



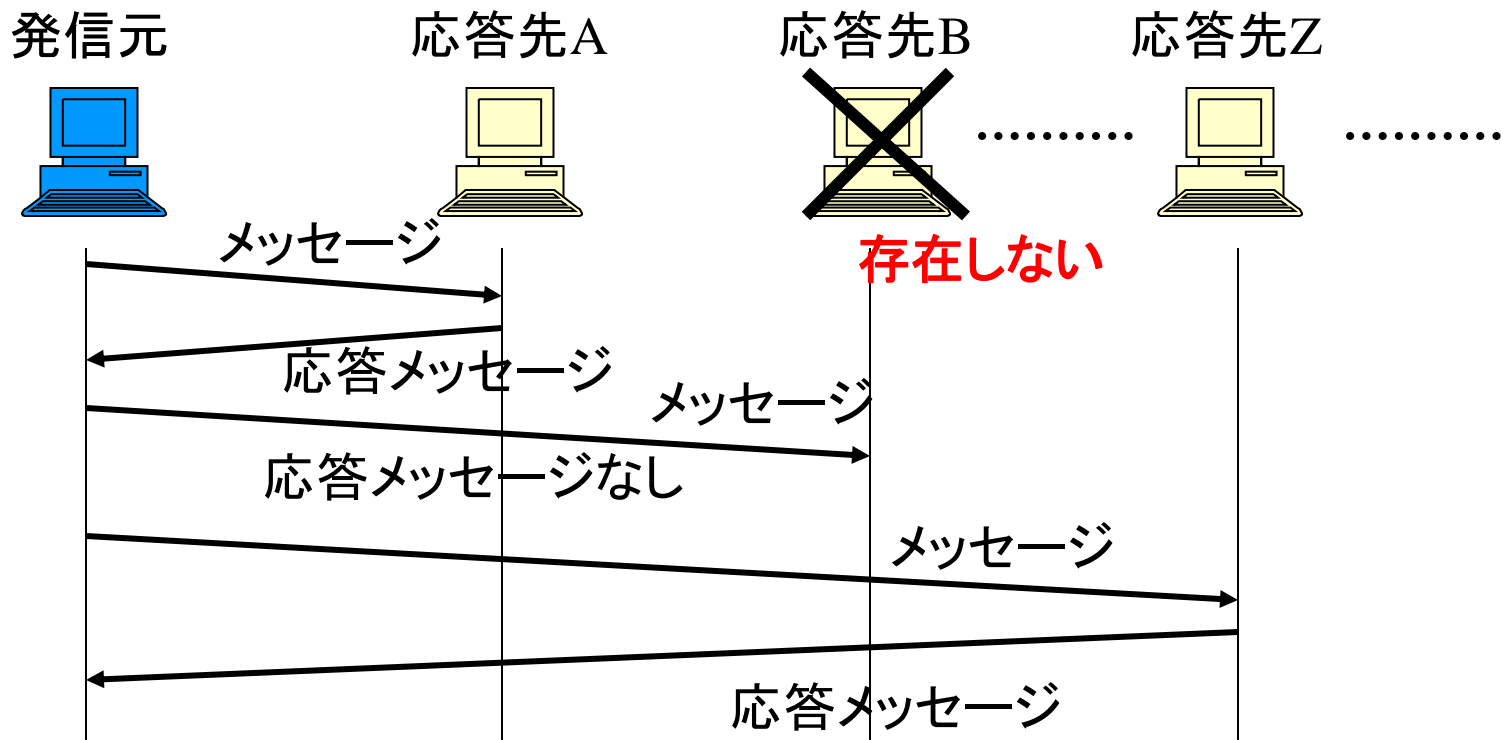
ネットワークスキャン実習

2002.9.20

稲田

ネットワークスキャンとは

- メッセージを送った際の応答を元に、ネットワーク上でのコンピュータ存在や提供されているサービスを探査すること



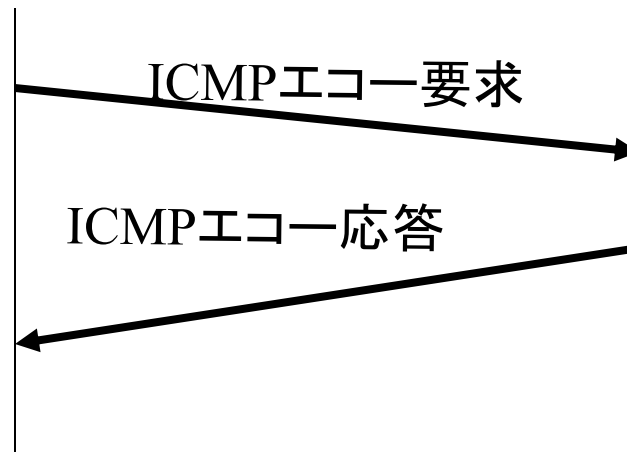
ping

- ICMPエコー要求とICMPエコー応答を利用
- エコー応答が返ってくれば、そのコンピュータは存在

クライアント



サーバ



pingの実行例

- hatsuneに対してpingを実行
 - ping ホスト名 (IPアドレス)
 - ICMPエコー応答パケットを受信 (パケットロス率 0%)
 - hatsuneというホストはネットワーク上に存在

```
inata@maaya[~]%ping -c 4 hatsune
PING hatsune.db.is.kyushu-u.ac.jp (133.5.18.167): 56 data bytes
64 bytes from 133.5.18.167: icmp_seq=0 ttl=254 time=0.766 ms
64 bytes from 133.5.18.167: icmp_seq=1 ttl=254 time=0.697 ms
64 bytes from 133.5.18.167: icmp_seq=2 ttl=254 time=0.734 ms
64 bytes from 133.5.18.167: icmp_seq=3 ttl=254 time=0.715 ms

--- hatsune.db.is.kyushu-u.ac.jp ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.697/0.728/0.766/0.026 ms
```

```
kterm
inata@maaya[~]%ping -c 4 133.5.18.168
PING 133.5.18.168 (133.5.18.168): 56 data bytes
64 bytes from 133.5.18.168: icmp_seq=0 ttl=254 time=1.086 ms
64 bytes from 133.5.18.168: icmp_seq=1 ttl=254 time=1.000 ms
64 bytes from 133.5.18.168: icmp_seq=2 ttl=254 time=1.103 ms
64 bytes from 133.5.18.168: icmp_seq=3 ttl=254 time=0.997 ms

--- 133.5.18.168 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.997/1.046/1.103/0.048 ms
inata@maaya[~]%ping -c 4 133.5.18.169
PING 133.5.18.169 (133.5.18.169): 56 data bytes
64 bytes from 133.5.18.169: icmp_seq=0 ttl=254 time=0.866 ms
64 bytes from 133.5.18.169: icmp_seq=1 ttl=254 time=0.743 ms
64 bytes from 133.5.18.169: icmp_seq=2 ttl=254 time=0.737 ms
64 bytes from 133.5.18.169: icmp_seq=3 ttl=254 time=0.794 ms

--- 133.5.18.169 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.737/0.785/0.866/0.052 ms
inata@maaya[~]%ping -c 4 133.5.18.170
PING 133.5.18.170 (133.5.18.170): 56 data bytes
64 bytes from 133.5.18.170: icmp_seq=0 ttl=254 time=0.976 ms
64 bytes from 133.5.18.170: icmp_seq=1 ttl=254 time=0.877 ms
64 bytes from 133.5.18.170: icmp_seq=2 ttl=254 time=0.854 ms
64 bytes from 133.5.18.170: icmp_seq=3 ttl=254 time=0.820 ms

--- 133.5.18.170 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.820/0.882/0.976/0.058 ms
inata@maaya[~]%ping -c 4 133.5.18.171
PING 133.5.18.171 (133.5.18.171): 56 data bytes

--- 133.5.18.171 ping statistics ---
4 packets transmitted, 0 packets received, 100% packet loss
```

133.5.18.168

パケットロス率0%
(存在、到達可能)

133.5.18.169

パケットロス率0%
(存在、到達可能)

133.5.18.170

パケットロス率0%
(存在、到達可能)

133.5.18.171

パケットロス率100%
(存在せず、到達不可
能)



rpcinfo(1)

- RPC (Remote Procedure Control)とは
 - 異なるコンピュータ上のプログラムをあたかも通常のサブルーチン呼び出しと同様な手順で読み出すことができるようにする機構
- RPCサーバに対してRPC呼び出しを行うことで、そこに登録されているRPCプログラム情報を表示
- rpcinfoは、このRPCサービスに関する情報を表示するコマンド



rpcinfo(2)

- rpcinfoコマンドを使用すると、利用可能なRPCプログラムと、RPCプログラムが使用するポート番号が得られる
 - %rpcinfo ホスト名
- オプションとして“-p”を指定すると、ホスト名で指定されたコンピュータ上のportmapperを調べ、登録されている全てのRPCプログラムリストを表示
 - %rpcinfo -p ホスト名

```
inata@maaya[~]%rpcinfo -p minako
program vers proto port
100000 4 tcp 111 portmapper
100000 3 tcp 111 portmapper
100000 2 tcp 111 portmapper
100000 4 udp 111 portmapper
100000 3 udp 111 portmapper
100000 2 udp 111 portmapper
100007 3 udp 32779 ypbind
100007 2 udp 32779 ypbind
100007 1 udp 32779 ypbind
100007 3 tcp 32775 ypbind
100007 2 tcp 32775 ypbind
100007 1 tcp 32775 ypbind
100232 10 udp 32785
100235 1 tcp 32777
100021 1 udp 4045 nlockmgr
100021 2 udp 4045 nlockmgr
100021 3 udp 4045 nlockmgr
100021 4 udp 4045 nlockmgr
100021 1 tcp 4045 nlockmgr
100021 2 tcp 4045 nlockmgr
100021 3 tcp 4045 nlockmgr
100021 4 tcp 4045 nlockmgr
300019 1 tcp 32778
300019 1 udp 32791
100024 1 udp 32811 status
100024 1 tcp 32781 status
100133 1 udp 32811
100133 1 tcp 32781
100005 1 udp 32839 mountd
100005 2 udp 32839 mountd
100005 3 udp 32839 mountd
100005 1 tcp 32784 mountd
100005 2 tcp 32784 mountd
100005 3 tcp 32784 mountd
```




nslookup

- DNSサービスを提供するBINDプログラムのバージョン番号の取得
- IPアドレス⇔ホスト名の対応を見る
- BINDのデータベースには、ホスト名からIPアドレスに変換する情報、メールを転送するためのメール転送サーバ情報や、BIND自身のバージョン情報などが格納されている

nslookup実行例

```
inata@maaya[~]%nslookup -q=txt -class=chaos version.bind minako
Server:  minako.db.is.kyushu-u.ac.jp
Address: 133.5.18.160
Aliases: 160.18.5.133.in-addr.arpa

VERSION.BIND    text = "8.3.1-REL"
inata@maaya[~]%
```



dig

- DNSサーバについての情報を得る
 - DNSサービスを提供するBINDプログラムのバージョン番号の取得
 - %dig @ホスト名 version.bind chaos txt
 - DNSサーバの保持する情報の取得
 - %dig @ホスト名 ドメイン名 query_type
 - query_typeはa(network address),ns(name servers)など

dig実行例[1]

```
inata@maaya[~]%dig @minako version.bind chaos txt

; <<>> DiG 8.3 <<>> @minako version.bind chaos txt
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUERY SECTION:
;;     version.bind, type = TXT, class = CHAOS

;; ANSWER SECTION:
VERSION.BIND.          OS CHAOS TXT      "8.3.1-REL"

;; Total query time: 1 msec
;; FROM: maaya.4f.db.is.kyushu-u.ac.jp to SERVER: minako 133.5.18.160
;; WHEN: Thu Sep 19 21:26:19 2002
;; MSG SIZE  sent: 30  rcvd: 64

inata@maaya[~]%
```

dig実行例[2]

```
inata@maaya[~]%dig @minako db.is.kyushu-u.ac.jp ns

; <<>> DiG 8.3 <<>> @minako db.is.kyushu-u.ac.jp ns
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 3
;; QUERY SECTION:
;;      db.is.kyushu-u.ac.jp, type = NS, class = IN

;; ANSWER SECTION:
db.is.kyushu-u.ac.jp.  1D IN NS      yoda.db.is.kyushu-u.ac.jp.
db.is.kyushu-u.ac.jp.  1D IN NS      minako.db.is.kyushu-u.ac.jp.
db.is.kyushu-u.ac.jp.  1D IN NS      ami.db.is.kyushu-u.ac.jp.

;; ADDITIONAL SECTION:
yoda.db.is.kyushu-u.ac.jp.  1D IN A  133.5.18.166
minako.db.is.kyushu-u.ac.jp.  1D IN A  133.5.18.160
ami.db.is.kyushu-u.ac.jp.  1D IN A  133.5.18.197

;; Total query time: 2 msec
;; FROM: maaya.4f.db.is.kyushu-u.ac.jp to SERVER: minako 133.5.18.160
;; WHEN: Thu Sep 19 21:36:23 2002
;; MSG SIZE  sent: 38  rcvd: 144
```



実習

注:「適当なホスト」は**研究室内のホスト(133.5.18.)**にしてください

- 適当なホストに対して ping を実行し、そのホストの存在を確認
 - pingは-cオプションでパケットの量を調節する
 - %ping -c パケット数
- 適当なホストについてrpcinfo,nslookup,digコマンドを使ってみる
 - rpcinfo -p ホスト名
 - nslookup -q=txt -class=chaos version.bind ホスト名
 - dig @ホスト名 version.bind chaos txt
 - dig @ホスト名 ドメイン名 query_type