BROCADE

DNSOPS.JP BoF

2012-11-21

DNS Load balancer と俺

ブロケードコミュニケーションズ
システムズ株式会社

甲野 謙一

私は誰


新宿ゴールデン街にて

甲野 謙一(36) as KonoKono

- 2000年から様々の形でLoad Balancerを担当
- BrocadeではADXの機能設計やPOST系を担当

ブロケード

- 事業
  - L2/L3とL4-7スイッチ
  - Fiber Channel SANスイッチ

今は、SANもIP/LANも両方やってます

合体!

Load Balancer業界の変遷
とメジャープレーヤー

FOUNDRY NETWORKS

BROCADE

founded in 1996 — 2008

zeus

Top Layer Security

Alteon Networks
THE SERVER SWITCHING COMPANY

Gigabit Ethernet NIC
Jumbo Frame

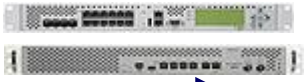
Bought by Nortel Networks

NORTEL

Bought by Radware

radware alteon

Alteon4 released



founded in 1996 — 2000 — 2009 — 2010

radware
founded in 1997

ArrowPoint
COMMUNICATIONS

CISCO

1997 Local Director

NETSCALER

CITRIX

1998

f5

founded in 1996

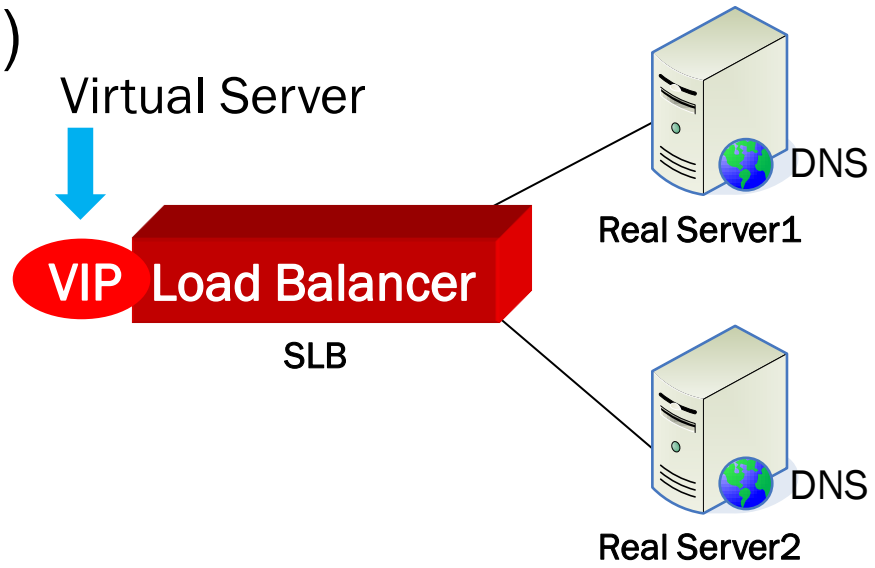
A10 Networks

Load Balancerとは

復習

4つの基本機能

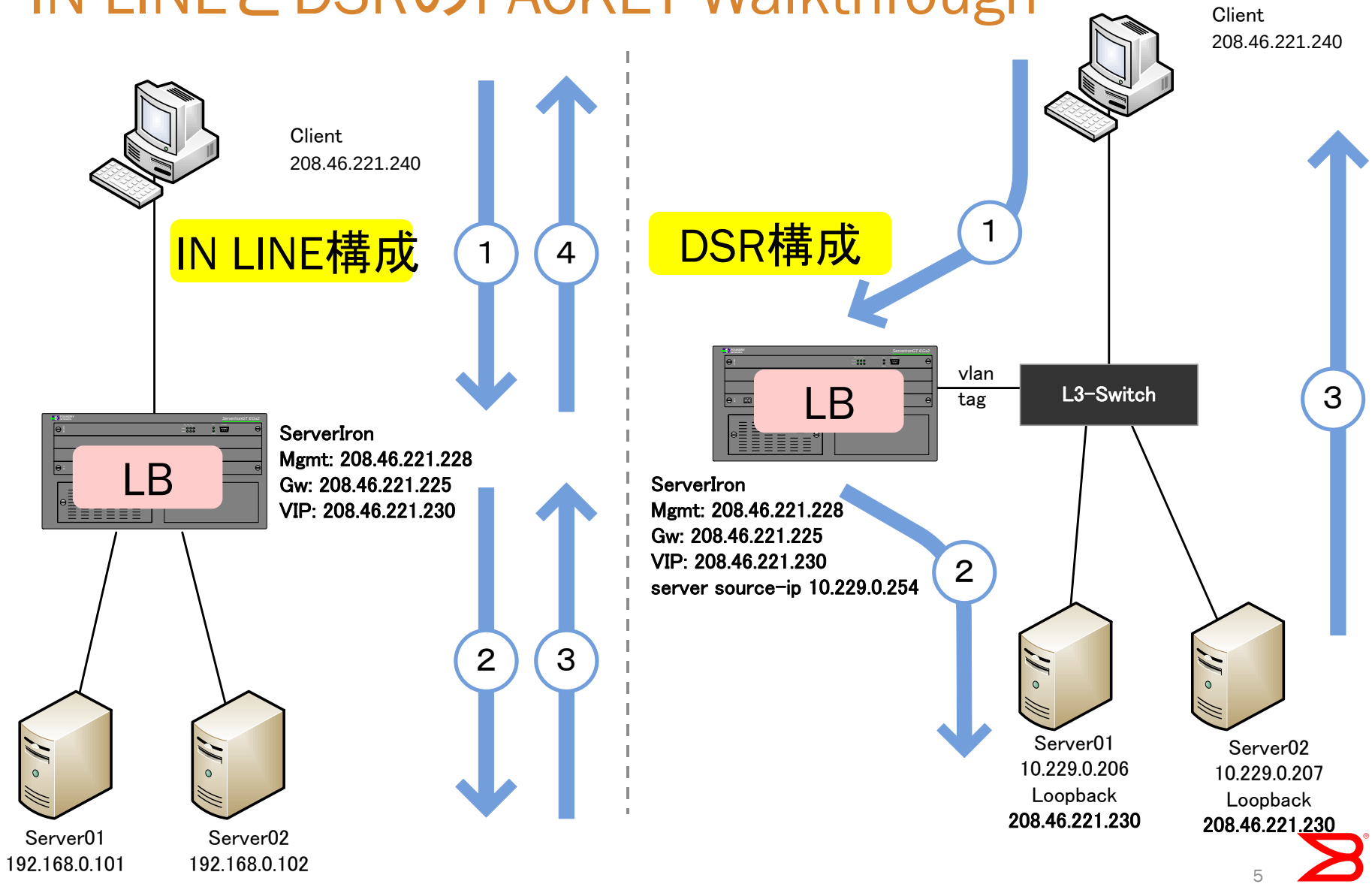
- 負荷分散機能
- アドレス変換機能(MACとIP)
- ヘルスチェック機能
- セッション維持機能



ネットワーク構成

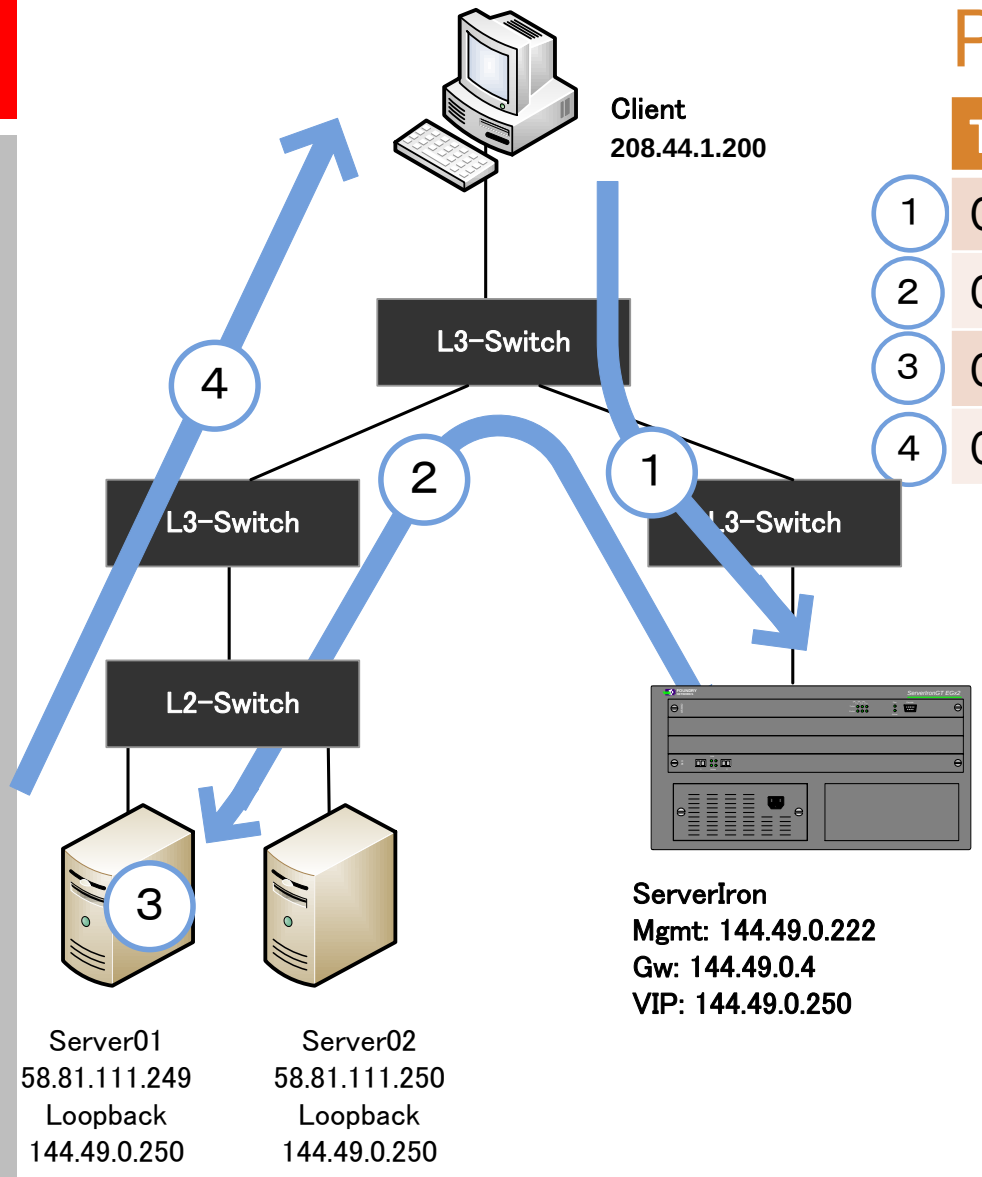
復習

IN LINEとDSRのPACKET Walkthrough



TOS Bit L3 DSR構成

ここ3年の話



PACKET Walkthrough

	TOS	Source IP	Destination IP
1	0x0	204.44.1.200	144.49.0.250
2	0x4	204.44.1.200	58.81.111.249
3	0x4	204.44.1.200	58.81.111.249
4	0x0	144.49.0.250	204.44.1.200

serverの iptable rewrite ruleで宛先を144.49.0.250に変更

FreeBSD用のReference moduleは、Yahoo!が公開している。

復習

DNS と Load balancer

よくある導入の目的

LB観点でのUDP DNSの特徴

- UDP
  - TCP FIN/RSTがないのでエントリ削除の制御Flagが無い
  - そのため、テーブルがパンクする危険性をはらむ

- UDP DNSはTransaction数が多い
  - ISPにおける Cache DNS Server
  - TLDやRoot Server

LBにおけるUDP DNS処理の特徴

- DNS queryのresponseを処理した段階で、即座にコネクションテーブルを削除
- IP Fragment時は、処理後、少し待って削除
- 短い間でFragment IDが重複しても大丈夫。実際に、どうしているかは実装依存
- 同一Source Portを使用した複数のQueryはIPヘッダのIdentificationやTransaction IDで識別することも

StatefulとStatelessの話

SLBモード	Stateful SLB	Stateless SLB
パフォーマンス (IXIAでの実測値)	普通 55万 qps (ADX1000を使用)	高速 225万 qps (ADX1000を使用)
セッション テーブルの作成	する	しない
コネクション コントロール	可能	不可
DNS Firewall	Queryの中を精査	不可
統計情報の表示	あり	なし
DNSSEC対応	IP Fragmentを含めて、どちらも対応	

Stateful SLBで接続制限する例

- Cache/権威DNSに対して同一Source Portを使用して通信されると、気が付きにくい(と思われる)
- 知らぬ間に攻撃されてることも

- LBでは識別可能なので、秒間の閾値を超えた時点で、ログだけ出力することや、一定時間制限を行う振る舞いが可能 LB導入のプラスαのモチベーション

Statelessの話

- LBとしてはパフォーマンスが高い点以外の特徴を発揮できず、実現できることが少ないのが特徴です。
- ですので、事例等をコメントさせて頂きます。

- 権威、Cache DNS共に、 Stateless SLBを採用している事業者もいますが、Statefulより圧倒的に少数
- **Brocadeの採用事例ですと、Root DNS Serverにおいて、Stateless SLB +** Anycast DNS/BGPの組み合わせで使用されています。
- 権威でもCache DNSでも、そもそも、LB使わないでサービスしているところも結構多いです

その他 TIPSなど

LBのDNS Firewall機能で頑張る

L7 DNS SLBのテンプレート

```
SeverIronADX (config-csw-dns-rule-TEST)#
  ~~snip~~
  query-dnssec-ok
  query-name
  query-rd-flag
  query-type
  ~~snip~~
  <cr>
```

```
SeverIronADX (config-csw-dns-rule-TEST)#query-type
  DECIMAL    dns-type-val
  a          Address Record
  aaaa       IPv6 Address Record
  cert       Certificate Record
  cname      Canonical Name Record
  dnskey     DNS Key Record
  ds         Delegation Signer
  ipseckey   IPSEC Key Record
  key        Key Record
  loc        Location Record
  mx         Mail Exchange Record
  ns         Name Server Record
  nsec       Next-Secure Record
  ptr        Pointer Record
  rrsig      DNSSEC Signature
  soa        Start of Authority Record
  srv        Service locator
  txt        Text Record
```

Flag (の一部) を見たり

↓

その結果によって、
- ✓ 振り分け先を変える
- ✓ rate limit をかける
- ✓ etc...

← QNAME/QTYPE を見たり、

L7 SLB記述用のスクリプト言語での例

Vulnerability Note VU#725188
ISC BIND 9 vulnerable to denial of service via dynamic update request

korori.pcap [Wires

File Edit View Go

Filter

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.210.40.253	10.210.40.1	DNS	93	Dynamic update SOA localhost

```
Frame 1: 93 bytes on wire (744 bits), 93 bytes captured (744 bits)
Ethernet II, Src: Vmware_89:8e:96 (00:50:56:89:8e:96), Dst: Vmware_89:8e:a1 (00:50:56:89:8e:a1)
Internet Protocol Version 4, Src: 10.210.40.253 (10.210.40.253), Dst
User Datagram Protocol, Src Port: 59070 (59070), Dst Port: domain (5
    Source port: 59070 (59070)
    Destination port: domain (53)
    Length: 59
  Checksum: 0xb1a4 [validation disabled]
Domain Name System (query)
    Transaction ID: 0xf497
  Flags: 0x2800 (Dynamic update)
      0... .... .... .... = Response: Message is a query
      .010 1... .... .... = Opcode: Dynamic update (5)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...0 .... .... = Recursion desired: Don't do query recursiv
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Zones: 1
    Prerequisites: 1
    Updates: 1
    Additional RRs: 0
  Zone
  Prerequisites
  Updates
```

```
0000  00 50 56 89 8e a1 00 50  56 89 8e 96 08 00 45 00   .PV....P V.....E.
0010  00 4f 00 00 40 00 40 11  d3 fc 0a d2 28 fd 0a d2   .O..@.@. ....(...
0020  28 01 e6 be 00 35 00 3b  b1 a4 f4 97 28 00 00 01   (....5.; ....(...
0030  00 01 00 01 00 00 09 6c  6f 63 61 6c 68 6f 73 74   .......l ocalhost
0040  00 00 06 00 01 c0 0c 00  ff 00 01 00 00 00 00 00   ........ ........
```

```
use OS_UDP;
use OS_SLB;

sub UDP_CLIENT_DATA{
        $payload = OS_UDP::payload;
        $mydata = unpack( "b*", $payload);
        $mystring  = substr ($mydata, 18, 4);
        if ($mystring == "0101" ) {
            OS_SLB::forward("30");
         } else {
            OS_SLB::forward("40");
         }
}
```

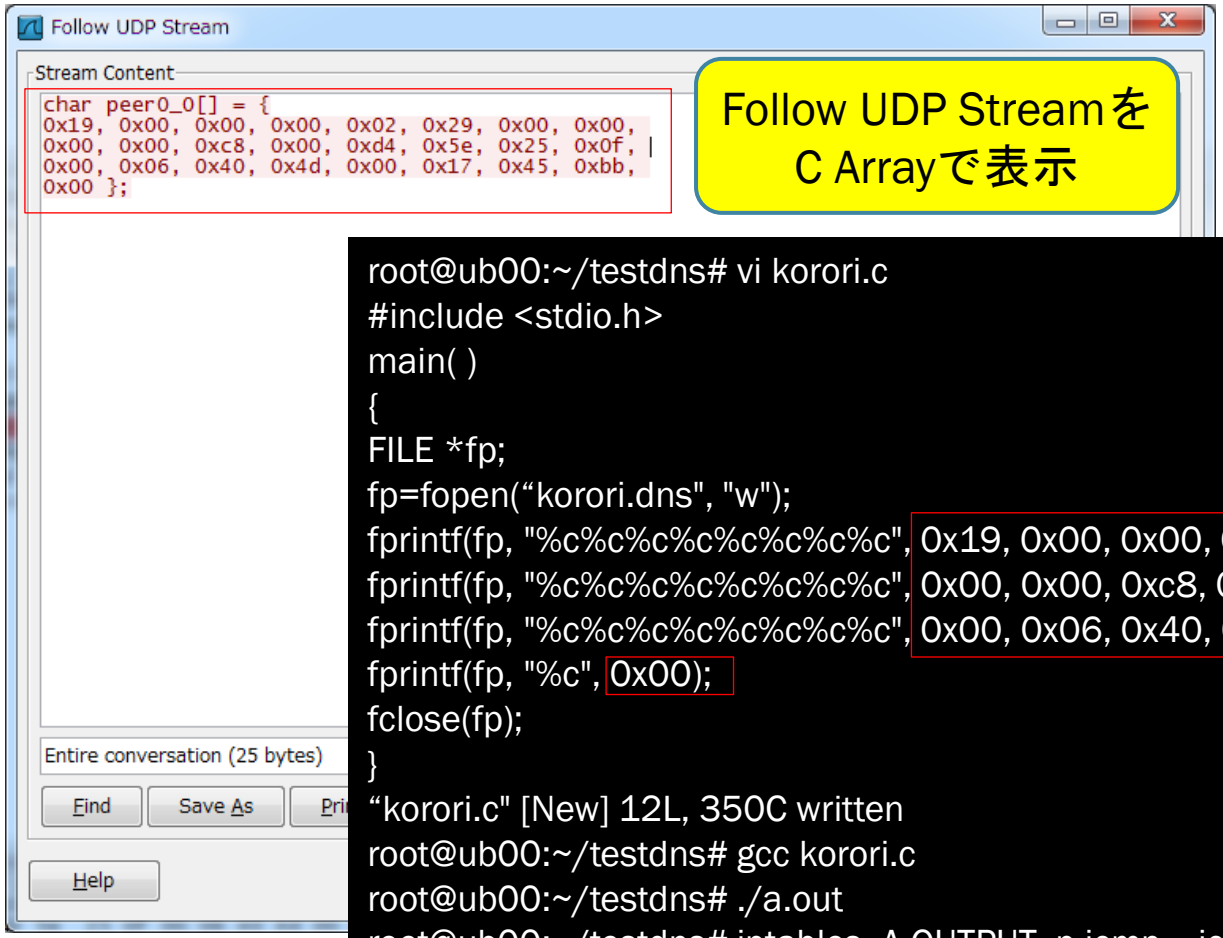
LBのOpenScript機能

```
[kkono@ns1 ~]# iptables -A INPUT -p udp --dport 53 -j DROP -m u32 --u32 '30>>27&0xF=5'
```

User Datagram Protocol (udp), 8 bytes | Packets: 1 Displayed: 1 Marked: 0 Load time: 0:00.187 | Profile: Default

DNSパケットの作り方

wiresharkとhping3を使用した場合



Follow UDP Streamを
C Arrayで表示

```
root@ub00:~/testdns# vi korori.c
#include <stdio.h>
main( )
{
FILE *fp;
fp=fopen("korori.dns", "w");
fprintf(fp, "%c%c%c%c%c%c%c%c", 0x19, 0x00, 0x00, 0x00, 0x02, 0x29, 0x00, 0x00);
fprintf(fp, "%c%c%c%c%c%c%c%c", 0x00, 0x00, 0xc8, 0x00, 0xd4, 0x5e, 0x25, 0x0f);
fprintf(fp, "%c%c%c%c%c%c%c%c", 0x00, 0x06, 0x40, 0x4d, 0x00, 0x17, 0x45, 0xbb);
fprintf(fp, "%c", 0x00);
fclose(fp);
}
"korori.c" [New] 12L, 350C written
root@ub00:~/testdns# gcc korori.c
root@ub00:~/testdns# ./a.out
root@ub00:~/testdns# iptables -A OUTPUT -p icmp  --icmp-type port-unreachable -j DROP
root@ub00:~/testdns# hping3 -2 -p 53 -E korori.dns -d 25 10.210.40.1 -c 1
```

ちなみに、3-wayhandshakeが必要なTCPだと、python scapyを使ってます

BROCADE

Thank you!