

ネットワークプログラミング 最終レポート

222C1021 今村優希

2024年8月2日

演習 2

演習 3

3.1

ソースコード 1 は、ソケット通信を利用し、学生番号を入力し、受診した文字列をテキストファイルに格納するプログラムである。図 1 が、このプログラムを実行した際の出力である。

IP アドレス 131.206.37.50 に接続できた。222C1021.txt の内容を受診し、222C1021-copy.txt に書き込むことができた。

ソースコード 1 ftp.c

```
/*222C1021 今村優希*/
/*ftp.c*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <fcntl.h>
#include <unistd.h>

int main(){
    int sockfd, fd;
    struct sockaddr_in address;
    char buf[80] = "\0";
    char end[10] = "exit_send";
    // INETドメイン、ストリームソケットを利用
    sockfd = socket(AF_INET, SOCK_STREAM, 0);

    //サーバー情報を設定
    address.sin_family = AF_INET;
    address.sin_addr.s_addr = inet_addr("131.206.37.50");
    address.sin_port = htons(50000);

    //クライアントからの接続要求とサーバ情報を確認
    //川原研のサーバーに接続
    int res = connect(sockfd, (struct sockaddr *)&address, sizeof(address));

    if(res == -1){
        perror("error\n");
        exit(1);
    }
}
```

```

}
printf("\n * server IP: %s, port: %d\n", inet_ntoa(address.sin_addr),
      ntohs(address.sin_port));

//strcpy(buf, "client will connect to server");
//write(sockfd, buf, strlen(buf));

//サーバーに学生番号を送信
printf("Your ID:");
scanf("%s", buf);
write(sockfd, buf, strlen(buf));

printf("read start\n");

/*222C1021-copy.txtのオープン*/
fd = open("222C1021-copy.txt", O_WRONLY);
if(fd == -1){
    fprintf(stderr, "can't open the file\n");
    exit(1);
}

/*サーバからのデータ受信*/
//入力がexit_sendになるまで読み込み、.txtに書き込む
while(1){
    memset(buf, '\0', sizeof(buf)); // buf[]読み込み前に初期化
    int point = read(sockfd, buf, sizeof(buf));
    write(fd, buf, strlen(buf));
    if(strcmp(buf, end) == 0){
        printf("read finish\n");
        break;
    }
}

close(fd); //222C1021-copy.txtのクローズ
close(sockfd); //ソケットの除去
}

```

```
● imamura@imamura-VirtualBox:~/ドキュメント/network prog/7/ftp$ gcc -o ftp ftp.c
● imamura@imamura-VirtualBox:~/ドキュメント/network prog/7/ftp$ ./ftp

* server IP: 131.206.37.50, port: 50000
Your ID:222C1021
read start
read finish
```

図 1 ftp.c を実行した結果

3-2

取得したテキストファイルは以下の内容であった。

課題：ルータにおける IP データグラム（パケット）転送

1. インターネットはパケット単位の蓄積交換方式（パケット交換方式）を採用するが、その利点と問題点を説明せよ。

2. 異種ネットワークを相互接続し、1. の主体となる「ルータ」において、到着パケットの転送先を決定する経路制御方式を一般的に何と呼ぶか答え、IPv4 アドレスを仮定した場合、どのように実現するか調査、説明せよ。

3. 代表的な経路情報交換プロトコルに RIP(Routing Information Protocol) と OSPF(Open Shortest Path First) がある。両者の違いを説明せよ。

課題 1

まず、蓄積交換方式とは、参考文献 [1] によると「インターネットではルータにおいてパケットをいったん蓄積して (store)、次のルータに転送する (forward) を意味している」とある。

したがって、蓄積交換方式のメリットとして、複数人で利用する際にインターネット資源を共有することができるという点が挙げられる。送信したいデータをパケットに分割することで、長いデータを送信したい等で資源を専有されることがなく、公平に使用することが可能である。

一方、デメリットとしてパケット到着の遅延と廃棄が発生する可能性があるという点が挙げられる。通信回線の利用率が高いとき、複数のパケットがルータに到着する可能性があり、バッファで待機することで遅延が発生する。さらに、この遅延が行き過ぎると、バッファにパケットが入らなくなり、パケットの廃棄が発生する。そうするとパケットの再送が必要で効率が落ちる可能性がある。

課題 2

ルータにおける到着パケットの転送先を決定する経路制御方式を経路制御またはルーティングという。さらに、ホップバイホップ制御を使っている。その制御の中でも 2 種類の制御方法があり、静的経路制御と動的経路制御が存在する。静的経路制御は、ルータの管理者が経路表を自ら作成しているのに対して、動的経路制御は経路表をルータ自身で行っている。よって、現在では後者の動的経路制御で行っているのが一般的である。

IPv4 アドレスと仮定し動的経路制御を用いて転送を行うと考える。まず、ルータは経路表を作成し、各宛先ネットワークの経路情報を保存している。下記で詳しく解答する動的経路制御プロトコルを利用してルータ同士で経路情報を交換し、更新を行っている。そして、ルータがパケットを受診したときに宛先 IP アドレスを確認する。その IP ア

ドレスと経路表を参照して次の転送先を決定し、転送を行っている。

課題 3

RIP は、参考文献 [2] によると、宛先アドレスとメトリックを含んだ経路情報を隣接するルータの間で交換し自己の経路表を作成するとされている。メトリックとは、経由するルータの数、ホップ数が使用されている。経路表は Bellman-Ford アルゴリズムを用いて最短経路で宛先アドレスに到達できるようにしている。この方法から、RIP は距離ベクトル型経路制御プロトコルとして使用されている。

OSPF は参考文献 [2] によると、近隣ルータの認識、リンク状態情報と LSDB の作成および管理、LSDB をもとにした最短経路作成、代表ルータ、階層化、EGP といった多くの機能を備えたプロトコルである。以下でそれぞれについて詳細に説明する。

3-3

作成した server は

備考

https://github.com/tanusai646/network_program/tree/main/7

参考文献

- [1] 尾家祐二 後藤滋樹 小西和憲 西尾章治郎, 1 インターネット入門, 株式会社岩波書店, 2002
- [2] 堀良影 池永全志 門林雄基 後藤滋樹, 2 ネットワーク相互接続, 株式会社岩波書店, 2001