation and closes the connection.

### Tradeoffs of design

1. One of the benefits of this design system is it uses a very simple structure that does not cause overhead to the CPU.
  - a. One of the benefits of using Buffer dictionary for index servers to store the topic and their holder is that access time is  $O(1)$ . Even if we have 1 million topics to hold in the buffer, it still takes constant time to find the elements.
  - b. In rare situations (e.g., many collisions), access time can degrade to  $O(n)$ , where  $n$  is the number of items in the dictionary

Since I'm building this system locally, there's not such issues regarding the security and the privileges to the data. But what if this system is not local, in that case, other peers directly accessing my topics is not very secure. Even if there are protocols, other peer nodes that are not reliable can corrupt my buffer. Therefore, we need to also make sure that authorization to the files is highly considered.

For this system, APIs are very sufficient as amount of the data being transferred are negligible, but if there's many peer nodes and each of them require a quite bit of bandwidth of network for file transfer, it will be painful.

## 2. Deploying 3 peers and index server.

As can be seen in the screenshot, 3 peers named **Chicago**, **London**, and **Paris** have connected to the index server. As soon as we run the peer file and specify `peername="name"`, it calls the `registerPeer` and sends the request to the index server. The index server going to store it in the list of active peers. Meanwhile, peers also start listening for the incoming connection while being connected to the index server.

*Serv-U*

```
server listening on 127.0.0.1 5000
Accepted connection from ('127.0.0.1', 48452)
[19:19:18] : [Request: {'requestName': 'registerPeer', 'peerName': 'chicago'}] P1
Topic: {}

Active Users: {'chicago': ('127.0.1.1', 7438)}
Accepted connection from ('127.0.0.1', 48454)

[19:19:20] : [Request: {'requestName': 'registerPeer', 'peerName': 'london'}] P2
Topic: {}

Active Users: {'chicago': ('127.0.1.1', 7438), 'london': ('127.0.1.1', 7122)}
Accepted connection from ('127.0.0.1', 48456)

[19:19:23] : [Request: {'requestName': 'registerPeer', 'peerName': 'paris'}] P3
Topic: {}

Active Users: {'chicago': ('127.0.1.1', 7438), 'london': ('127.0.1.1', 7122), 'paris': ('127.0.1.1', 6051)} P3

issacode@DESKTOP-J6HG0H2:~/secondPA/02_Muradli_Isa_PA#2$ make runPeer peername=chicago
Calling...
[19:19:18] : [Registering Peer Node: chicago] P1
[19:19:18] : [Response from Index Server: Registered Peer: chicago at port 7438]

2 - CreateTopic
3 - DeleteTopic
4 - SendMessage to topic
5 - Subscribe
6 - Pull
7 - Unregister Peer and exit!
Choose option-->
[19:19:18] : [Listening for messages on 127.0.1.1 7438]
```

```
issacode@DESKTOP-J6HG0H2:~/secondPA/02_Muradli_Isa_PA#2$ make runPeer peername=london
Calling...
[19:19:20] : [Registering Peer Node: london] P2
[19:19:20] : [Response from Index Server: Registered Peer: london at port 7122]

2 - CreateTopic
3 - DeleteTopic
4 - SendMessage to topic
5 - Subscribe
6 - Pull
7 - Unregister Peer and exit!
Choose option-->
[19:19:20] : [Listening for messages on 127.0.1.1 7122]
```

```
issacode@DESKTOP-J6HG0H2:~/secondPA/02_Muradli_Isa_PA#2$ make runPeer peername=paris
Calling...
[19:19:23] : [Registering Peer Node: paris]
[19:19:23] : [Response from Index Server: Registered Peer: paris at port 6051]

[19:19:23] : [Listening for messages on 127.0.1.1 6051] P3

2 - CreateTopic
3 - DeleteTopic
4 - SendMessage to topic
5 - Subscribe
6 - Pull
7 - Unregister Peer and exit!
Choose option-->
```

## a. Create topic API.

Peer nodes can create a topic on their own side or choose to host it in the peer they want. For that we **Option 2: Create Topic**.

**Let's first create a topic on the local side.** As you can see once we choose option 2, the terminal prompts to ask for PeerName where to create, and topic name.

After it builds the request and sends it to the index server. The Index server receives the request and deserializes it. After it checks for the Peer node and if the topic exists already in its buffer. It fulfills the request and creates the response **Approved**. The response was sent back to the peer **Chicago** and the topic was created locally in peer Chicago, it's showing on the right side of the screenshot. Because the topic was just created no messages and subscriptions exist.

```
Topic: {}

Active Users: {'chicago': ('127.0.1.1', 7438), 'london': ('127.0.1.1', 7122)}
Accepted connection from ('127.0.0.1', 48456)

[19:19:23] : [Request: {'requestName': 'registerPeer', 'peerName': 'paris'}]

Topic: {}

Active Users: {'chicago': ('127.0.1.1', 7438), 'london': ('127.0.1.1', 7122), 'paris': ('127.0.1.1', 6051)}

[19:25:27] : [Request: {'requestName': 'createTopic', 'caller': 'chicago', 'peerToWrite': 'chicago', 'topic': 'hello'}]
-----
{'response': 'Approved'}

Topic: {'hello': 'chicago'}

Active Users: {'chicago': ('127.0.1.1', 7438), 'london': ('127.0.1.1', 7122), 'paris': ('127.0.1.1', 6051)}

issacode@DESKTOP-J6HG0H2:~/secondPA/02_Muradli_Isa_PA#2$ make runPeer peername=paris
Calling...
```

The terminal session shows the creation of a topic 'hello' in peer Chicago. The session is labeled 'Server' on the left and 'Peer' on the right. Red boxes highlight the user input 'chicago' and 'hello', the response 'Approved', and the topic entry in the peer's buffer.

**Let's create a topic now in the other node.**

This time **Peer London** will try to create a topic **welcome** in **Peer Chicago**.

4 - First Peer London sends the request to the index server. The index server checks if a specified peer node exists in the active directory.

5 - If verifies for the forward and sends back to peer London the connection credentials of the peer Chicago.

6. London receives it, sees the forwarding message, and takes the credentials of peer Chicago to connect.

7. It establishes the connection with peer Chicago

8. Peer Chicago accepts the connection from London.

9. Deserializes the request and fulfills it locally.

10. It sends back an acknowledgement message "Topic Welcome is created!"

11/12 – Peer London receives back acknowledgment about the creation of a topic in peer Chicago

13 – The server also updates the list of topics and assigns the topic Welcome to Chicago in its Topic buffer.

CHICAGO

```

issacode@DESKTOP-J6HG0H2:~/secondPA/02_Muradli_Isa_PA#2
[19:25:27] : [Request: {'requestName': 'createTopic', 'caller': 'chicago', 'peerToWrite': 'chicago', 'topic': 'Hello'}]
-----
{'response': 'Approved'}
Topic: {'Hello': 'chicago'}
Active Users: {'chicago': ('127.0.1.1', 7438), 'london': ('127.0.1.1', 7122), 'paris': ('127.0.1.1', 6051)}
[19:33:44] : [Request: {'requestName': 'createTopic', 'caller': 'london', 'peerToWrite': 'chicago', 'topic': 'Welcome'}]
-----
{'response': 'forward', 'PeerName': '127.0.1.1', 'PeerPort': 7438} 5
Topic: {'Hello': 'chicago', 'Welcome': 'chicago'} 13
Active Users: {'chicago': ('127.0.1.1', 7438), 'london': ('127.0.1.1', 7122), 'paris': ('127.0.1.1', 6051)}

issacode@DESKTOP-J6HG0H2:~/secondPA/02_Muradli_Isa_PA#2$ make runPeer peername=paris
Calling...
[19:19:23] : [Registering Peer Node: paris]
[19:19:23] : [Response from Index Server: Registered Peer: paris at port 6051]
[19:19:23] : [Listening for messages on 127.0.1.1 6051]

2 - CreateTopic
3 - DeleteTopic
4 - SendMessage to topic
5 - Subscribe
6 - Pull
7 - Unregister Peer and exit!
Choose option-->

```

8

Server

```

4 - SendMessage to topic
5 - Subscribe
6 - Pull
7 - Unregister Peer and exit!
Choose option-->
[19:33:44] : [Node chicago accepted connection from ('127.0.0.1', 47212)]
[19:33:44] : [Handling the incoming request : {'requestName': 'createTopic', 'caller': 'london', 'peerToWrite': 'chicago', 'topic': 'Welcome'}]
[[19:33:44]] : [Finished incoming request from london. Response back to peer node: Topic Welcome is created!] 9
----- 10
Which peer node you want to create topic -> chicago 1
What topic you want to create -> Welcome 2
Creating topic...
[19:33:44] : f[Node london called request : {'requestName': 'createTopic', 'caller': 'london', 'peerToWrite': 'chicago', 'topic': 'Welcome'}] 3
[19:33:44] : [Response from server: forward to chicago] 6
----- 7
Sending message to Peer
[19:33:44] : [Connection established with ('127.0.1.1', 7438)] 8
[19:33:44] : [Node london sending request {'requestName': 'createTopic', 'caller': 'london', 'peerToWrite': 'chicago', 'topic': 'Welcome'} to Peer at 127.0.1.1 : 7438] 9
[19:33:44] : [Node london received reply: Topic Welcome is created!] 11
----- 12
[19:33:44] : [Topic Welcome is created at Peer: chicago]

```

11/79

LONDON

If peer London tries to create the same topic in Chicago or even locally, the Index server will check in its topics buffer if there is such a topic. Therefore, the node gets back a response as “Topic Welcome already created”

```

[19:44:32] : [Request: {'requestName': 'createTopic', 'caller': 'london', 'peerToWrite': 'chicago', 'topic': 'Welcome'}]
-----
{'response': 'Topic Welcome already exists!'} 3
Topic: {'Hello': 'chicago', 'Welcome': 'chicago'}
Active Users: {'chicago': ('127.0.1.1', 7438), 'london': ('127.0.1.1', 7122), 'paris': ('127.0.1.1', 6051)}

issacode@DESKTOP-J6HG0H2:~/secondPA/02_Muradli_Isa_PA#2$ make runPeer peername=paris
Calling...
[19:19:23] : [Registering Peer Node: paris]
[19:19:23] : [Response from Index Server: Registered Peer: paris at port 6051]
[19:19:23] : [Listening for messages on 127.0.1.1 6051]

2 - CreateTopic
3 - DeleteTopic
4 - SendMessage to topic
5 - Subscribe
6 - Pull
7 - Unregister Peer and exit!
Choose option-->

```

Topic Welcome is created!

1

Choose option-->

2

Which peer node you want to create topic -> chicago

3

What topic you want to create -> Welcome

4

Creating topic...

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

## b. Delete topic API.

- 1- Peer London tries to delete the topic “**sampleTopic**” that it hosts. It exists on the server side as can be seen in point
- 2/3 - London creates a request to send to the index server to delete it.
- 4 – The server deserializes the request and deletes it from the topics buffer. As it's shown in the? Mark put in the screenshot
- 5- London receives acknowledgement of deletion
- 6- It's also deleted locally from London's side

```
Active Users: {'chicago': ('127.0.1.1', 7438), 'london': ('127.0.1.1', 7122), 'paris': ('127.0.1.1', 6051)}
```

[19:49:43] : [Request: {'requestName': 'createTopic', 'caller': 'london', 'peerToWrite': '3', 'topic': 'sampleTopic'}]

-----

{'response': 'Topic sampleTopic already exists!'} 1

Topic: {'hello': 'chicago', 'Welcome': 'chicago', 'sampleTopic': 'london'}

```
Active Users: {'chicago': ('127.0.1.1', 7438), 'london': ('127.0.1.1', 7122), 'paris': ('127.0.1.1', 6051)}
```

[19:49:52] : [Request: {'requestName': 'deleteTopic', 'caller': 'london', 'topic': 'sampleTopic'}]

Topic: {'hello': 'chicago', 'Welcome': 'chicago'} 2

```
Active Users: {'chicago': ('127.0.1.1', 7438), 'london': ('127.0.1.1', 7122), 'paris': ('127.0.1.1', 6051)}
```

issacode@DESKTOP-J6HG0H2:~/secondPA/02\_Muradli\_Isa\_PA#2\$ make runPeer peername=paris Calling...

[19:19:23] : [Registering Peer Node: paris]

[19:19:23] : [Response from Index Server: Registered Peer: paris at port 6051]

[19:19:23] : [Listening for messages on 127.0.1.1 6051]

2 - CreateTopic  
3 - DeleteTopic  
4 - SendMessage to topic  
5 - Subscribe  
6 - Pull  
7 - Unregister Peer and exit!  
Choose option-->

4 - SendMessage to topic  
5 - Subscribe  
6 - Pull  
7 - Unregister Peer and exit!

[19:33:44]: [Handling the incoming request : {'requestName': 'createTopic', 'caller': 'london', 'peerToWrite': 'chicago', 'topic': 'Welcome'}]

[19:33:44]: [Finished incoming request from london. Response back to peer node: Topic Welcome is created!] 3

6 - Pull  
7 - Unregister Peer and exit!  
Choose option-->3

What topic you want to delete -> sampleTopic 4

Deleting topic...  
[19:49:52] : [Node london called request : {'requestName': 'deleteTopic', 'caller': 'london', 'topic': 'sampleTopic'}]

[19:49:52] : [Response from server: Topic sampleTopic is deleted from node london!] 5

Message : {} 6

Subscriptions : {}

2 - CreateTopic  
3 - DeleteTopic  
4 - SendMessage to topic  
5 - Subscribe  
6 - Pull  
7 - Unregister Peer and exit!  
Choose option-->

If London tries to delete a topic from Chicago, it can also be done.

1. London specifies a topic name to delete.
2. Creates the request and send it to the index server.
3. The server receives, handles the request and send forwarding response for London
4. Server updates buffer by deleting topic “**hello**”.
5. London receives **forwarding** and gets the credentials of Chicago
6. Establishes the connection with Chicago
7. Chicago accepts the connections
8. London sends the request for execution
9. Chicago accomplishes the request and deletes the topic locally
10. Sends the **Topic hello is deleted** back to peer London
11. London receives the message and closes the connection with Chicago

```

Accepted connection from ('127.0.0.1', 48464)
[19:57:45] : [Request: {'requestName': 'registerPeer', 'peerName': 'london'}
Topic: {'hello': 'chicago', 'Welcome': 'chicago'}
Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510)}
Accepted connection from ('127.0.0.1', 48466)
[19:57:52] : [Request: {'requestName': 'registerPeer', 'peerName': 'paris'}
Topic: {'hello': 'chicago', 'Welcome': 'chicago'}
Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}
[19:58:05] : [Request: {'requestName': 'deleteTopic', 'caller': 'london', 'topic': 'hello'}  

Topic: {'Welcome': 'chicago'} Deleted 4
Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}
```

---

```

issacode@DESKTOP-J6HG0H2:~/secondPA/02_Muradli_Isa_PA#2$ make runPeer peername=paris
Calling...
[19:57:52] : [Registering Peer Node: paris]
[19:57:52] : [Response from Index Server: Registered Peer: paris at port 5191]
[19:57:52] : [Listening for messages on 127.0.1.1 5191]
2 - CreateTopic
3 - DeleteTopic
4 - SendMessage to topic
5 - Subscribe
6 - Pull
```

---

```

4 - SendMessage to topic
5 - Subscribe
6 - Pull
7 - Unregister Peer and exit!
Choose option-->
[19:58:05]: [Node chicago accepted connection from ('127.0.0.1', 40462)] 7
[19:58:05]: [Handling the incoming request : {'requestName': 'deleteTopic', 'caller': 'london', 'topic': 'hello'}]
[[19:58:05]] : [Finished incoming request from london. Response back to peer node: Topic hello is deleted!] 9
Message: {'Welcome': []} 10
Subscriptions: {'Welcome': []}
} Deleted 10
```

---

```

What topic you want to delete -> hello 1
[19:58:05] : [Request forwarded to chicago] 5
Deleting topic... 2
[19:58:05] : [Node london called request : {'requestName': 'deleteTopic', 'caller': 'london', 'topic': 'hello'}]
[19:58:05] : [Request forwarded to chicago] 5
Sending message to Peer
[19:58:05] : [Connection established with ('127.0.1.1', 7436)] 6
[19:58:05] : [Node london sending request {'requestName': 'deleteTopic', 'caller': 'london', 'topic': 'hello'} to Peer at 127.0.1.1 : 7436] 8
[19:58:05] : [Node london received reply: Topic hello is deleted!] 11
```

### c. Send API.

First peer Paris is sending a message to its own topic personalTopic

1/2 It gives the topic name and the message to send

3 – A message is being sent to the topic

4 – It creates the request and sends it to the index server.

5 – The index server checks if there's such a topic and who is the holder of that topic. Since Paris itself is the holder of a topic, it does not forward it.

6 – Paris gets from the server that the message is sent to the topic.

7- It gets added to the message Buffer

It successfully finishes the request

```
Topic: {'Welcome': 'chicago', 'CS550': 'london', 'personalTopic': 'paris'}

Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}

[20:39:43] : [Request: {'requestName': 'deleteTopic', 'caller': 'paris', 'topic': 'exit'}

Topic: {'Welcome': 'chicago', 'CS550': 'london', 'personalTopic': 'paris'}

Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}

[20:39:55] : [Request: {'requestName': 'send', 'caller': 'paris', 'topic': 'personalTopic', 'message': '12345'}

Topic: {'Welcome': 'chicago', 'CS550': 'london', 'personalTopic': 'paris'} ← itself
Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}
```

What topic you want to send message -> personalTopic 1

What message to send -> 12345 2

Sending message 12345 to personalTopic 3

```
[20:39:55] : [Node paris called request : {'requestName': 'send', 'caller': 'paris', 'topic': 'personalTopic', 'message': '12345'}]
```

[20:39:55] : [Response from server: Message sent to topic] 6

Message : {'personalTopic': ['12345']}

Subscriptions : {'personalTopic': []}

2 - CreateTopic

3 - DeleteTopic

4 - SendMessage to topic

5 - Subscribe

✓ done 7

5

[3/225]

4

In this scenario, peer Paris is sending the message **HelloFromParis** to peer node London.

1 – Create the request to send the message **HelloFromParis** to topic **CS550**.

2/3 – The Index server deserializes the request and checks for who is the holder of the topic **CS550**, it is London

4 – Peer Paris gets response from the index server **Forward to Node London** and also provides the address of the London peer node

5. Paris establishes the connection with the peer node London

6. London accepts the incoming connection from Paris

7. Paris send the same request with the message and topic name to the London

8. London deserializes the request and handles the request.

9. It adds the message to the topic CS550 and sends back the acknowledgment

10 – Paris receives a response from London that **HelloFromParis** has been added to the **CS550** in peer London

```
['response': 'Approved']

Topic: {'Welcome': 'chicago', 'CS550': 'london'} 2

Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}

[20:34:16] : [Request: {'requestName': 'send', 'caller': 'paris', 'topic': 'CS550', 'message': 'HelloFromParis'}] 3

Topic: {'Welcome': 'chicago', 'CS550': 'london'}

Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)} 4

[12/161] 5

Sending message HelloFromParis to CS550 1

[20:34:16] : [Node paris called request: {'requestName': 'send', 'caller': 'paris', 'topic': 'CS550', 'message': 'HelloFromParis'}]

[20:34:16] : [Response from server: Forwarded to Node london] 4

Sending message to Peer 5

[20:34:16] : [Connection established with ('127.0.1.1', 5510)] 5

[20:34:16] : [Node paris sending request: {'requestName': 'send', 'caller': 'paris', 'topic': 'CS550', 'message': 'HelloFromParis'} to Peer at 127.0.1.1 : 5510] 6

[20:34:16] : [Node paris received reply: Message 'HelloFromParis' is added to CS550] 10

[20:34:16] : [Message HelloFromParis is send to Topic:CS550 at ['127.0.1.1']] 7

5 - Subscribe
6 - Pull
7 - Unregister Peer and exit!
Choose option-->
[19:58:05]: [Node chicago accepted connection from ('127.0.0.1', 40462)]
[19:58:05]: [Handling the incoming request: {'requestName': 'deleteTopic', 'caller': 'london', 'topic': 'hello'}]
[[19:58:05]] : [Finished incoming request from london. Response back to peer node: Topic hello is deleted!]
Message: {'Welcome': []}
Subscriptions: {'Welcome': []} 8

2 - CreateTopic
3 - DeleteTopic
4 - SendMessage to topic
5 - Subscribe
6 - Pull
7 - Unregister Peer and exit!
Choose option-->
[20:34:16]: [Node london accepted connection from ('127.0.0.1', 55752)] 6
[20:34:16]: [Handling the incoming request: {'requestName': 'send', 'caller': 'paris', 'topic': 'CS550', 'message': 'HelloFromParis'}] 8
[[20:34:16]] : [Finished incoming request from paris. Response back to peer node: Message 'HelloFromParis' is added to CS550] 9
Message: {'CS550': ['HelloFromParis']}
Subscriptions: {'CS550': []}
```

## d. Subscribe API.

1. Peer node Paris chooses to subscribe to the topic Welcome
  2. It creates a request and sends to the index server for the holder of that topic
  3. Index server deserializes the request and find the holder of the topic (Chicago) and sends it to Paris
  4. Paris receives that forwarding response from Index server
  5. Establishes a connection with the peer Chicago
  6. Peer node Chicago accepts the connection
  7. Peer Paris forwards the request to the Chicago node
- 8/9/10 Chicago node fulfills the request and adds the Paris as subscribed to the topic welcome
11. Peer node Paris gets the response that it has been subscribed to the topic

```

Topic: {'Welcome': 'chicago', 'CS550': 'london', 'personalTopic': 'paris'}
Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}
[21:12:41] : [Request: {'requestName': 'subscribe', 'caller': 'paris', 'topic': 'Welcome'}]
Topic: {'Welcome': 'chicago', 'CS550': 'london', 'personalTopic': 'paris'}
Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}
[21:15:08] : [Request: {'requestName': 'subscribe', 'caller': 'london', 'topic': 'Welcome'}]
Topic: {'Welcome': 'chicago', 'CS550': 'london', 'personalTopic': 'paris'}
Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}
[21:12:41] : [Request: {'requestName': 'subscribe', 'caller': 'paris', 'topic': 'Welcome'}]
[21:12:41] : [Subscribing to topic Welcome]
[21:12:41] : [Node paris called request : {'requestName': 'subscribe', 'caller': 'paris', 'topic': 'Welcome'}]
4 [21:12:41] : [Response from server : Forwarded to Topic Holder chicago]
1 What topic you want to subscribe -> Welcome [12/264]
2 [21:12:41] : [Subscribing to topic Welcome]
3 [21:12:41] : [Node paris called request : {'requestName': 'subscribe', 'caller': 'paris', 'topic': 'Welcome'}]
4 [21:12:41] : [Response from server : Forwarded to Topic Holder chicago]
5 [21:12:41] : [Connection established with ('127.0.1.1', 7436)]
6 [21:12:41] : [Node paris sending request {'requestName': 'subscribe', 'caller': 'paris', 'topic': 'Welcome'} to Peer at 127.0.1.1 : 7436]
7 [21:12:41] : [Node paris received reply: {'response': 'Peer london is subscribed to Welcome!'}]
8 [21:12:41] : [Node london accepted connection from ('127.0.1.1', 40466)]
9 [21:12:41] : [Handling the incoming request : {'requestName': 'subscribe', 'caller': 'paris', 'topic': 'Welcome'}]
10 [21:12:41] : [Finished incoming request from paris. Response back to peer node : {'response': 'Peer paris is subscribed to Welcome!'}]
11 [21:15:08] : [Connection established with ('127.0.1.1', 7436)]
12 [21:15:08] : [Node london sending request {'requestName': 'subscribe', 'caller': 'london', 'topic': 'Welcome'} to Peer at 127.0.1.1 : 7436]
13 [21:15:08] : [Node london received reply: {'response': 'Peer london is subscribed to Welcome!'}]
14 [21:15:08] : [Message : {'CS550': ['HelloFromParis']}]
15 [21:15:08] : [Subscriptions : {'CS550': []}]
16 2 - CreateTopic
17 3 - DeleteTopic
18 4 - SendMessage to topic
19 5 - Subscribe
20 6 - Pull
21 7 - Unregister Peer and exit!
Choose option-->

```

### Request to subscribe to a topic that doesn't exist.

When Paris sends a request to subscribe to **NonExistingTopic**, the Index server checks for that topic and sends back the response as "**Topic NonExistingTopic doesn't exist to subscribe!**"

```

Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}
[21:35:28] : [Request: {'requestName': 'subscribe', 'caller': 'paris', 'topic': 'NonExistingTopic'}]
Topic: {'Welcome': 'chicago', 'CS550': 'london', 'personalTopic': 'paris'}
Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}
1 What topic you want to subscribe -> NonExistingTopic [3/286]
2 [21:35:28] : [Subscribing to topic NonExistingTopic]
3 [21:35:28] : [Node paris called request : {'requestName': 'subscribe', 'caller': 'paris', 'topic': 'NonExistingTopic'}]
4 [21:35:28] : [Response from server: Topic NonExistingTopic doesn't exist to subscribe!]

```

Server

PARIS

## e. Pull API.

1. Paris request to pull from topic **Welcome**
2. Create request to send to index server
3. Index server deserializes the request and find the holder of the topic (Chicago)
4. Paris gets back response and needs to forward the request to the peer Chicago
5. Paris creates connection with Chicago
6. Chicago accepts the incoming connection from Paris
7. Paris forwards the request to Chicago
8. Chicago handles the incoming request and fetches the unread message from **Welcome** topic
9. Send back the response **[123, abc]**
10. Paris receives the message **[123, abc]**

**Note:** The garbage collection hasn't been called because London who subscribed to the Welcome hasn't read the message. That's why Chicago doesn't flush the message in the **Welcome** topic

```

Topic: {'Welcome': 'chicago', 'CS550': 'london', 'personalTopic': 'paris'}
Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}
[22:08:43]: [Request: {'requestName': 'send', 'caller': 'chicago', 'topic': 'Welcome', 'message': 'abc'}]

Topic: {'Welcome': 'chicago', 'CS550': 'london', 'personalTopic': 'paris'}
Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}
3 [22:08:51]: [Request: {'requestName': 'pull', 'caller': 'paris', 'topic': 'Welcome'}]

Topic: {'Welcome': 'chicago', 'CS550': 'london', 'personalTopic': 'paris'}
Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}

PARIS [12:31]
What topic you want to pull from -> Welcome 1
2 Pulling messages from Welcome
[22:08:51]: [Node paris called request : {'requestName': 'pull', 'caller': 'paris', 'topic': 'Welcome'}]

4 [22:08:51]: [Response from server : Forwarded to chicago]
Sending message to Peer
5 [22:08:51]: [Connection established with ('127.0.1.1', 7436)]
7 [22:08:51]: [Node paris sending request {'requestName': 'pull', 'caller': 'paris', 'topic': 'Welcome'} to Peer at 127.0.1.1 : 7436]
10 [22:08:51]: [Node paris received reply: {'response': ['123', 'abc']}]

CHICAGO
6 - Pull
7 - Unregister Peer and exit!
Choose option-->
[22:08:51]: [Node chicago accepted connection from ('127.0.1.1', 40470)] 6
[22:08:51]: [Handling the incoming request : {'requestName': 'pull', 'caller': 'paris', 'topic': 'Welcome'}]
['123', 'abc']
{'response': ['123', 'abc']}
[[22:08:51]: [Finished incoming request from paris. Response back to peer node : {'response': ['123', 'abc']}]] 8
Message: {'Welcome': ['123', 'abc']}
Subscriptions: {'Welcome': ['paris', 'london']}

LONDON
[21:15:08]: [Connection established with ('127.0.1.1', 7436)]
[21:15:08]: [Node london sending request {'requestName': 'subscribe', 'caller': 'london', 'topic': 'Welcome'} to Peer at 127.0.1.1 : 7436]
[21:15:08]: [Node london received reply: {'response': 'Peer london is subscribed to Welcome!'}] 9

Message : {'CS550': ['HelloFromParis']}
Subscriptions : {'CS550': []}
2 - CreateTopic
3 - DeleteTopic
4 - SendMessage to topic
5 - Subscribe
6 - Pull
7 - Unregister Peer and exit!
Choose option-->

```

After London pulls from the topic Welcome, the node Chicago calls garbage collections and all messages are gone. Then, Paris pull again from Welcome topic and gets empty response as we expected.

1. London send request to Chicago to pull from Welcome
2. Since all subscribers have read the messages, Chicago flushes all messages and garbage collected
3. London receives **123, abc** which it called previously
4. Paris pull again from topic Welcome and gets empty message. It works correct Great!

```

Topic: {'Welcome': 'chicago', 'C5550': 'london', 'personalTopic': 'paris'}
Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}
[22:18:15] : [Request: {'requestName': 'pull', 'caller': 'london', 'topic': 'Welcome'}]
Topic: {'Welcome': 'chicago', 'C5550': 'london', 'personalTopic': 'paris'}
Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}
[22:18:15] : [Request: {'requestName': 'pull', 'caller': 'paris', 'topic': 'Welcome'}]
Topic: {'Welcome': 'chicago', 'C5550': 'london', 'personalTopic': 'paris'}
Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}
[22:19:52] : [Request: {'requestName': 'pull', 'caller': 'paris', 'topic': 'Welcome'}]
Topic: {'Welcome': 'chicago', 'C5550': 'london', 'personalTopic': 'paris'}
Active Users: {'chicago': ('127.0.1.1', 7436), 'london': ('127.0.1.1', 5510), 'paris': ('127.0.1.1', 5191)}

PARIS
PULL
AGAIN
[22:19:52] : [Node paris called request : {'requestName': 'pull', 'caller': 'paris'}[8/348]
c : 'Welcome']
[22:19:52] : [Response from server : Forwarded to chicago]
4
Sending message to Peer
[22:19:52] : [Connection established with ('127.0.1.1', 7436)]
[22:19:52] : [Node paris sending request {'requestName': 'pull', 'caller': 'paris', 'topic': 'Welcome'} to Peer at 127.0.1.1 : 7436]
[22:19:52] : [Node paris received reply: {'response': 'Empty'}]
Empty 5
Message : {'personalTopic': ['12345']}
Subscriptions : {'personalTopic': []}
[22:18:15] : [Node london accepted connection from ('127.0.0.1', 40474)]
[22:18:15] : [Handling the incoming request : {'requestName': 'pull', 'caller': 'london', 'topic': 'Welcome'}]
7
['123', 'abc']
{'response': ['123', 'abc']}
Garbage collected!
[[22:18:15]] : [Finished incoming request from london. Response back to peer node: {'response': ['123', 'abc']}]
Message: ['Welcome': []]
Flushed 3
Subscriptions: {'Welcome': ['paris', 'london']}
[22:19:52] : [Node chicago accepted connection from ('127.0.0.1', 40474)]
LONDON
READ
[22:18:15] : [Connection established with ('127.0.1.1', 7436)]
[22:18:15] : [Node london sending request {'requestName': 'pull', 'caller': 'london', 'topic': 'Welcome'} to Peer at 127.0.1.1 : 7436]
[22:18:15] : [Node london received reply: {'response': ['123', 'abc']}]
2
Message : {'C5550': ['HelloFromParis']}
Subscriptions : {'C5550': []}
2 - CreateTopic
3 - DeleteTopic
4 - SendMessage to topic
5 - Subscribe
6 - Pull
7 - Unregister Peer and exit!
Choose option-->

```

## f. Unregister API.

When the peer node unregisters, it's asked 2 questions whether it wants to delete everything (1) or transfer topics to other peers(2). In the picture below, it chooses to delete everything. As you can see, Initial and after. The topic of Chicago hello was in the directory of the index server, but after it got deleted and Chicago was removed from list of active peers

```

('response': 'Approved')
Topic: {'hello': 'chicago'}
Initial
Active Users: {'chicago': ('127.0.1.1', 7923)}
[23:30:39] : [Request: {'requestName': 'send', 'caller': 'chicago', 'topic': 'hello', 'message': '12345'}]
Topic: {'hello': 'chicago'}
Active Users: {'chicago': ('127.0.1.1', 7923)}
[23:30:41] : [Request: {'requestName': 'unregisterPeer', 'nodeToTransfer': 'None', 'peerName': 'chicago', 'messageBuffer': {'hello': ['12345']}, 'subscriptions': {'hello': []}, 'readhistory': {}}]
2
Topic: {}
After
Active Users: {}

3 - DeleteTopic
4 - SendMessage to topic
5 - Subscribe
6 - Pull
7 - Unregister Peer and exit!
Choose option-->

what you want to do ?
1- Delete existing topics
2- Transfer existing topics
-->1
Peer
Server
issacode@DESKTOP-J6HG0H2:~/secondPA/02_Muradli_Isa_PA#2$

1
[23:30:41] : [Node chicago called request: unregisterPeer]
8
[23:30:41] : Response from server : chicago has been deleted!
[23:30:41] : [Shutting the node chicago]
[23:30:41] : [Exiting from peer chicago]
G1: Oracle*
"DESKTOP-J6HG0H2" 23:30 12-Oct-24

```

If it chooses to transfer, then the user is prompted to specify the peer name it wants to transfer all the topic information. As you can see all data has been transferred to peer London

- 1- Chicago chooses London as a peer to transfer all info
- 2- It creates the request to the index server.

- 3- The Index server receives the request and finds the credentials of the London
- 4- Chicago establishes the connection with London
- 5- London accepts the incoming connection
- 6- Chicago send all information to London
- 7- London handles the request
- 8- London send back the confirmation of the request
- 9- Chicago receives the confirmation
- 10- London populates its buffers with the information from the request

```

Topic: {'CS550': 'chicago'}

Active Users: {'london': ('127.0.1.1', 5080), 'chicago': ('127.0.1.1', 7560)}

[23:37:30] : [Request: {'requestName': 'deleteTopic', 'caller': 'london', 'topic': '2'}]

Topic: {'CS550': 'chicago'} Initial

Active Users: {'london': ('127.0.1.1', 5080), 'chicago': ('127.0.1.1', 7560)}

③ [23:37:37] : [Request: {'requestName': 'unregisterPeer', 'nodeToTransfer': 'london', 'peerName': 'chicago', 'messageBuffer': {'CS550': ['Good morning']}, 'subscriptions': {'CS550': []}, 'readHistory': {}}]

Topic: {'CS550': 'london'} After

Active Users: {'london': ('127.0.1.1', 5080)}


What node you want to transfer topics to --> london ① 2
[23:37:37] : [Node chicago called request: unregisterPeer] ②

[23:37:37] : Response from server : Node chicago has been transferred to london. chicago has been deleted ③

[23:37:37] : [All topics transferred to london]
Sending message to Peer ④

[23:37:37] : [Connection established with ('127.0.1.1', 5080)] ⑤

[23:37:37] : [Node chicago sending request {'requestName': 'unregisterPeer', 'nodeToTransfer': 'london', 'peerName': 'chicago', 'messageBuffer': {'CS550': ['Good morning']}, 'subscriptions': {'CS550': [], 'readHistory': {}}}] ⑥

[23:37:37] : [Node chicago received reply: {'response': 'All data has been transferred!'}] ⑦

[4/1918] 2 - CreateTopic
3 - DeleteTopic
4 - SendMessage to topic
5 - Subscribe
6 - Pull
7 - Unregister Peer and exit!
Choose option-->
[23:37:37]: [Node london accepted connection from ('127.0.0.1', 55862)] ⑧

[23:37:37]: [Handling the incoming request : {'requestName': 'unregisterPeer', 'nodeToTransfer': 'london', 'peerName': 'chicago', 'messageBuffer': {'CS550': ['Good morning']}, 'subscriptions': {'CS550': []}, 'readHistory': {}}] ⑨

[[23:37:37]] : [Finished incoming request from chicago. Response back to peer node: {'response': 'All data has been transferred!'}] ⑩

Message: {'CS550': ['Good morning']}
Subscriptions: {'CS550': []}

```

**Ensuring multiple peer node can simultaneously publish and subscribe to a topic**

Here peer node **musa** request to subscribe to the topic of node **Isa**. As it can be seen from the screen shot, right bottom is the node **musa** is requesting to subscribe to topic bA, why node **isa** is constantly creating topic in his end. While creating the topic, **peer Isa** can also execute the request from the **peed Musa** and subscribe him to one of his topic. It worked!

Let's consider when 2 nodes(Paris, London) subscribing simultaneously to topics, while 1 node(Chicago) is publishing topics. When one of 2 nodes that want to subscribe to a topic, server will provide the connection credentials of the node that holds that topic - C. After that node P or L will establish connection with node C and node C will receive the same request and execute it while it constantly publishing the topics.

**Red** - As you can observe, node C sending continuous create topic requests to the server. Then, it receives node P requests simultaneously subscribe to a topic, once server receives it and check in the buffer if node C is the holder, it sends back the credentials -09:57:31

09:57:31 - Immediately node P forwards the request to node C

09:57:31 – Node C receives it (top right ) and fulfills the request. In the response it's stating that Peer Paris is subscribed to lk!

After Peer P receives the same response and subscribe function finished printed out.  
Same scenario applies to the blue mentioned circles.

**Note: Peer Chicago simultaneously handles the publishing the topic and handling the incoming subscribe requests from the peers Paris and London.**

**Server**

```

pic': 'Lz'}
[09:57:31] : [Request: {'requestName': 'createTopic', 'caller': 'chicago', 'peerToWrite': 'chicago', 'topic': 'ME'}]
[09:57:31] : [Request: {'requestName': 'subscribe', 'caller': 'paris', 'topic': 'IK'}]
[09:57:31] : [Request: {'requestName': 'createTopic', 'caller': 'chicago', 'peerToWrite': 'chicago', 'topic': 'Hg'}]
[09:57:31] : [Request: {'requestName': 'subscribe', 'caller': 'london', 'topic': 'OT'}]
[09:57:31] : [Request: {'requestName': 'createTopic', 'caller': 'chicago', 'peerToWrite': 'chicago', 'topic': 'bt'}]
[09:57:31] : [Request: {'requestName': 'subscribe', 'caller': 'paris', 'topic': 'ER'}]
[09:57:31] : [Subscribing to topic OT]
[09:57:31] : [Node london called request : {'requestName': 'subscribe', 'caller': 'london', 'topic': 'OT'}]
[09:57:31] : [Response from server : Forwarded to Topic Holder chicago]
[09:57:31] : [Subscribing to topic fg]

```

**CHICAGO pub!**

**Server**

```

[09:57:31] : [Node chicago accepted connection from ('127.0.0.1', 48766)]
Creating topic...
[09:57:31] : [Handling the incoming request : {'requestName': 'subscribe', 'caller': 'paris', 'topic': 'IK'}]
[09:57:31] : [Node chicago called request : {'requestName': 'createTopic', 'caller': 'chicago', 'peerToWrite': 'chicago', 'topic': 'bt'}]
[[09:57:31]] : [Finished incoming request. Response back to peer node: {'response': 'Peer paris is subscribed to IK!'}]
[09:57:31] : [Node chicago accepted connection from ('127.0.0.1', 48770)]
[09:57:31] : [Handling the incoming request : {'requestName': 'subscribe', 'caller': 'london', 'topic': 'OT'}]
[[09:57:31]] : [Finished incoming request. Response back to peer node: {'response': 'Peer london is subscribed to OT!'}]
[09:57:31] : [Node paris received reply: {'response': 'Peer paris is subscribed to CD!'}]
[09:57:31] : [Subscribed function finished!]
[09:57:31] : [Subscribing to topic IK]
[09:57:31] : [Node paris called request : {'requestName': 'subscribe', 'caller': 'paris', 'topic': 'IK'}]
[09:57:31] : [Response from server : Forwarded to Topic Holder chicago]
[09:57:31] : [Subscribing to topic fg]

```

**PARIS Subs**

**London sub**

## Measuring the average response time

**Note:** each node makes 100,000 requests. Please be aware when benchmarking average response time, server first needs to populate the topics buffer with data. So, give it some time. You will be ready to call benchmark for queries once it shows Server is running on (IP, Port). Every node submits 100,000 requests. We created dummy data in the serverForBenchmarking 1 million random topic. Simultaneously multiple nodes query the index server for random topic names and doing so we calculate the average response time. We can see that as the number of nodes increase, the average response time also increase. It's also due to the fact that we use the dictionary which gives us  $O(1)$  access time and server is local. If server was somewhere outside of local network and topics were stored in different data structure, they it could be very long time to get the response. Average response time will not be sweet.

### 2 Peer nodes: Average = 1.937747e-08

```
[21:12:39] : [Response from Index Server: No such user or topic]
Average responses time = 2.4836063385009765e-08

[21:12:39] : [Response from Index Server: No such user or topic]
[21:12:39] : [Response from Index Server: No such user or topic]

Average responses time = 1.3918876647949219e-08
```

### 4 Peer nodes: Average= 3.14605274e-08

```
[21:44:10] : [Response from Index Server: No such user or topic]
Average responses time = 3.953695297241211e-08

[21:44:11] : [Response from Index Server: No such user or topic]
Average responses time = 4.76837158203125e-08

[21:44:12] : [Response from Index Server: No such user or topic]
Average responses time = 1.4998912811279295e-08

[21:44:11] : [Response from Index Server: No such user or topic]
[21:44:11] : [Response from Index Server: No such user or topic]

Average responses time = 2.3622512817382813e-08
```

### 8 Peer nodes: Average= 3.10340522e-08

```
[22:04:44] : [Response from Index Server: No such user or topic]
[22:04:44] : [Response from Index Server: No such user or topic]
Average responses time = 3.6563873291015624e-08

[22:04:52] : [Response from Index Server: No such user or topic]
[22:04:52] : [Response from Index Server: No such user or topic]
Average responses time = 1.6703605651855468e-08

[22:04:46] : [Response from Index Server: No such user or topic]
[22:04:46] : [Response from Index Server: No such user or topic]
Average responses time = 3.963708877563477e-08

[22:04:47] : [Response from Index Server: No such user or topic]
[22:04:47] : [Response from Index Server: No such user or topic]
Average responses time = 4.4679641723632815e-08

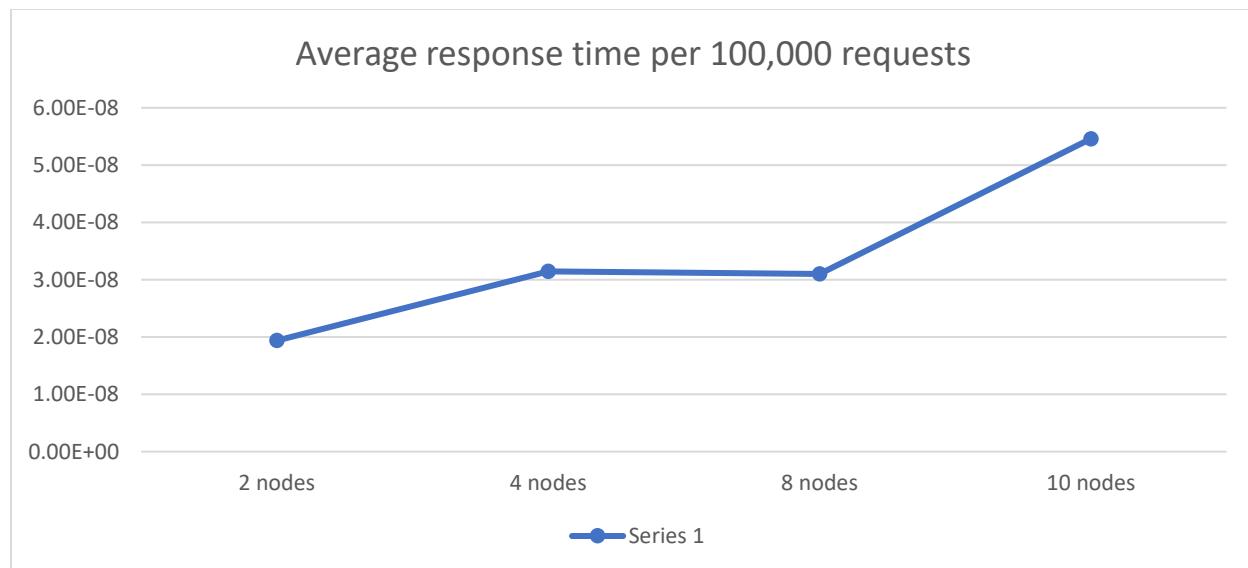
[22:04:51] : [Response from Index Server: No such user or topic]
[22:04:51] : [Response from Index Server: No such user or topic]
Average responses time = 1.878499984741211e-08

[22:04:47] : [Response from Index Server: No such user or topic]
[22:04:47] : [Response from Index Server: No such user or topic]
Average responses time = 2.8150081634521483e-08
```

```
[22:04:50] : [Response from Index Server: icFy] [22:04:50] : [Response from Index Server: No such user or topic]
Average responses time = 4.36854362487793e-08 Average responses time = 2.0067691802978514e-08
```

## 10 Peer nodes: Average= 5.46002421e-08

```
[22:30:28] : [Response from Index Server: No such user or topic] [22:30:38] : [Response from Index Server: No such user or topic]
Average responses time = 7.421016693115234e-08 Average responses time = 5.64122200012207e-08
[22:30:14] : [Response from Index Server: No such user or topic] [22:30:32] : [Response from Index Server: No such user or topic]
Average responses time = 7.144212722778321e-08 Average responses time = 5.2685737609863284e-08
[22:30:42] : [Response from Index Server: No such user or topic] [22:30:37] : [Response from Index Server: No such user or topic]
Average responses time = 2.8297901153564454e-08 Average responses time = 7.03597068786621e-08
[22:30:42] : [Response from Index Server: No such user or topic] [22:30:42] : [Response from Index Server: No such user or topic]
Average responses time = 2.183675765991211e-08 Average responses time = 2.6478767395019533e-08
[22:30:31] : [Response from Index Server: No such user or topic] [22:30:42] : [Response from Index Server: No such user or topic]
Average responses time = 9.16290283203125e-08 Average responses time = 5.264759063720703e-08
```



## Benchmarking

<b>Requests</b>	<i>Create</i>	<i>Delete</i>	<i>Send</i>	<i>Subscribe</i>	<i>Pull</i>
<b>1 Node</b>	<b>54720</b>	<b>87323</b>	<b>61516</b>	<b>70646</b>	<b>101631</b>
<b>2 Node</b>	<b>74243</b>	<b>59127</b>	<b>67330</b>	<b>54371</b>	<b>118180</b>
<b>3 Node</b>	<b>61050</b>	<b>64534</b>	<b>71953</b>	<b>59352</b>	<b>62650</b>
<b>4 Node</b>	<b>46841</b>	<b>44737</b>	<b>40552</b>	<b>46853</b>	<b>44724</b>
<b>5 Node</b>	<b>33562</b>	<b>38909</b>	<b>21247</b>	<b>24609</b>	<b>31356</b>
<b>6 Node</b>	<b>31956</b>	<b>47095</b>	<b>25591</b>	<b>29428</b>	<b>45787</b>
<b>7 Node</b>	<b>35600</b>	<b>43421</b>	<b>36086</b>	<b>37194</b>	<b>42308</b>
<b>8 Node</b>	<b>33190</b>	<b>30162</b>	<b>31691</b>	<b>27542</b>	<b>36311</b>

<b>Throughput</b>	<i>Create</i>	<i>Delete</i>	<i>Send</i>	<i>Subscribe</i>	<i>Pull</i>
<b>1 Node</b>	<b>182.4</b>	<b>291.07</b>	<b>205.05</b>	<b>235.48</b>	<b>338.77</b>
<b>2 Node</b>	<b>247</b>	<b>196</b>	<b>224</b>	<b>181</b>	<b>394</b>
<b>3 Node</b>	<b>203.5</b>	<b>215</b>	<b>239</b>	<b>198</b>	<b>208.5</b>
<b>4 Node</b>	<b>156.3</b>	<b>148.7</b>	<b>135</b>	<b>155.6</b>	<b>149.3</b>
<b>5 Node</b>	<b>111.85</b>	<b>129.14</b>	<b>70.432</b>	<b>81.95</b>	<b>113.97</b>
<b>6 Node</b>	<b>107.96</b>	<b>154.8</b>	<b>82.73</b>	<b>98.61</b>	<b>150.9</b>
<b>7 Node</b>	<b>116.7</b>	<b>144.74</b>	<b>119.28</b>	<b>123.04</b>	<b>141.6</b>
<b>8 Node</b>	<b>109.90</b>	<b>101.23</b>	<b>105.606</b>	<b>94.486</b>	<b>121.79</b>

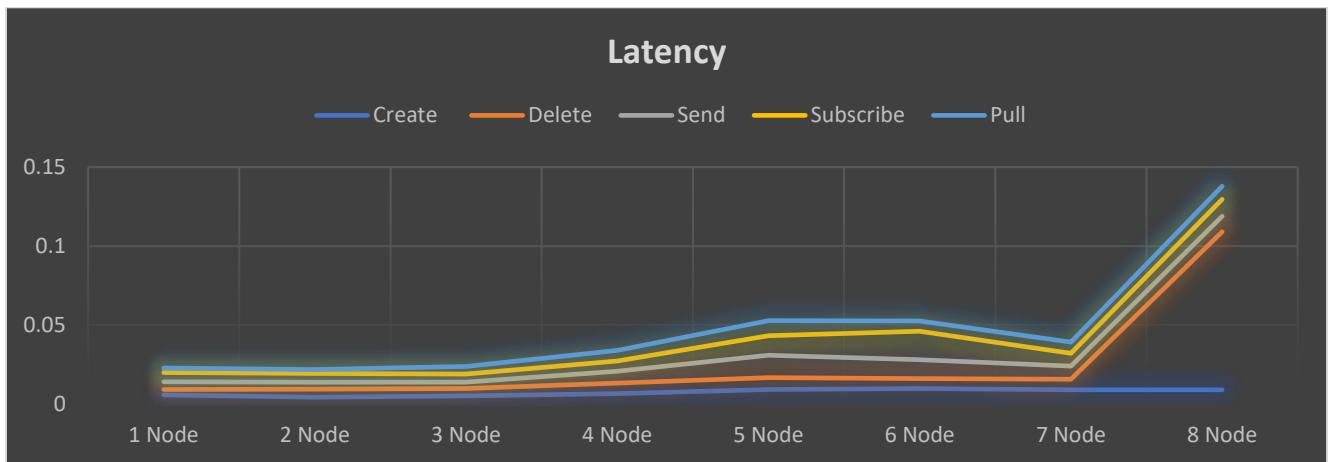
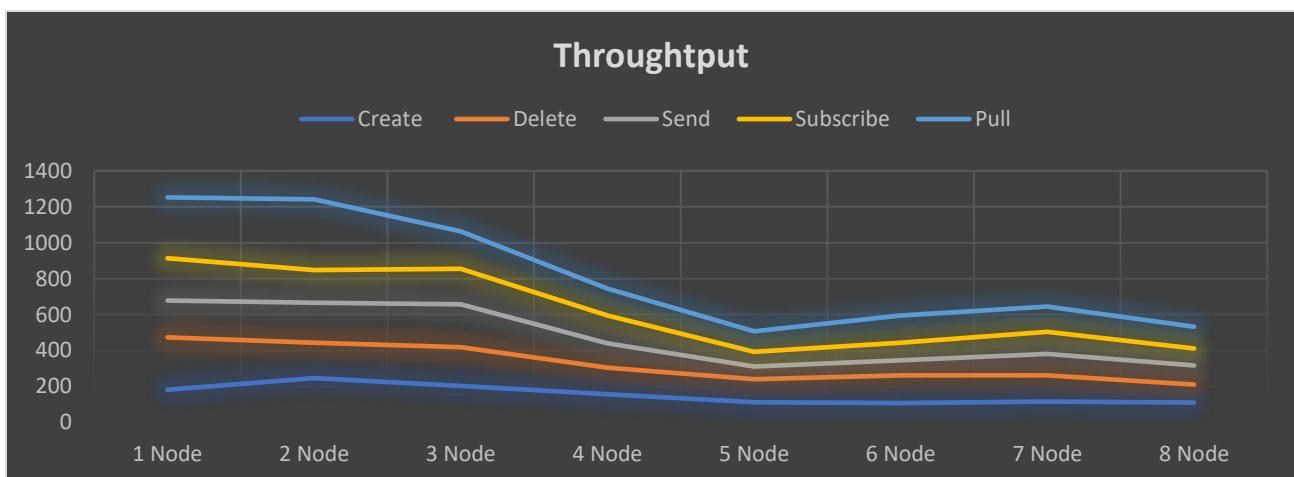
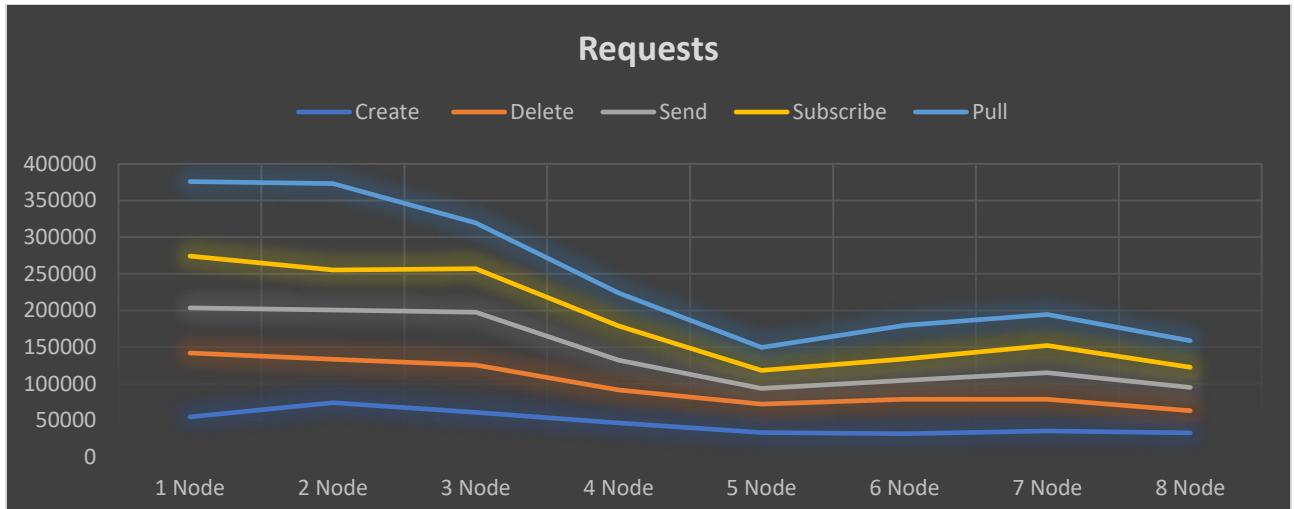
Note that latency has been rounded at 5<sup>th</sup> digit after the decimal point

<b>Latency</b>	<i>Create</i>	<i>Delete</i>	<i>Send</i>	<i>Subscribe</i>	<i>Pull</i>
<b>1 Node</b>	<b>0.00548</b>	<b>0.00343</b>	<b>0.00487</b>	<b>0.00587</b>	<b>0.00295</b>
<b>2 Node</b>	<b>0.00403</b>	<b>0.00507</b>	<b>0.00445</b>	<b>0.00551</b>	<b>0.00253</b>
<b>3 Node</b>	<b>0.00409</b>	<b>0.00463</b>	<b>0.00415</b>	<b>0.00504</b>	<b>0.00478</b>
<b>4 Node</b>	<b>0.00635</b>	<b>0.00673</b>	<b>0.00743</b>	<b>0.00649</b>	<b>0.00671</b>
<b>5 Node</b>	<b>0.00886</b>	<b>0.00769</b>	<b>0.01418</b>	<b>0.01223</b>	<b>0.00972</b>
<b>6 Node</b>	<b>0.00953</b>	<b>0.00639</b>	<b>0.01191</b>	<b>0.01802</b>	<b>0.00665</b>
<b>7 Node</b>	<b>0.00864</b>	<b>0.00685</b>	<b>0.00831</b>	<b>0.00803</b>	<b>0.00709</b>
<b>8 Node</b>	<b>0.00881</b>	<b>0.10320</b>	<b>0.00983</b>	<b>0.01076</b>	<b>0.00829</b>

First when we have up to 3 nodes, most of the API requests remains the almost the same, but starting from 4 nodes concurrently sending APIs to the index server. Due to the fact that it puts overload on the server to switch and fulfill the requests, it's not surprising that after one point server will decrease in productivity even if it supports concurrency. Having to switch every time will cost more time than actual time needed to finish the request.

Similarly, as the number of requests decrease as number of nodes increase, throughput will be affected as well. It shows that starting from 5 nodes, it declines.

Additionally, as the throughput decrease, it will consequently lead to the more latency in the system. All incoming requests should be addressed so that nodes might need to wait for their turn for server to take care. Therefore, the more requests in the system, the more you might need to wait.



## 1 Peer node

### a. Create Topic

```
Creating topic...
[10:37:08] : f[Node isa called request : {'requestName': 'createTopic', 'caller': 'isa', 'peerToWrite': 'isa', 'topic': 'jsQx'}]
[10:37:08] : [Response from server: Approved]

Number of request : 54720, timePassed: 5 mins, throughput : 182.4 Latency = 0.005479630535980415
"DESKTOP-J6HG0H2" 10:37 13-Oct-24
```

### b. Delete Topic

```
Deleting topic...
[10:52:08] : [Node isa called request : {'requestName': 'deleteTopic', 'caller': 'isa', 'topic': 'xYss'}]
[10:52:08] : [Response from server: Topic xYss does not exist]

Number of request : 87323, timePassed: 5 mins, throughput : 291.07666666666665 Latency = 0.003433894072475
"DESKTOP-J6HG0H2" 10:52 13-Oct-24
```

### c. Send

```
Sending message XuVr to ngAQ
[11:05:14] : [Node isa called request : {'requestName': 'send', 'caller': 'isa', 'topic': 'ngAQ', 'message': 'XuVr'}]
[11:05:14] : [Response from server: Topic ngAQ does not exist]

Number of request : 61516, timePassed: 5 mins, throughput : 205.05333333333334 Latency = 0.004874404656173148
"DESKTOP-J6HG0H2" 11:05 13-Oct-24
```

### d. Subscribe

```
Subscribed function finished!
```

```
Number of request : 70646, timePassed: 5 mins, throughput : 235.48666666666668 Latency = 0.004244785992
"DESKTOP-J6HG0H2" 11:05 13-Oct-24
```

### e. Pull

```
Pulling messages from WfMo
[11:23:10] : [Node isa called request : {'requestName': 'pull', 'caller': 'isa', 'topic': 'WfMo'}]
[11:23:10] : [Response from server: Topic WfMo does not exist]

Number of request : 101631, timePassed: 5 mins, throughput : 338.77 Latency = 0.0029503138682245708
"DESKTOP-J6HG0H2" 11:23 13-Oct-24
```

## 2 Peer Nodes

For the benchmarking of 2 or more peers, we will consider their average of nodes connected for the graph!

### a. Create Topic

```
Creating topic...
[11:32:21] : f[Node Peer1] called request : {'requestName': 'createTopic', 'caller': 'Peer1', 'peerToWrite': 'Peer1', 'topic': 'fYYz'}]
[11:32:21] : [Response from server: Approved]
```

Number of request : 74618, timePassed: 5 mins, throughput : 248.7266666666666 Latency = 0.004018766877670961

```
Creating topic...
[11:32:21] : f[Node Peer2] called request : {'requestName': 'createTopic', 'caller': 'Peer2', 'peerToWrite': 'Peer2', 'topic': 'SaOM'}]
[11:32:21] : [Response from server: Approved]
```

Number of request : 73869, timePassed: 5 mins, throughput : 246.23 Latency = 0.00405907566691441

### b. Delete Topic

```
Deleting topic...
[12:05:03] : [Node Peer1 called request : {'requestName': 'deleteTopic', 'caller': 'Peer1', 'topic': 'fTGy'}]
[12:05:03] : [Response from server: Topic fTGy does not exist]
```

Number of request : 59349, timePassed: 5 mins, throughput : 197.83 Latency = 0.005053744589031676

```
Deleting topic...
[12:05:06] : [Node Peer2 called request : {'requestName': 'deleteTopic', 'caller': 'Peer2', 'topic': 'aTaq'}]
[12:05:06] : [Response from server: Topic aTaq does not exist]
```

Number of request : 58905, timePassed: 5 mins, throughput : 196.35 Latency = 0.005091185719925806

### c. Send

```
Sending message XDBC to Kjqq
[12:14:48] : [Node Peer2 called request : {'requestName': 'send', 'caller': 'Peer2', 'topic': 'Kjqq', 'message': 'XDBC'}]
[12:14:48] : [Response from server: Topic Kjqq does not exist]
```

Number of request : 67361, timePassed: 5 mins, throughput : 224.5366666666666 Latency = 0.004451442505631243

```
Sending message peSD to XLvF
[12:14:49] : [Node Peer1 called request : {'requestName': 'send', 'caller': 'Peer1', 'topic': 'XLvF', 'message': 'peSD'}]
[12:14:49] : [Response from server: Topic XLvF does not exist]
```

Number of request : 67299, timePassed: 5 mins, throughput : 224.33 Latency = 0.0044558169762091565

#### d. Subscribe

```
Subscribed function finished!  
Number of request : 54117, timePassed: 5 mins, throughput : 180.39 Latency = 0.005541720098580107
```

```
Subscribed function finished!  
[12:24:18] : [Subscribing to topic yGlc]  
[12:24:18] : You can't subscribe to your own topic  
Number of request : 54626, timePassed: 5 mins, throughput : 182.08666666666667 Latency = 0.005489815338997063
```

#### e. Pull

```
Pulling messages from MNwZ  
[12:47:17] : [Node Peer1 called request : {'requestName': 'pull', 'caller': 'Peer1', 'topic': 'MNwZ'}]  
[12:47:17] : [Response from server: Topic MNwZ does not exist]  
Number of request : 118374, timePassed: 5 mins, throughput : 394.58 Latency = 0.0025326527526347334
```

```
Pulling messages from JMbX  
[12:47:18] : [Node Peer2 called request : {'requestName': 'pull', 'caller': 'Peer2', 'topic': 'JMbX'}]  
[12:47:18] : [Response from server: Topic JMbX does not exist]  
Number of request : 117987, timePassed: 5 mins, throughput : 393.29 Latency = 0.002540888888119744
```

[second\_se0:python3\*

## 3 Peer Nodes

#### a. Create Topic

```
Creating topic...  
[12:57:18] : f[Node Peer1 called request : {'requestName': 'createTopic', 'caller': 'Peer1', 'topic': 'yGlc'}]  
[12:57:18] : [Response from server: Topic yGlc created]  
Number of request : 60973,  
timePassed: 5 mins,  
throughput : 203.2433333333334  
Latency = 0.004918255324306854
```

```
Creating topic...  
[12:57:18] : f[Node Peer2 called request : {'requestName': 'createTopic', 'caller': 'Peer2', 'topic': 'yGlc'}]  
[12:57:18] : [Response from server: Topic yGlc created]  
Number of request : 61036,  
timePassed: 5 mins,  
throughput : 203.45333333333335  
Latency = 0.004913256594765834
```

```
Creating topic...  
[12:57:18] : f[Node Peer3 called request : {'requestName': 'createTopic', 'caller': 'Peer3', 'topic': 'yGlc'}]  
[12:57:18] : [Response from server: Topic yGlc created]  
Number of request : 61167,  
timePassed: 5 mins,  
throughput : 203.89  
Latency = 0.004902525440961766
```

## b. Delete Topic

<b>Deleting topic...</b> [13:07:41] : [Node Peer1 called request : {'requestName': 'deleteTopic', 'caller': 'Peer1', 'topic': 'Mfet'}]  Number of request : 64620, timePassed: 5 mins, throughput : 215.4 Latency = 0.00464117810365328	<b>Deleting topic...</b> [13:07:44] : [Node Peer2 called request : {'requestName': 'deleteTopic', 'caller': 'Peer2', 'topic': 'Mfet'}]  Number of request : 64787, timePassed: 5 mins, throughput : 215.95666666666666 Latency = 0.004629096075599655	<b>Deleting topic...</b> [13:07:45] : [Node Peer3 called request : {'requestName': 'deleteTopic', 'caller': 'Peer3', 'topic': 'Mfet'}]  Number of request : 64445, timePassed: 5 mins, throughput : 214.81666666666666 Latency = 0.00465378781639121
--	--	---

## c. Send

**Sending message qmKO to Mfet**

```
[13:19:09] : [Node Peer1 called request : {'requestName': 'send', 'caller': 'Peer1', 'topic': 'Mfet', 'message': 'qmKO'}]
[13:19:09] : [Response from server: Topic Mfet does not exist]

Number of request : 72022, timePassed: 5 mins, throughput : 240.0733333333332 Latency = 0.004163706009014098
```

**Sending message GupN to LSHH**

```
[13:19:09] : [Node Peer2 called request : {'requestName': 'send', 'caller': 'Peer2', 'topic': 'LSHH', 'message': 'GupN'}]
[13:19:09] : [Response from server: Topic LSHH does not exist]

Number of request : 72377, timePassed: 5 mins, throughput : 241.25666666666666 Latency = 0.004143136733883771
```

**Sending message ORUl to HvVy**

```
[13:19:11] : [Node Peer3 called request : {'requestName': 'send', 'caller': 'Peer3', 'topic': 'HvVy', 'message': 'ORUl'}]
[13:19:11] : [Response from server: Topic HvVy does not exist]

Number of request : 71673, timePassed: 5 mins, throughput : 238.91 Latency = 0.00418435032167153
```

## d. Subscribe

<b>Subscribing to top...</b> [13:30:16] : [Node Peer1 called request : {'requestName': 'subscribe', 'topic': 'top'}]  Subscribed function finished!  Number of request : 59593, timePassed: 5 mins, throughput : 198.6433333333335 Latency = 0.005032438960059456	<b>Subscribing to top...</b> [13:30:17] : [Node Peer2 called request : {'requestName': 'subscribe', 'topic': 'top'}]  Subscribed function finished!  Number of request : 59444, timePassed: 5 mins, throughput : 198.14666666666666 Latency = 0.005044957440519195	<b>Subscribed function finished!</b>  Number of request : 59268, timePassed: 5 mins, throughput : 197.56 Latency = 0.005059838130051383
--	---	---

## e. Pull

<b>Pulling messages from jpgg</b> [13:42:42] : [Node Peer1 called request : {'requestName': 'pull', 'topic': 'jpgg'}]  Number of request : 62624, timePassed: 5 mins, throughput : 208.74666666666667 Latency = 0.004788338441323673	<b>Pulling messages from sffi</b> [13:42:41] : [Node Peer2 called request : {'requestName': 'pull', 'topic': 'sffi'}]  Number of request : 62593, timePassed: 5 mins, throughput : 208.6433333333335 Latency = 0.004790626038353611	<b>Pulling messages from VJys</b> [13:42:43] : [Node Peer3 called request : {'requestName': 'pull', 'topic': 'VJys'}]  Number of request : 62699, timePassed: 5 mins, throughput : 208.99666666666667 Latency = 0.004782405975197725
---	--	---

## 4 Peer Nodes

### a. Create Topic

Creating topic...  [13:54:04] : f[Node <u>Peer1</u> called re  [13:54:04] : [Response from Node Pe  Number of request : 46633, timePassed: 5 mins, throughput : 155.44333333333333 Latency = 0.006431219945031529	Creating topic...  [13:54:04] : f[Node <u>Peer2</u> called re  [13:54:04] : [Response from Node Pe  Number of request : 46710, timePassed: 5 mins, throughput : 155.7 Latency = 0.0064207562949054635
Creating topic...  [13:54:04] : f[Node <u>Peer3</u> called re  [13:54:04] : [Response from Node Pe  Number of request : 47253, timePassed: 5 mins, throughput : 157.51 Latency = 0.006346774015987098	Creating topic...  [13:54:04] : f[Node <u>Peer4</u> called re  [13:54:04] : [Response from Node Pe  Number of request : 47063, timePassed: 5 mins, throughput : 156.87666666666667 Latency = 0.006372111644533725

### b. Delete Topic

Deleting topic...  [14:06:59] : [Node <u>Peer1</u> called re  [14:06:59] : [Response from Node Pe  Number of request : 44477, timePassed: 5 mins, throughput : 148.25666666666666 Latency = 0.006743407001259929	Deleting topic...  [14:06:59] : [Node <u>Peer2</u> called re  [14:06:59] : [Response from Node Pe  Number of request : 44964, timePassed: 5 mins, throughput : 149.88 Latency = 0.006670254541561173
Deleting topic...  [14:06:59] : [Node <u>Peer3</u> called re  [14:06:59] : [Response from Node Pe  Number of request : 44641, timePassed: 5 mins, throughput : 148.8033333333334 Latency = 0.006717939839985088	Deleting topic...  [14:07:00] : [Node <u>Peer4</u> called re  [14:07:00] : [Response from Node Pe  Number of request : 44657, timePassed: 5 mins, throughput : 148.85666666666665 Latency = 0.006716093443739357

### c. Send

Sending message BhkS to glgk [14:18:46] : [Node Peer1 called request] [14:18:46] : [Response from server]  Number of request : 40534, timePassed: 5 mins, throughput : 135.1133333333333 Latency = 0.007399205039930595	Sending message NfPH to tRIJ [14:18:45] : [Node Peer2 called request] [14:18:45] : [Response from server]  Number of request : 40691, timePassed: 5 mins, throughput : 135.63666666666666 Latency = 0.007369927694843745
Sending message YUQY to tFPO [14:18:45] : [Node Peer3 called request] [14:18:45] : [Response from server]  Number of request : 40469, timePassed: 5 mins, throughput : 134.89666666666668 Latency = 0.007410697108211261	Sending message xcoM to dvAT [14:18:46] : [Node Peer4 called request] [14:18:46] : [Response from server]  Number of request : 40704, timePassed: 5 mins, throughput : 135.68 Latency = 0.007367677963192358

### d. Subscribe

[14:31:13] : [Subscribing to topic] [14:31:13] : [Node Peer1 called request] [14:31:13] : [Response from server]  Subscribed function finished!  Number of request : 46445, timePassed: 5 mins, throughput : 154.81666666666666 Latency = 0.0064572565239101294	[14:31:13] : [Node Peer2 called request] [14:31:13] : [Response from server]  Subscribed function finished!  Number of request : 46757, timePassed: 5 mins, throughput : 155.85666666666665 Latency = 0.006413895205206012
[14:31:12] : [Node Peer3 called request] [14:31:12] : [Response from server]  Subscribed function finished!  Number of request : 46999, timePassed: 5 mins, throughput : 156.6633333333333 Latency = 0.006381135450636809	[14:31:13] : [Node Peer4 called request] [14:31:13] : [Response from server]  Subscribed function finished!  Number of request : 46921, timePassed: 5 mins, throughput : 156.40333333333334 Latency = 0.006391716075572097

## e. Pull

Pulling messages from bTCq [14:46:20] : [Node Peer1 called request]  [14:46:20] : [Response from server: Approved]  Number of request : 44584, timePassed: 5 mins, throughput : 148.61333333333334 Latency = 0.006726366248354115	Pulling messages from xauT [14:46:18] : [Node Peer2 called request]  [14:46:18] : [Response from server: Approved]  Number of request : 44740, timePassed: 5 mins, throughput : 149.13333333333333 Latency = 0.006703508529057714
Pulling messages from htPQ [14:46:21] : [Node Peer3 called request]  [14:46:21] : [Response from server: Approved]  Number of request : 44544, timePassed: 5 mins, throughput : 148.48 Latency = 0.006731721267489524	Pulling messages from Imna [14:46:23] : [Node Peer4 called request]  [14:46:23] : [Response from server: Approved]  Number of request : 45498, timePassed: 5 mins, throughput : 151.66 Latency = 0.006590929889297469

## 5 Peer Nodes

### a. Create Topic

Creating topic... [14:54:54] : f[Node Peer1 called request]  [14:54:54] : [Response from server: Approved]  Number of request : 33518, timePassed: 5 mins, throughput : 111.72666666666667 Latency = 0.008948066872069173	Creating topic... [14:54:54] : f[Node Peer2 called request]  [14:54:54] : [Response from server: Approved]  Number of request : 33979, timePassed: 5 mins, throughput : 113.26333333333334 Latency = 0.008826634500701635	Creating topic... [14:54:54] : f[Node Peer3 called request]  [14:54:54] : [Response from server: Approved]  Number of request : 33337, timePassed: 5 mins, throughput : 111.12333333333333 Latency = 0.008996749341203602
Creating topic... [14:54:54] : f[Node Peer4 called request]  [14:54:54] : [Response from server: Approved]  Number of request : 33938, timePassed: 5 mins, throughput : 113.12666666666667 Latency = 0.008836878576137621	Creating topic... [14:54:54] : f[Node Peer5 called request : {'requestName': 'topic1'}]  [14:54:54] : [Response from server: Approved]  Number of request : 34069, timePassed: 5 mins, throughput : 113.56333333333333 Latency = 0.008803480060363194	

## b. Delete Topic

<p><u>Deleting topic...</u> [15:08:08] : [Node <u>Peer1</u> called request] [15:08:08] : [Response from s]</p> <p>Number of request : 38580, timePassed: 5 mins, throughput : 128.6 Latency = 0.0077745162371590446</p>	<p><u>Deleting topic...</u> [15:08:09] : [Node <u>Peer2</u> called] [15:08:09] : [Response f]</p> <p>Number of request : 38762, timePassed: 5 mins, throughput : 129.20666666666668 Latency = 0.007737790779840847</p>	<p><u>Deleting topic...</u> [15:08:04] : [Node <u>Peer3</u> called] [15:08:04] : [Response f]</p> <p>Number of request : 39098, timePassed: 5 mins, throughput : 130.32666666666665 Latency = 0.007671067742619882</p>
<p><u>Deleting topic...</u> [15:08:10] : [Node <u>Peer4</u> called] [15:08:10] : [Response f]</p> <p>Number of request : 38904, timePassed: 5 mins, throughput : 129.68 Latency = 0.0077095837314111466</p>	<p><u>Deleting topic...</u> [15:08:06] : [Node <u>Peer5</u> called] [15:08:06] : [Response f]</p> <p>Number of request : 39653, timePassed: 5 mins, throughput : 132.17666666666668 Latency = 0.007564234738992905</p>	

## c. Send Topic

<p><u>Sending message YdMY to RdZf</u> [15:23:18] : [Node <u>Peer1</u> called] [15:23:18] : [Response f]</p> <p>Number of request : 21379, timePassed: 5 mins, throughput : 71.263333333334 Latency = 0.014029112526395656</p>	<p><u>Sending message aygX to PVDN</u> [15:23:21] : [Node <u>Peer2</u> called request] [15:23:21] : [Response from s]</p> <p>Number of request : 21143, timePassed: 5 mins, throughput : 70.47666666666667 Latency = 0.014187357246731988</p>	<p><u>Sending message xMsU to sdLC</u> [15:23:18] : [Node <u>Peer3</u> called] [15:23:18] : [Response f]</p> <p>Number of request : 21431, timePassed: 5 mins, throughput : 71.43666666666667 Latency = 0.013997022228197356</p>
<p>[15:23:19] : [Node <u>Peer4</u> called] [15:23:19] : [Response f]</p> <p>Number of request : 21272, timePassed: 5 mins, throughput : 70.90666666666667 Latency = 0.01410062316890586</p>	<p><u>Sending message STxC to VlkUX</u> [15:23:22] : [Node <u>Peer5</u> called] [15:23:22] : [Response f]</p> <p>Number of request : 21427, timePassed: 5 mins, throughput : 71.42333333333333 Latency = 0.01399793014894894</p>	

#### d. Subscribe

[15:38:46] : [Node <u>Peer1</u> called [15:38:46] : [Response f  Subscribed function finished!  Number of request : 24350, timePassed: 5 mins, throughput : 81.16666666666667 Latency = 0.01231732790475019	[15:38:51] : [Subscribing to top [15:38:51] : [Node <u>Peer2</u> called [15:38:51] : [Response f  Subscribed function finished!  Number of request : 24592, timePassed: 5 mins, throughput : 81.97333333333333 Latency = 0.012196729118434656	[15:38:44] : [Node <u>Peer3</u> called [15:38:44] : [Response f  Subscribed function finished!  Number of request : 24670, timePassed: 5 mins, throughput : 82.23333333333333 Latency = 0.0121586846783425
[15:38:49] : [Subscribing to top [15:38:49] : [Node <u>Peer4</u> called [15:38:49] : [Response f  Subscribed function finished!  Number of request : 24582, timePassed: 5 mins, throughput : 81.94 Latency = 0.012202213928024643	[15:38:45] : [Subscribing to top [15:38:45] : [Node <u>Peer5</u> called [15:38:45] : [Response f  Subscribed function finished!  Number of request : 24486, timePassed: 5 mins, throughput : 81.62 Latency = 0.012250446994778067	

#### e. Pull

Pulling messages from dchB [16:02:15] : [Node <u>Peer1</u> called re [16:02:15] : [Response fr  Number of request : 36640, timePassed: 5 mins, throughput : 122.133333333334 Latency = 0.008184469409905146	Pulling messages from rLjW [16:02:06] : [Node <u>Peer2</u> called re [16:02:06] : [Response fr  Number of request : 32620, timePassed: 5 mins, throughput : 108.733333333333 Latency = 0.009193760152976986	Pulling messages from jjmb [16:01:57] : [Node <u>Peer3</u> called re [16:01:57] : [Response fr  Number of request : 30356, timePassed: 5 mins, throughput : 101.18666666666667 Latency = 0.009879578045822287
Pulling messages from OOQp [16:02:01] : [Node <u>Peer4</u> called re [16:02:01] : [Response fr  Number of request : 30770, timePassed: 5 mins, throughput : 102.56666666666666 Latency = 0.009746444477134608	Pulling messages from KJmU [16:02:09] : [Node <u>Peer5</u> called re [16:02:09] : [Response fr  Connection established with other  [16:02:09] : [Node Peer5 received Number of request : 33575, timePassed: 5 mins, throughput : 111.91666666666667 Latency = 0.008932391327881404	

## 6 Peer Nodes

### a. Create Topic

<p><u>Creating topic...</u></p> <pre>[16:24:25] : f[Node <u>Peer1</u> called [16:24:25] : [Response fr  Number of request : 35075, timePassed: 5 mins, throughput : 116.91666666666667 Latency = 0.008551765701554965</pre>	<p><u>Creating topic...</u></p> <pre>[16:24:25] : f[Node <u>Peer2</u> called [16:24:25] : [Response fr  Number of request : 31485, timePassed: 5 mins, throughput : 104.95 Latency = 0.009526838308973162</pre>	<p><u>Creating topic...</u></p> <pre>[16:24:25] : f[Node <u>Peer3</u> called r [16:24:25] : [Response : {  Number of request : 31768, timePassed: 5 mins, throughput : 105.89333333333333 Latency = 0.009441360317168744</pre>
<p><u>Creating topic...</u></p> <pre>[16:24:25] : f[Node <u>Peer4</u> called [16:24:25] : [Response fr  Number of request : 31785, timePassed: 5 mins, throughput : 105.95 Latency = 0.009435960491517063</pre>	<p><u>Creating topic...</u></p> <pre>[16:24:25] : f[Node <u>Peer5</u> called [16:24:25] : [Response fr  Number of request : 31438, timePassed: 5 mins, throughput : 104.79333333333334 Latency = 0.00954038206952637</pre>	<p><u>Creating topic...</u></p> <pre>[16:24:25] : f[Node <u>Peer6</u> called [16:24:25] : [Response fr  Number of request : 31565, timePassed: 5 mins, throughput : 105.21666666666667 Latency = 0.009501900703880954</pre>

### b. Delete Topic

<p><u>Deleting topic...</u></p> <pre>[16:37:40] : [Node <u>Peer1</u> called [16:37:40] : [Response fr  Number of request : 47635, timePassed: 5 mins, throughput : 158.7833333333333 Latency = 0.006296485496665945</pre>	<p><u>Deleting topic...</u></p> <pre>[16:37:41] : [Node <u>Peer2</u> called r [16:37:41] : [Response fr  Number of request : 46283, timePassed: 5 mins, throughput : 154.27666666666667 Latency = 0.006479999388191232</pre>	<p><u>Deleting topic...</u></p> <pre>[16:37:39] : [Node <u>Peer3</u> called [16:37:39] : [Response fr  Number of request : 46075, timePassed: 5 mins, throughput : 153.58333333333334 Latency = 0.006509399693489592</pre>
<p><u>Deleting topic...</u></p> <pre>[16:37:42] : [Node <u>Peer4</u> called r [16:37:42] : [Response fr  Number of request : 47569, timePassed: 5 mins, throughput : 158.5633333333333 Latency = 0.00630495676506851</pre>	<p><u>Deleting topic...</u></p> <pre>[16:37:41] : [Node <u>Peer5</u> called re [16:37:41] : [Request for Connection established with other [16:37:41] : [Node <u>Peer5</u> received  Number of request : 47558, timePassed: 5 mins, throughput : 158.52666666666667 Latency = 0.006306512792255125</pre>	<p><u>Deleting topic...</u></p> <pre>[16:37:38] : [Node <u>Peer6</u> called r [16:37:38] : [Response fr  Number of request : 46679, timePassed: 5 mins, throughput : 155.59666666666666 Latency = 0.006425214606995075</pre>

### c. Send Topic

Sending message vfRP to Dbit [16:54:03] : [Node Peer1 called] [16:54:03] : [Response finished]  Number of request : 24912, timePassed: 5 mins, throughput : 83.04 Latency = 0.012039052029412453	Sending message yqou to PNlV [16:54:01] : [Node Peer2 called] [16:54:01] : [Response finished]  Number of request : 25115, timePassed: 5 mins, throughput : 83.71666666666667 Latency = 0.011942426308250846	Sending message liwY to gclm [16:54:05] : [Node Peer3 called] [16:54:05] : [Response finished]  Number of request : 25358, timePassed: 5 mins, throughput : 84.52666666666667 Latency = 0.011828515495596948
Sending message lwyy to aDCc [16:54:03] : [Node Peer4 called] [16:54:03] : [Response finished]  Number of request : 25501, timePassed: 5 mins, throughput : 85.00333333333333 Latency = 0.011761579601060015	[16:54:03] : [Finished incoming message] [16:54:03] : [Node Peer5 called] [16:54:03] : [Response finished]  Number of request : 24974, timePassed: 5 mins, throughput : 83.24666666666667 Latency = 0.012010721636418012	Sending message PHix to qHPz [16:53:56] : [Node Peer6 called] [16:53:56] : [Response finished]  Number of request : 24861, timePassed: 5 mins, throughput : 82.87 Latency = 0.01206561910200328

### d. Subscribe

[17:10:03] : [Subscribing to topic] [17:10:03] : [Node Peer1 called] [17:10:03] : [Response finished]  Subscribed function finished!  Number of request : 29307, timePassed: 5 mins, throughput : 97.69 Latency = 0.010233796354276837	[17:10:03] : [Subscribing to topic] [17:10:03] : [Node Peer2 called] [17:10:03] : [Response finished]  Subscribed function finished!  Number of request : 29883, timePassed: 5 mins, throughput : 99.61 Latency = 0.01003639113131933	[17:10:18] : [Subscribing to topic] [17:10:18] : [Node Peer3 called] [17:10:18] : [Response finished]  Subscribed function finished!  Number of request : 30117, timePassed: 5 mins, throughput : 100.39 Latency = 0.009957977914555272
[17:10:18] : [Subscribing to topic] [17:10:18] : [Node Peer4 called] [17:10:18] : [Response finished]  Subscribed function finished!  Number of request : 30525, timePassed: 5 mins, throughput : 101.75 Latency = 0.009825315389547262	[17:10:12] : [Node Peer5 called] [17:10:12] : [Response finished]  Subscribed function finished!  Number of request : 29458, timePassed: 5 mins, throughput : 98.19333333333333 Latency = 0.010180262151807881	[17:10:17] : [Node Peer6 receiving message] [17:10:17] : [Response finished]  Subscribed function finished!  Number of request : 29806, timePassed: 5 mins, throughput : 99.35333333333334 Latency = 0.010061929931914512

e. Pull

```
Pulling messages from pVnk
[17:26:29] : [Node Peer1 called re
[17:26:29] : [Response fro
Number of request : 45251,
timePassed: 5 mins,
throughput : 150.83666666666667
Latency = 0.006627525751852746
```

```
Pulling messages from rPep  
[17:26:28] : [Node Peer2 called  
[17:26:28] : [Response f  
  
Number of request : 45654,  
timePassed: 5 mins,  
throughput : 152.18  
Latency = 0.0065690308826252845
```

```
Pulling messages from UIBl  
[17:26:29] : [Node Peer3 called  
  
[17:26:29] : [Response f  
  
Number of request : 44828,  
timePassed: 5 mins,  
throughput : 149.42666666666668  
Latency = 0.0066899322023184936
```

```
Pulling messages from Puga
[17:26:27] : [Node Peer4 called
[17:26:27] : [Response -
Number of request : 45941,
timePassed: 5 mins,
throughput : 153.13666666666666
Latency = 0.006528199641488306
```

```
Pulling messages from JsnC
[17:26:24] : [Node Peer5] called
[17:26:24] : [Response

Number of request : 45780,
timePassed: 5 mins,
throughput : 152.6
Latency = 0.006551079325094761
```

```
Pulling messages from oMFn
[17:26:29] : [Node Peer6 called
[17:26:29] : [Response

Number of request : 45147,
timePassed: 5 mins,
throughput : 150.49
Latency = 0.006642278047099838
```

7 Peer Nodes

### a. Create Topic

```
Creating topic...  
[17:35:00] : f[Node Peer1 called  
[17:35:00] : [Response f  
Number of request : 35563,  
timePassed: 5 mins,  
throughput : 118.54333333333334  
Latency = 0.008434273852502335
```

```
Creating topic...
[17:35:00] : f[Node Peer2]
[17:35:00] : [Respo

Number of request : 34600
timePassed: 5 mins,
throughput : 115.33333333333333
Latency = 0.00866857981406
thread finished
issacode@DESKTOP-76HGQH2:~
```

```
Creating topic...
[17:35:00] : f[Node Peer3 called request
[17:35:00] : [Response from se

Number of request : 34992,
timePassed: 5 mins,
throughput : 116.64
Latency = 0.008571434447522244
```

```
Creating topic...
[17:35:00] : f[Node Peer4 called
[17:35:00] : [Response fr
Number of request : 34984,
timePassed: 5 mins,
throughput : 116.61333333333333
latency = 0.00057330010831533
```

```
Creating topic...
[17:35:00] : f[Node Peer5 called
[17:35:00] : [Response fr
Number of request : 35439,
timePassed: 5 mins,
throughput : 118.13
```

```
Creating topic...
[17:35:00] : f[Node Peer6 called
[17:35:00] : [Response f
Number of request : 35264,
timePassed: 5 mins,
throughput : 117.54666666666667
```

Creating topic...

```
[17:35:00] : f[Node Peer7 called request : {'requestNa  
[17:35:00] : [Response from server: Approved]
```

```
Number of request : 35684,  
timePassed: 5 mins,  
throughput : 118.94666666666667  
Latency = 0.008405373417439849
```

## b. Delete Topic

Deleting topic...

```
[17:46:14] : [Node Peer1 called request  
[17:46:14] : [Response from ser
```

```
Number of request : 43497,  
timePassed: 5 mins,  
throughput : 144.99  
Latency = 0.006895454702156534
```

Deleting topic...

```
[17:46:13] : [Node Peer2 called re  
[17:46:13] : [Response fro
```

```
Number of request : 43421,  
timePassed: 5 mins,  
throughput : 144.7366666666668  
Latency = 0.0069077787761469035
```

Deleting topic...

```
[17:46:17] : [Node Peer3 called r  
[17:46:17] : [Response fr
```

```
Number of request : 43123,  
timePassed: 5 mins,  
throughput : 143.7433333333334  
Latency = 0.006955553543600001
```

Deleting topic...

```
[17:46:15] : [Node Peer4 called r  
[17:46:15] : [Response fr
```

```
Number of request : 43471,  
timePassed: 5 mins,  
throughput : 144.9033333333334  
Latency = 0.006899274414840691
```

Deleting topic...

```
[17:46:13] : [Node Peer5 called r  
[17:46:13] : [Response fr
```

```
Number of request : 43949,  
timePassed: 5 mins,  
throughput : 146.4966666666667  
Latency = 0.0068242680202146395
```

Deleting topic...

```
[17:46:19] : [Node Peer6 called r  
[17:46:19] : [Response fr
```

```
Number of request : 44314,  
timePassed: 5 mins,  
throughput : 147.7133333333334  
Latency = 0.006767939029846315
```

Deleting topic...

```
[17:46:14] : [Node Peer7 called r  
[17:46:14] : [Response fr
```

```
Number of request : 44019,  
timePassed: 5 mins,  
throughput : 146.73  
Latency = 0.0068137190051041855
```

### c. Send

[17:57:59] : [Node Peer1 received message from 0xhO]  [17:57:59] : [Message gAi received]  Number of request : 35948, timePassed: 5 mins, throughput : 119.82666666666667 Latency = 0.008343110555958879	Sending message Nzhz to RuJm  [17:58:01] : [Node Peer2 called response f]  Number of request : 35869, timePassed: 5 mins, throughput : 119.56333333333333 Latency = 0.008361708034772855	Sending message eNbZ to oxHO  [17:58:05] : [Node Peer3 called response f]  Number of request : 36606, timePassed: 5 mins, throughput : 122.02 Latency = 0.008193647673252091
Sending message rzzo to dhUw  [17:58:00] : [Node Peer4 called response f]  Number of request : 36210, timePassed: 5 mins, throughput : 120.7 Latency = 0.008282497382829018	Sending message rHqY to yhSq  [17:58:04] : [Node Peer5 called response f]  Number of request : 36108, timePassed: 5 mins, throughput : 120.36 Latency = 0.008306157949478692	Sending message Apjt to zLft  [17:58:04] : [Node Peer6 called response f]  Number of request : 34722, timePassed: 5 mins, throughput : 115.74 Latency = 0.008638234438142788
Sending message EziU to AWOG  [17:58:01] : [Node Peer7 called response f]  Number of request : 36056, timePassed: 5 mins, throughput : 120.18666666666667 Latency = 0.008318808474455014		

### d. Subscribe

Subscribed function finished!  [18:10:51] : [Subscribing to topic 0xhO] [18:10:51] : [Node Peer2 called response f]  Subscribed function finished!  Number of request : 38385, timePassed: 5 mins, throughput : 127.95 Latency = 0.00781341193476055	[18:10:51] : [Subscribing to topic 0xhO] [18:10:51] : [Node Peer1 called response f]  Subscribed function finished!  Number of request : 37387, timePassed: 5 mins, throughput : 124.62333333333333 Latency = 0.008022446304222521	[18:10:46] : [Subscribing to topic 0xhO] [18:10:46] : [Node Peer3 called response f]  Subscribed function finished!  Number of request : 37132, timePassed: 5 mins, throughput : 123.77333333333333 Latency = 0.008077714863900132
--	--	--

[18:10:46] : [Subscribing to topic WcYw]	[18:10:44] : [Subscribing to top	[18:10:46] : [Subscribing to top
[18:10:46] : [Node Peer4 called request :	[18:10:44] : [Node Peer5 called	[18:10:46] : [Node Peer6 called
[18:10:46] : [Response from serve	[18:10:44] : [Response f	[18:10:46] : [Response f
Subscribed function finished!	Subscribed function finished!	Subscribed function finished!
Number of request : 36634, timePassed: 5 mins, throughput : 122.113333333333 Latency = 0.00818766995458264	Number of request : 37339, timePassed: 5 mins, throughput : 124.463333333333 Latency = 0.008032797292098768	Number of request : 36864, timePassed: 5 mins, throughput : 122.88 Latency = 0.008136200718581676
[18:10:47] : [Subscribing to t		
[18:10:47] : [Node Peer7 calle		
[18:10:47] : [Response		
Subscribed function finished!		
Number of request : 37716, timePassed: 5 mins, throughput : 125.72 Latency = 0.0079520564782732		

## e. Pull

Pulling messages from dJmj	Pulling messages from yICq	Pulling messages from kymv
[18:27:36] : [Node Peer1 called	[18:27:42] : [Node Peer2 called	[18:27:47] : [Node Peer3 called
[18:27:36] : [Response	[18:27:42] : [Response f	[18:27:47] : [Response f
Number of request : 42121, timePassed: 5 mins, throughput : 140.403333333334 Latency = 0.007120156808330905	Number of request : 42614, timePassed: 5 mins, throughput : 142.04666666666665 Latency = 0.007037589107690135	Number of request : 42126, timePassed: 5 mins, throughput : 140.42 Latency = 0.0071185236217594944
- fourth_session (sort: index) —	Pulling messages from RtKq	Pulling messages from DGAD
Number of request : 42359, timePassed: 5 mins, throughput : 141.19666666666666 Latency = 0.00708021725081596	[18:27:44] : [Node Peer5 called re	[18:27:45] : [Node Peer6 called
[18:27:45] : [Node Peer4 accepted	[18:27:44] : [Response fro	[18:27:45] : [Response
[18:27:33] : [Node Peer7 calle		
[18:27:33] : [Response		
Number of request : 42309, timePassed: 5 mins, throughput : 141.03 Latency = 0.007088209427578825	Number of request : 42608, timePassed: 5 mins, throughput : 142.02666666666667 Latency = 0.007039049212931692	Number of request : 42445, timePassed: 5 mins, throughput : 141.48333333333332 Latency = 0.007065960616382388

## 8 Peer Nodes

### a. Create Topic

Creating topic... [18:39:46] : f[Node Peer1 called [18:39:46] : [Response from Node Peer1]  Number of request : 33190, timePassed: 5 mins, throughput : 110.6333333333334 Latency = 0.00903655745674954	Creating topic... [18:39:46] : f[Node Peer2 called [18:39:46] : [Response from Node Peer2]  Number of request : 33867, timePassed: 5 mins, throughput : 112.89 Latency = 0.008856343952692245	Creating topic... [18:39:46] : f[Node Peer3 called [18:39:46] : [Response from Node Peer3]  Number of request : 33768, timePassed: 5 mins, throughput : 112.56 Latency = 0.008882824623649704
Creating topic... [18:39:46] : f[Node Peer4 called [18:39:46] : [Response from Node Peer4]  Number of request : 32954, timePassed: 5 mins, throughput : 109.84666666666666 Latency = 0.009101302182275766	Creating topic... [18:39:46] : f[Node Peer5 called [18:39:46] : [Response from Node Peer5]  Number of request : 32972, timePassed: 5 mins, throughput : 109.90666666666667 Latency = 0.009096817991679464	Creating topic... [18:39:46] : f[Node Peer6 called [18:39:46] : [Response from Node Peer6]  Number of request : 33555, timePassed: 5 mins, throughput : 111.85 Latency = 0.008938719501191255
Creating topic... [18:39:46] : f[Node Peer7 called [18:39:46] : [Response from Node Peer7]  Number of request : 33718, timePassed: 5 mins, throughput : 112.393333333333 Latency = 0.008895423087200326	Creating topic... [18:39:46] : f[Node Peer8 called [18:39:46] : [Response from Node Peer8]  Number of request : 34069, timePassed: 5 mins, throughput : 113.563333333333 Latency = 0.008803522566877898	

### b. Delete Topic

Deleting topic... [19:02:44] : [Node Peer1 called [19:02:44] : [Response from Node Peer1]  Number of request : 29743, timePassed: 5 mins, throughput : 99.143333333333 Latency = 0.010083831278833638	Deleting topic... [19:02:43] : [Node Peer2 called [19:02:43] : [Response from Node Peer2]  Number of request : 29757, timePassed: 5 mins, throughput : 99.19 Latency = 0.010080053300175227	Deleting topic... [19:02:50] : [Node Peer3 called [19:02:50] : [Response from Node Peer3]  Number of request : 31500, timePassed: 5 mins, throughput : 105.0 Latency = 0.009522359711783272
--	--	--

<p>Deleting topic...</p> <p>[19:02:48] : [Node Peer4 called</p> <p>[19:02:48] : [Response f</p> <p>Number of request : 30659, timePassed: 5 mins, throughput : 102.19666666666667 Latency = 0.009783583567051423</p>	<p>Deleting topic...</p> <p>[19:02:35] : [Node Peer5 called</p> <p>[19:02:35] : [Response</p> <p>Number of request : 29610, timePassed: 5 mins, throughput : 98.7 Latency = 0.010129834566113112</p>	<p>Deleting topic...</p> <p>[19:02:40] : [Node Peer6 called</p> <p>[19:02:40] : [Response</p> <p>Number of request : 28426, timePassed: 5 mins, throughput : 94.75333333333333 Latency = 0.010552081821554107</p>
<p>Deleting topic...</p> <p>[19:02:39] : [Node Peer7 called</p> <p>[19:02:39] : [Response</p> <p>Number of request : 29164, timePassed: 5 mins, throughput : 97.2133333333334 Latency = 0.010285146686704116</p>	<p>Deleting topic...</p> <p>[19:02:43] : [Node Peer8 called</p> <p>[19:02:43] : [Response</p> <p>Number of request : 29252, timePassed: 5 mins, throughput : 97.50666666666666 Latency = 0.010253733402235505</p>	

### c. Send

<p>Sending message LwkZ to RdEa</p> <p>[19:19:02] : [Node Peer1 called req</p> <p>[19:19:02] : [Response from</p> <p>Number of request : 32353, timePassed: 5 mins, throughput : 107.843333333333 Latency = 0.009270863942310283</p>	<p>Sending message PXLx to sntU</p> <p>[19:18:53] : [Node Peer2 called</p> <p>[19:18:53] : [Response</p> <p>Number of request : 30074, timePassed: 5 mins, throughput : 100.2466666666667 Latency = 0.009973492625229471</p>	<p>Sending message ElvR to UGqL</p> <p>[19:18:55] : [Node Peer3 called</p> <p>[19:18:55] : [Response</p> <p>Number of request : 30306, timePassed: 5 mins, throughput : 101.02 Latency = 0.009897488371764034</p>
<p>Sending message tgqZ to WSom</p> <p>[19:18:52] : [Node Peer4 called</p> <p>[19:18:52] : [Response f</p> <p>Number of request : 30347, timePassed: 5 mins, throughput : 101.15666666666667 Latency = 0.009883221014329981</p>	<p>Sending message CpmT to fGaT</p> <p>[19:18:57] : [Node Peer5 called r</p> <p>[19:18:57] : [Response fr</p> <p>Number of request : 31682, timePassed: 5 mins, throughput : 105.60666666666667 Latency = 0.009467481558434734</p>	<p>Sending message BZWu to kUyX</p> <p>[19:18:53] : [Node Peer6 called</p> <p>[19:18:53] : [Response fr</p> <p>Number of request : 30493, timePassed: 5 mins, throughput : 101.64333333333333 Latency = 0.00983694736319458</p>

Sending message DNiT to IXaC

```
[19:18:53] : [Node Peer7 called]
[19:18:53] : [Response from IXaC]
Number of request : 30193,
timePassed: 5 mins,
throughput : 100.64333333333333
Latency = 0.0099339444501558
```

Sending message ITfo to IlDC

```
[19:18:59] : [Node Peer8 called]
[19:18:59] : [Response from IlDC]
Number of request : 31292,
timePassed: 5 mins,
throughput : 104.30666666666667
Latency = 0.009584075553054839
```

#### d. Subscribe

```
[19:32:55] : [Node Peer1 called]
[19:32:55] : [Response from Peer1]
Subscribed function finished!
Number of request : 29164,
timePassed: 5 mins,
throughput : 97.2133333333334
Latency = 0.01028489979054764
```

```
[19:32:50] : [Subscribing to topic]
[19:32:50] : [Node Peer2 called]
[19:32:50] : [Response from Peer2]
Subscribed function finished!
Number of request : 27542,
timePassed: 5 mins,
throughput : 91.80666666666667
Latency = 0.010889314270324241
```

```
[19:32:47] : [Subscribing to topic]
[19:32:47] : [Node Peer3 called]
[19:32:47] : [Response from Peer3]
Subscribed function finished!
Number of request : 30517,
timePassed: 5 mins,
throughput : 101.72333333333333
Latency = 0.009827356974138017
```

```
[19:32:51] : [Subscribing to topic]
[19:32:51] : [Node Peer4 called]
[19:32:51] : [Response from Peer4]
Subscribed function finished!
Number of request : 28346,
timePassed: 5 mins,
throughput : 94.48666666666666
Latency = 0.010581700305431369
```

```
[19:32:48] : [Subscribing to topic]
[19:32:48] : [Node Peer5 called]
[19:32:48] : [Response from Peer5]
Subscribed function finished!
Number of request : 27856,
timePassed: 5 mins,
throughput : 92.8533333333334
Latency = 0.010767956725966307
```

```
[19:32:45] : [Subscribing to topic]
[19:32:45] : [Node Peer6 called]
[19:32:45] : [Response from Peer6]
Subscribed function finished!
Number of request : 28252,
timePassed: 5 mins,
throughput : 94.17333333333333
Latency = 0.010616905220368757
```

## e. Pull

Pulling messages from mKSm [19:52:53] : [Node Peer1 called request]  [19:52:53] : [Response from server]  Number of request : 36177, timePassed: 5 mins, throughput : 120.59 Latency = 0.008290633859499165	Pulling messages from xZPK [19:52:53] : [Node Peer2 called request]  [19:52:53] : [Response from server]  Number of request : 37591, timePassed: 5 mins, throughput : 125.3033333333333 Latency = 0.00797855873378859
Pulling messages from NMBN [19:52:59] : [Node Peer3 called request]  [19:52:59] : [Response from server]  Number of request : 35921, timePassed: 5 mins, throughput : 119.73666666666666 Latency = 0.008350784796316572	Pulling messages from HHcd [19:53:01] : [Node Peer4 called request]  [19:53:01] : [Response from server]  Number of request : 36117, timePassed: 5 mins, throughput : 120.39 Latency = 0.008304174686318553
Pulling messages from UIcM [19:53:02] : [Node Peer5 called request]  [19:53:02] : [Response from server]  Number of request : 36318, timePassed: 5 mins, throughput : 121.06 Latency = 0.008258035978731837	Pulling messages from GlKB [19:53:00] : [Node Peer6 called request]  [19:53:00] : [Response from server]  Number of request : 36406, timePassed: 5 mins, throughput : 121.35333333333334 Latency = 0.008238053142399027  [19:53:00] : [Node Peer6 accepted message]
Pulling messages from VYPt [19:52:52] : [Node Peer7 called request]  [19:52:52] : [Response from server]  Number of request : 38860, timePassed: 5 mins, throughput : 129.53333333333333 Latency = 0.007717771083499989	Pulling messages from RDKD [19:53:00] : [Node Peer8 called request]  [19:53:00] : [Response from server]  Number of request : 37374, timePassed: 5 mins, throughput : 124.58 Latency = 0.008024511013468879