lab3 fabric搭建peer并加入通道

姓名： 陈奕衡

学号： PB20000024

实验目的以及要求

- 了解fabric上的基本配置
- 在fabric网络中添加peer，并加入网络
- 了解fabric上的基本证书和网络

实验平台

- Windows 10 professional
- ubuntu虚拟机（ssh远程连接，配备有注册好的ca服务器）

实验步骤

在虚拟机上已经有注册好的ca服务器，端口号为7054

注册身份并获取证书

首先在 TLS 服务上注册 Peer 身份，该部分所用指令如下：

```
export FABRIC_CA_CLIENT_TLS_CERTFILES=/home/ubuntu/fabric/tls-ca/crypto/tls-ca-
cert.pem

export FABRIC_CA_CLIENT_HOME=/home/ubuntu/fabric/tls-ca/admin

// 注册身份
fabric-ca-client register --id.name PB20000024 --id.secret cyh --id.type peer -u
https://172.16.4.35:7054

//获得对应的tls_msp
//存储tls_msp的位置
export FABRIC_CA_CLIENT_MSPDIR=/home/ubuntu/blockchain-lab
fabric-ca-client enroll -u https://PB20000024:cyh@172.16.4.35:7052 --
enrollment.profile tls --csr.hosts PB20000024

//注册成功后，可以把keystore下的文件存储为证书．当然，你也可以手动将其重新命名．
mv tls-msp/keystore/*_sk tls-msp/keystore/key.pem
mv msp/keystore/*_sk msp/keystore/key.pem
```

实现效果如下：

```
ubuntu@cyh:~$ fabric-ca-client register --id.name PB20000024 --id.secret cyh --id.type peer -u https://172.16.4.35:7052
2022/06/24 18:18:47 [INFO] Configuration file location: /home/ubuntu/fabric/tls-ca/admin/fabric-ca-client-config.yaml
2022/06/24 18:18:47 [INFO] TLS Enabled
2022/06/24 18:18:47 [INFO] TLS Enabled
Password: cyh
```

```
ubuntu@cyh:~$ fabric-ca-client enroll -u https://PB20000024:cyh@172.16.4.35:7052 --enrollment.profile tls --csr.hosts PB20000024
2022/06/24 18:23:02 [INFO] TLS Enabled
2022/06/24 18:23:02 [INFO] generating key: &{A:ecdsa S:256}
2022/06/24 18:23:02 [INFO] encoded CSR
2022/06/24 18:23:02 [INFO] Stored client certificate at /home/ubuntu/blockchain-lab/signcerts/cert.pem
2022/06/24 18:23:02 [INFO] Stored TLS root CA certificate at /home/ubuntu/blockchain-lab/tlscacerts/tls-172-16-4-35-7052.pem
2022/06/24 18:23:02 [INFO] Stored Issuer public key at /home/ubuntu/blockchain-lab/IssuerPublicKey
2022/06/24 18:23:02 [INFO] Stored Issuer revocation public key at /home/ubuntu/blockchain-lab/IssuerRevocationPublicKey
```

```
.
├── IssuerPublicKey
├── IssuerRevocationPublicKey
├── cacerts
├── keystore
│   └── 0a534dd460f3d51c8d08416699ba775a971a1677295b28728b9e05f6910aa541_sk
├── signcerts
│   └── cert.pem
├── tlscacerts
│   └── tls-172-16-4-35-7052.pem
└── user

5 directories, 5 files
```

之后注册组织的peer身份：

```
    export FABRIC_CA_CLIENT_TLS_CERTFILES=/home/ubuntu/fabric/org1/ca/crypto/ca-cert.pem
    export FABRIC_CA_CLIENT_HOME=/home/ubuntu/fabric/org1/ca/admin

    // 注册身份
    fabric-ca-client register --id.name $PB20000024 --id.secret $cyh --id.type peer -u https://172.16.4.35:7054 --csr.hosts ${HOSTNAME}

    //获得对应的tls_msp
    //存储tls_msp的位置
    export FABRIC_CA_CLIENT_MSPDIR=/home/ubuntu/blockchain-lab
    fabric-ca-client enroll -u https://PB20000024:cyh@172.16.4.35:7054
```

实现效果如下：

```
ubuntu@cyh:~$ fabric-ca-client register --id.name PB20000024 --id.secret cyh --id.type peer -u https://172.16.4.35:7054
2022/06/24 17:54:56 [INFO] Configuration file location: /home/ubuntu/fabric/org1/ca/admin/fabric-ca-client-config.yaml
2022/06/24 17:54:56 [INFO] TLS Enabled
2022/06/24 17:54:56 [INFO] TLS Enabled
Error: Response from server: Error Code: 71 - Authorization failure
```

```
ubuntu@cyh:~$ fabric-ca-client enroll -u https://PB20000024:cyh@172.16.4.35:7054 --enrollment.profile tls --csr.hosts PB20000024
2022/06/24 17:56:05 [INFO] TLS Enabled
2022/06/24 17:56:05 [INFO] generating key: &{A:ecdsa S:256}
2022/06/24 17:56:05 [INFO] encoded CSR
2022/06/24 17:56:06 [INFO] Stored client certificate at /home/ubuntu/blockchain-lab/signcerts/cert.pem
2022/06/24 17:56:06 [INFO] Stored TLS root CA certificate at /home/ubuntu/blockchain-lab/tlscacerts/tls-172-16-4-35-7054.pem
2022/06/24 17:56:06 [INFO] Stored Issuer public key at /home/ubuntu/blockchain-lab/IssuerPublicKey
2022/06/24 17:56:06 [INFO] Stored Issuer revocation public key at /home/ubuntu/blockchain-lab/IssuerRevocationPublicKey
```

```
ubuntu@cyh:~/blockchain-lab$ tree
.
├── msp
│   ├── IssuerPublicKey
│   ├── IssuerRevocationPublicKey
│   ├── cacerts
│   │   └── 172-16-4-35-7054.pem
│   ├── keystore
│   │   └── 88cca621708a46a1d0b4f2a9ac98201deb5b554b9ecbcb1e43114986e4829673_sk
│   ├── signcerts
│   │   └── cert.pem
│   └── user
└── tls-msp
    ├── IssuerPublicKey
    ├── IssuerRevocationPublicKey
    ├── cacerts
    ├── keystore
    │   └── 2678b33e2ee2dbf918b1440842623b3f75f9bfe9cded38b58cea4ac78ad47d3a_sk
    ├── signcerts
    │   └── cert.pem
    ├── tlscacerts
    │   └── tls-172-16-4-35-7052.pem
    └── user
```

使用如下指令打印证书：

```
查看pem证书
openssl x509 -in ./msp/signcerts/cert.pem -text
```

```
ubuntu@cyh:~$ openssl x509 -in ./blockchain-lab/signcerts/cert.pem -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            3f:ed:ee:b6:4b:dd:bc:e7:06:a5:27:8d:3e:74:30:ec:6d:d3:0f:17
        Signature Algorithm: ecdsa-with-SHA256
        Issuer: C = US, ST = North Carolina, O = Hyperledger, OU = Fabric, CN = ca-org1
        Validity
            Not Before: May 18 14:25:00 2022 GMT
            Not After : Jun 24 17:56:00 2023 GMT
        Subject: C = US, ST = North Carolina, O = Hyperledger, OU = peer, CN = PB20000024
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (256 bit)
                pub:
                    04:50:25:b8:9d:72:f0:44:f9:be:fd:19:67:72:10:
                    67:9a:fd:55:1f:d9:96:24:64:70:16:cd:be:8f:b3:
                    aa:28:64:81:6c:16:a6:90:fe:9d:a8:3c:ce:d2:08:
                    97:bd:bd:e6:8b:26:50:e6:12:e1:cb:33:ea:a7:29:
                    a1:22:6d:c9:02
                ASN1 OID: prime256v1
                NIST CURVE: P-256
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment, Key Agreement
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Subject Key Identifier:
                61:25:68:81:B9:ED:31:6F:6A:50:39:3E:3F:BA:70:F0:2D:E3:17:6D
            X509v3 Authority Key Identifier:
                BB:B7:B6:CD:79:52:A5:17:DC:BB:6B:6A:3E:DF:E1:3E:6F:D8:48:2C
            X509v3 Subject Alternative Name:
                DNS:PB20000024
            1.2.3.4.5.6.7.8.1:
                {"attrs":{"hf.Affiliation":"","hf.EnrollmentID":"PB20000024","hf.Type":"peer"}}
    Signature Algorithm: ecdsa-with-SHA256
    Signature Value:
        30:45:02:21:00:dc:68:f4:3f:d0:f6:a1:a0:14:48:7e:28:bc:
        13:a8:58:77:6a:99:d2:90:d1:93:6c:b0:c6:13:a1:68:f6:79:
        b8:02:20:5a:a9:b5:73:54:04:e0:67:ec:08:e0:2b:0d:ec:d7:
        ec:03:a9:63:06:2b:e4:f5:c4:97:a5:71:21:94:f1:fd:0b
-----BEGIN CERTIFICATE-----
MIICtjCCAlygAwIBAgIUP+3utkvdvOcGpSeNPnQw7G3TDxcwCgYIKoZIzj0EAwIw
XzELMAkGA1UEBhMCVVMxFzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMRQwEgYDVQQK
EwtIeXBlcmxlZGdlcjEPMA0GA1UECxMGRmFicmljMRAwDgYDVQQDEwdjYS1vcmcx
MB4XDTIyMDUxODE0MjUwMFoXDTIzMDYyNDE3NTYwMFowYDELMAkGA1UEBhMCVVMx
FzAVBgNVBAgTDk5vcnRoIENhcm9saW5hMRQwEgYDVQQKEwtIeXBlcmxlZGdlcjEN
MAsGA1UECxMEcGVlcjETMBEGA1UEAxMKUEIyMDAwMDAyNDBZMBMGByqGSM49AgEG
CCqGSM49AwEHA0IABFAluJ1y8ET5vv0ZZ3IQZ5r9VR/ZliRkcBbNvo+zqihkgWwW
ppD+nag8ztIIl7295osmUOYS4csz6qcpoSJtyQKjgfQwgfEwDgYDVR0PAQH/BAQD
AgOoMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggrBgEFBQcDAjAMBgNVHRMBAf8EAjAA
MB0GA1UdDgQWBBRhJWiBue0xb2pQOT4/unDwLeMXbTAfBgNVHSMEGDAWgBS7t7bN
eVKlF9y7a2o+3+E+b9hILDAVBgNVHREEDjAMggpQQjIwMDAwMDI0MFsGCCoDBAUG
BwgBBE97ImF0dHJzIjp7ImhmLkFmZmlsaWF0aW9uIjoiIiwiaGYuRW5yb2xsbWVu
dElEIjoiUEIyMDAwMDAyNCIsImhmLlR5cGUiOiJwZWVyIn19MAoGCCqGSM49BAMC
A0gAMEUCIQDcaPQ/0PahoBRIfii8E6hYd2qZ0pDRk2ywxhOhaPZ5uAIgWqm1c1QE
4GfsCOArDezX7AOpYwYr5PXEl6VxIZTx/Qs=
-----END CERTIFICATE-----
```

启动Peer节点

主要工作为

1. 构造msp文件夹
2. 设置环境变量（如果有需要）
3. 启动peer和CLI

msp文件夹配置

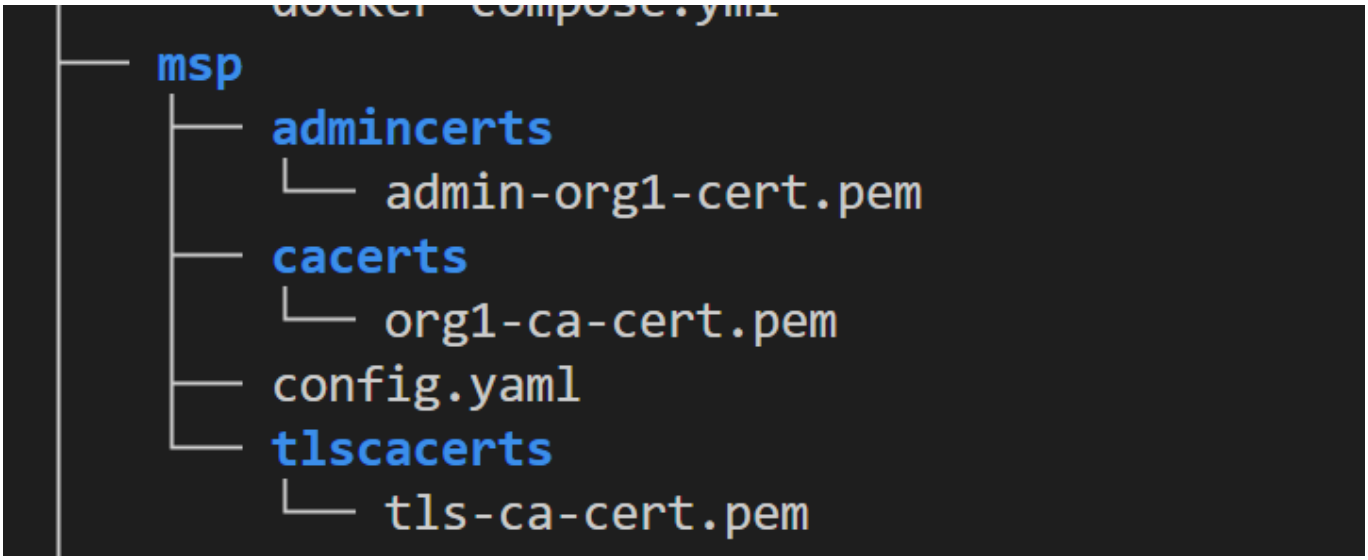
msp文件夹对应位置为

```
org/msp
org/admin/msp
org/peer/msp
```

其中org/msp对应的证书为：

```
cp ${HOME}/fabric/org1/admin/msp/signcerts/cert.pem org1/msp/admincerts/admin-
org1-cert.pem
cp ${HOME}/fabric/org1/ca/crypto/ca-cert.pem org1/msp/cacerts/org1-ca-cert.pem
cp ${HOME}/fabric/org1/peer2/assets/tls-ca/tls-ca-cert.pem
org1/msp/tlscacerts/tls-ca-cert.pem
```

构建完成后的org1文件组织：



对应需要创建创建每个组织的config.yaml文件

```
NodeOUs:
  Enable: true
  ClientOUIdentifier:
    Certificate: cacerts/org1-ca-cert.pem
```

```
    OrganizationalUnitIdentifier: client
  PeerOUIdentifier:
    Certificate: cacerts/org1-ca-cert.pem
    OrganizationalUnitIdentifier: peer
  AdminOUIdentifier:
    Certificate: cacerts/org1-ca-cert.pem
    OrganizationalUnitIdentifier: admin
  OrdererOUIdentifier:
    Certificate: cacerts/org1-ca-cert.pem
    OrganizationalUnitIdentifier: orderer
```

之后新建一个docker-compose文件，如下：

```
---
version: "3"
networks:
  fabric-ca:
    external:
      name: fabric-ca
services:
  PB20000024:
    container_name: PB20000024
    environment:
      - "GOPROXY=https://goproxy.cn,direct"
      - GO111MODULE=on
      - CORE_PEER_ID=org-peer
      - "CORE_PEER_ADDRESS=PB20000024:7051"
      - CORE_PEER_LOCALMSPID=org1MSP
      - CORE_PEER_MSPCONFIGPATH=/home/ubuntu/fabric/org1/peer2/msp
      - "CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock"
      - CORE_VM_DOCKER_ATTACHSTDOUT=true
      - CORE_VM_DOCKER_HOSTCONFIG_NETWORKMODE=fabric-ca
      - CORE_PEER_TLS_ENABLED=true
      - CORE_PEER_TLS_CERT_FILE=/home/ubuntu/fabric/org1/peer2/tls-
msp/signcerts/cert.pem
      - CORE_PEER_TLS_KEY_FILE=/home/ubuntu/fabric/org1/peer2/tls-
msp/keystore/key.pem
      - CORE_PEER_TLS_ROOTCERT_FILE=/home/ubuntu/fabric/org1/peer2/tls-
msp/tlscacerts/tls-172-16-4-35-7052.pem
      - CORE_PEER_GOSSIP_USELEADERELECTION=false
      - CORE_PEER_GOSSIP_ORGLEADER=true
      - "CORE_PEER_GOSSIP_EXTERNALENDPOINT=PB20000024:7051"
      - CORE_PEER_GOSSIP_SKIPHANDSHAKE=true
    image: "hyperledger/fabric-peer:amd64-2.2.0"
    networks:
      - fabric-ca
    volumes:
      - "/var/run:/host/var/run"
      - "/home/ubuntu/fabric/org1/peer2:/home/ubuntu/fabric/org1/peer2"
    working_dir: /opt/gopath/src/github.com/hyperledger/fabric/org1/peer2
```

开启peer服务之后可以看到如下语句，证明启动成功：

```
2022-06-26 08:01:38.075 UTC [nodeCmd] serve -> INFO 017 Starting peer with ID=[org-peer], network ID=[dev], address=[PB20000
024:7051]
2022-06-26 08:01:38.075 UTC [nodeCmd] serve -> INFO 018 Started peer with ID=[org-peer], network ID=[dev], address=[PB200000
24:7051]
```

之后建立类似docker-compose文件用于启动cli容器，如下：

```
version: "3"
networks:
  fabric-ca:
    external:
      name: fabric-ca
services:
  cli-org1:
    container_name: cli-org1
    image: hyperledger/fabric-tools:amd64-2.2.0
    tty: true
    stdin_open: true
    environment:
      - GOPROXY=https://goproxy.cn,direct
      - GO111MODULE=on
      - GOPATH=/opt/gopath
      - CORE_VM_ENDPOINT=unix:///host/var/run/docker.sock
      - FABRIC_LOGGING_SPEC=DEBUG
      - CORE_PEER_ID=cli-org1
      # peer服务对应的位置
      - CORE_PEER_ADDRESS=peer2-org1:7051
      - CORE_PEER_LOCALMSPID=org1MSP
      - CORE_PEER_TLS_ENABLED=true
      # tls根证书
      - CORE_PEER_TLS_ROOTCERT_FILE=/home/ubuntu/fabric/org1/peer2/tls-
msp/tlscacerts/tls-172-16-4-35-7052.pem
      # tls证书
      - CORE_PEER_TLS_CERT_FILE=/home/ubuntu/fabric/org1/peer2/tls-
msp/signcerts/cert.pem
      # 存储的密钥
      - CORE_PEER_TLS_KEY_FILE=/home/ubuntu/fabric/org1/peer2/tls-
msp/keystore/key.pem
      - CORE_PEER_MSPCONFIGPATH=/home/ubuntu/fabric/org1/peer2/msp
       working_dir: /opt/gopath/src/github.com/hyperledger/fabric/org1
    command: /bin/bash
    volumes:
      - /home/ubuntu/fabric/org1/peer2:/home/ubuntu/fabric/org1/peer2
      -
/home/ubuntu/fabric/org1/peer2/assets/chaincode:/opt/gopath/src/github.com/hyperle
dger/fabric-samples/chaincode
      - /home/ubuntu/fabric/org1/admin:/home/ubuntu/fabric/org1/admin
    networks:
      - fabric-ca
```

建立完成之后便可进入通道：

```
2022-06-26 09:39:45.689 UTC [msp] setupSigningIdentity -> DEBU 017 Signing identity expires at 2023-06-26 06:51:00 +0000 UTC
2022-06-26 09:39:45.692 UTC [msp] GetDefaultSigningIdentity -> DEBU 018 Obtaining default signing identity
2022-06-26 09:39:45.700 UTC [grpc] Infof -> DEBU 019 parsed scheme: ""
2022-06-26 09:39:45.702 UTC [grpc] Infof -> DEBU 01a scheme "" not registered, fallback to default scheme
2022-06-26 09:39:45.705 UTC [grpc] Infof -> DEBU 01b ccResolverWrapper: sending update to cc: {[{peer2-org1:7051  <nil> 0 <n
il>}] <nil> <nil>}
2022-06-26 09:39:45.708 UTC [grpc] Infof -> DEBU 01c ClientConn switching balancer to "pick_first"
2022-06-26 09:39:45.709 UTC [grpc] Infof -> DEBU 01d Channel switches to new LB policy "pick_first"
2022-06-26 09:39:45.711 UTC [grpc] Infof -> DEBU 01e Subchannel Connectivity change to CONNECTING
2022-06-26 09:39:45.714 UTC [grpc] Infof -> DEBU 01f Subchannel picks a new address "peer2-org1:7051" to connect
2022-06-26 09:39:45.716 UTC [grpc] UpdateSubConnState -> DEBU 020 pickfirstBalancer: HandleSubConnStateChange: 0xc000286bc0,
 {CONNECTING <nil>}
2022-06-26 09:39:45.720 UTC [grpc] Infof -> DEBU 021 Channel Connectivity change to CONNECTING
2022-06-26 09:39:45.763 UTC [grpc] Infof -> DEBU 022 Subchannel Connectivity change to READY
2022-06-26 09:39:45.766 UTC [grpc] UpdateSubConnState -> DEBU 023 pickfirstBalancer: HandleSubConnStateChange: 0xc000286bc0,
 {READY <nil>}
2022-06-26 09:39:45.768 UTC [grpc] Infof -> DEBU 024 Channel Connectivity change to READY
2022-06-26 09:39:45.769 UTC [channelCmd] InitCmdFactory -> INFO 025 Endorser and orderer connections initialized
2022-06-26 09:39:45.773 UTC [msp.identity] Sign -> DEBU 026 Sign: plaintext: 0ACA080A5C08031A0C08E1D2E0950610...631A0D0A0B47
65744368616E6E656C73
2022-06-26 09:39:45.774 UTC [msp.identity] Sign -> DEBU 027 Sign: digest: FD4E9EBA3115947993998BEA4E26637D4E8AB26604D333FE90
FE7A877266D831
Channels peers has joined:
mychannel
```

如下可知通道建立成功：

```
2022-06-26 16:16:52.040 UTC [msp.identity] Sign -> DEBU 027 Sign: digest: 67452CA8D9B31195E4BB0FD59024A1244DAB4382C8E1A6E6400660FE873DDE0C
Channels peers has joined:
mychannel
```

添加通道

利用如下指令加入通道：

```
//设置身份
export CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/org1/admin/msp
//添加通道对应的区块
peer channel join -b mychannel.block
//查看加入的通道
peer channel list
```

成功加入通道之后，可以看到日志中区块之间执行gossip协议

```
2022-06-26 16:17:25.955 UTC [gossip.discovery] expireDeadMembers -> WARN 8c2 Exiting
2022-06-26 16:17:25.996 UTC [gossip.comm] sendToEndpoint -> WARN 8c3 Failed obtaining connection for admin2:7051, PKIid:1ee842e5ba778a0de6adde
2856a0d1768fd9bfe718b69bc3d3a4afa3e1ce8f94 reason: context deadline exceeded
2022-06-26 16:17:26.000 UTC [gossip.comm] sendToEndpoint -> WARN 8c4 Failed obtaining connection for PB19071405:7051, PKIid:4cd1e8c89441db8dc1
1db3d47126f908ebba39cb84962610da73ba9edcebada3 reason: context deadline exceeded
2022-06-26 16:17:26.884 UTC [gossip.comm] sendToEndpoint -> WARN 8c5 Failed obtaining connection for PB19051069:7051, PKIid:5f3c843f50758e8256
cd59e2e2f4edbf5e5be1ab998192a3ac731453064eac86 reason: context deadline exceeded
2022-06-26 16:17:26.886 UTC [gossip.discovery] expireDeadMembers -> WARN 8c6 Entering [5f3c843f50758e8256cd59e2e2f4edbf5e5be1ab998192a3ac73145
3064eac86]
2022-06-26 16:17:26.886 UTC [gossip.discovery] expireDeadMembers -> WARN 8c7 Closing connection to Endpoint: PB19051069:7052, InternalEndpoint
: PB19051069:7051, PKI-ID: 5f3c843f50758e8256cd59e2e2f4edbf5e5be1ab998192a3ac731453064eac86, Metadata:
2022-06-26 16:17:26.886 UTC [gossip.discovery] expireDeadMembers -> WARN 8c8 Exiting
2022-06-26 16:17:26.967 UTC [gossip.comm] sendToEndpoint -> WARN 8c9 Failed obtaining connection for PB19061392:7051, PKIid:6079b005fab19130d0
2620edf10ffc2629e33c6aade11f03594eac4e7c962185 reason: context deadline exceeded
2022-06-26 16:17:26.972 UTC [gossip.comm] sendToEndpoint -> WARN 8ca Failed obtaining connection for PB19111700_3:7051, PKIid:17bb5b4b36a4f816
fbd2e9af755eff75524ed90c04fcfeec4705d1ffefe5bd6d reason: context deadline exceeded
2022-06-26 16:17:26.977 UTC [gossip.comm] sendToEndpoint -> WARN 8cb Failed obtaining connection for kekePB19111661:7051, PKIid:ac0a7ed5b1bfa5
2f5106bf0947cc3c0c003a8cda9b74a900d98aa87d8ec74309 reason: context deadline exceeded
2022-06-26 16:17:27.143 UTC [gossip.comm] sendToEndpoint -> WARN 8cc Failed obtaining connection for PB19020497DJX-I:7051, PKIid:b49d58bf107d5
385d28aba73c23843cf4ce09e83f9c0f88973b4e38ef2fc1ee1 reason: context deadline exceeded
```