

# Enhanced Port Forwarding functions with CGNAT

draft-chan-tsvwg-eipf-cgnat-02.txt

Louis Chan

Juniper Networks

IETF 116, Mar 2023

# draft-chan-tsvwg-eipf-cgnat-02.txt

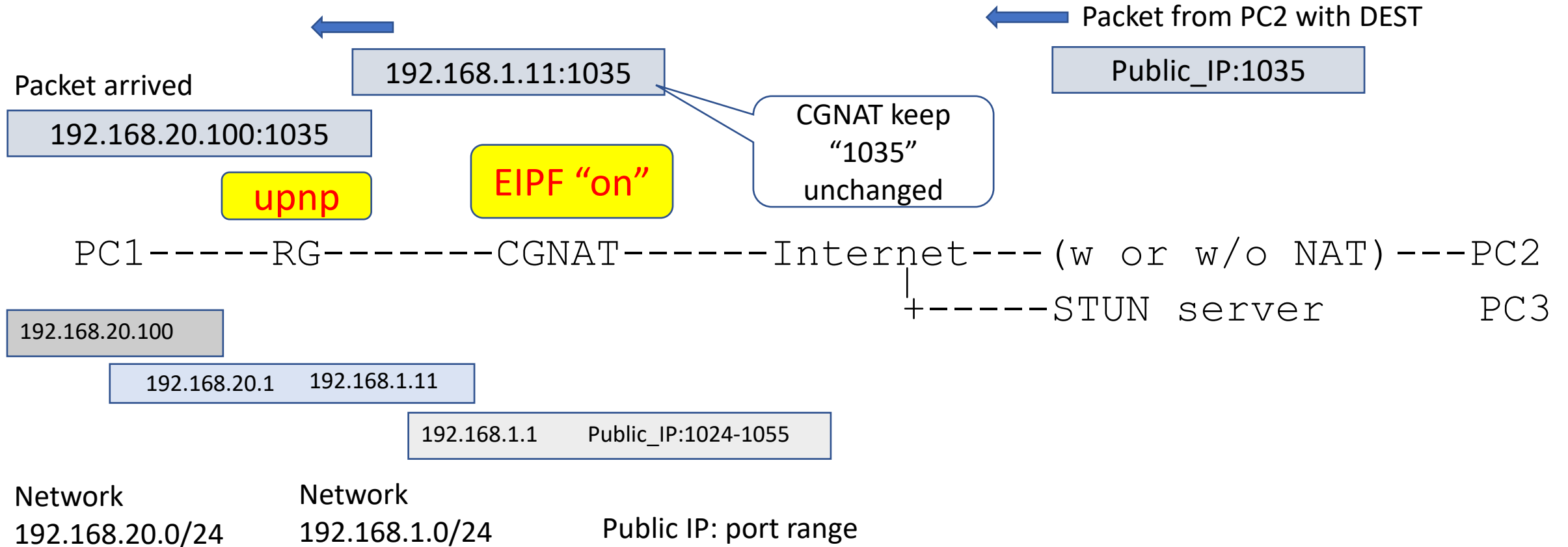
Problem statement:

- RFC5128 provides methods for setting up P2P connection behind NAT44. However,
  - Only works for UDP in live situation
  - For TCP, it has low success rate.
    - e.g. Direct TCP connection for webcam does not work
  - It hole punching method needs a common 3<sup>rd</sup> party server
- Need a solution working for TCP (plus UDP) under CGNAT
  - Each party could run independently
- It requires CGNAT to support EIPF (Endpoint Independent Port Forwarding)
  - Compatible with EIM

# Endpoint Independent port forwarding (EIPF) Enhancement

- Allow TCP/UDP incoming connection through CGNAT WITHOUT changing the DEST port
  - DEST port is actually allocated from CGNAT as outgoing source port per private IP
- Allow chain of forwarding of the same DEST port from CGNAT, RG and hence to the end device
  - Note: One TCP/UDP could only be forward to ONE selected private IP behind RG in incoming direction.
    - E.g. public 200.1.1.1:1234 could only be point to one private IP, like 192.168.1.10 for incoming session
    - But multiple devices behind the RG, depending on configuration, could be potentially allowed to share 200.1.1.1:1234 as source port for outgoing connections so long as there is no clash of session.

# Demo: incoming TCP session for NAT444



1. Use STUN to discover opening port (1035 in this demo)
2. Use UPNP to enable port forwarding at RG
3. TCP services allowed

# PC1@192.168.20.100

```
root@debian8-upnp:/home/louis/upnp#  
root@debian8-upnp:/home/louis/upnp#  
root@debian8-upnp:/home/louis/upnp#  
root@debian8-upnp:/home/louis/upnp#  
root@debian8-upnp:/home/louis/upnp# sh service.sh  
web service  
1. checking stun  
XorMappedAddress = 58.152.214.89:1035
```

Public IP  
w/ external port 1035  
detected

```
2. request to RG via upnp  
external 192.168.1.11:1035 TCP is redirected to internal 192.168.20.100:1035 (duration=0)
```

```
3. start http server
```

```
Try http://58.152.214.89:1035
```

Start http server  
locally with port  
1035

Request to RG for port  
mapping TCP 1035 to  
local host

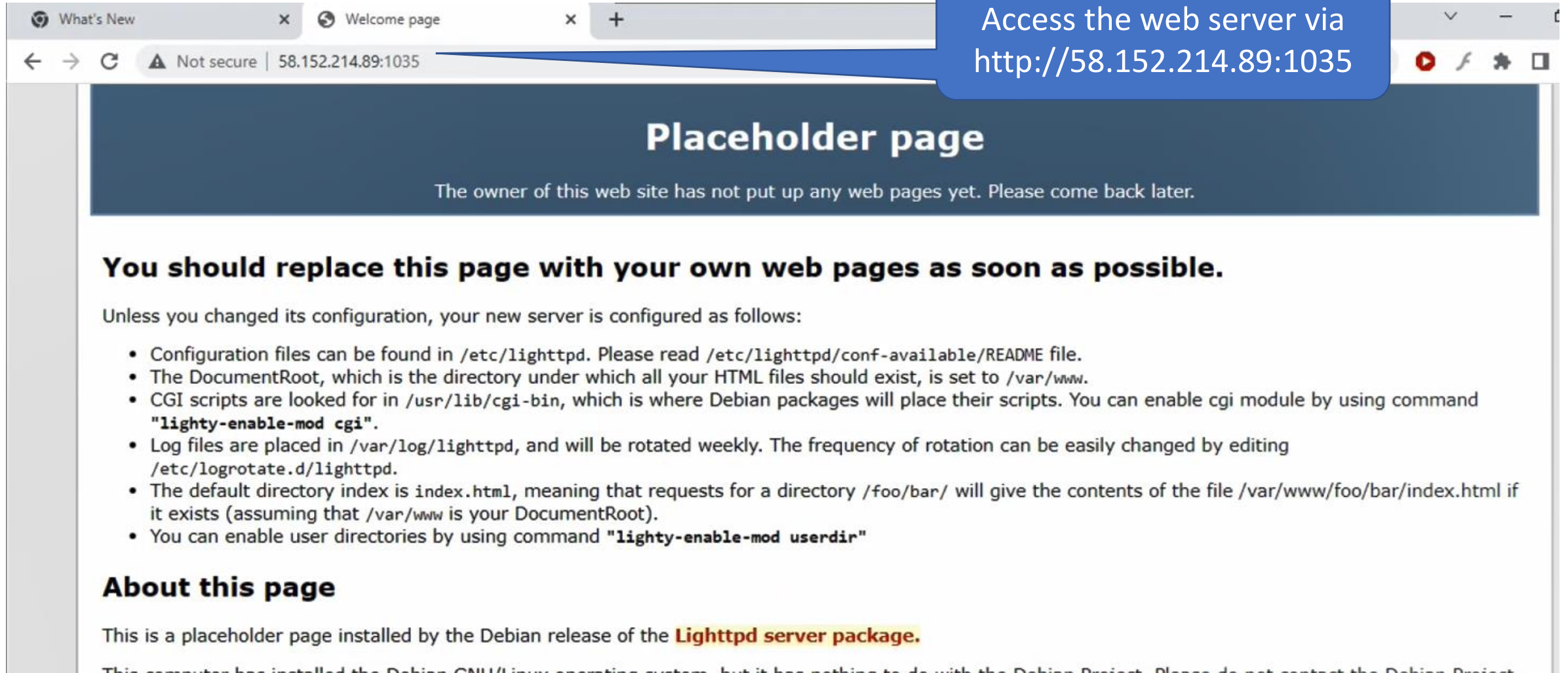
```
=====  
ssh/sftp service  
using similar procedure  
XorMappedAddress = 58.152.214.89:1037
```

```
Try ssh -p 1037 louis@58.152.214.89
```

Use the same  
procedure, and redirect  
port 1037 at RG to local  
ssh port 22

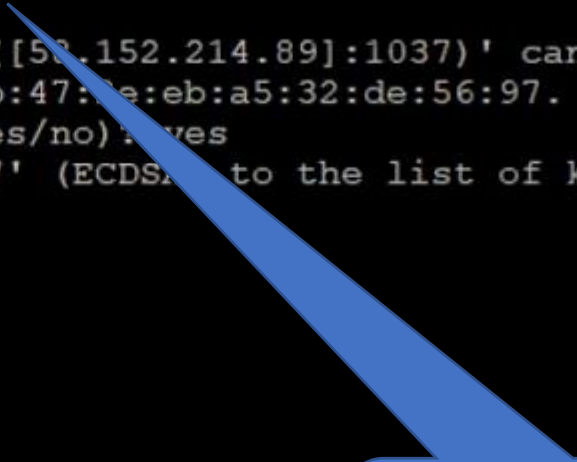
```
root@debian8-upnp:/home/louis/upnp#
```

# PC2: Test the Web service



# PC3: access the ssh service

```
root@pi-deb8:~#  
root@pi-deb8:~#  
root@pi-deb8:~# ssh -p 1037 louis@58.152.214.89  
The authenticity of host '[58.152.214.89]:1037 ([58.152.214.89]:1037)' can't be established.  
ECDSA key fingerprint is f4:c9:ea:c7:15:36:ad:2b:47:ae:eb:a5:32:de:56:97.  
Are you sure you want to continue connecting (yes/no): yes  
Warning: Permanently added '[58.152.214.89]:1037' (ECDSA) to the list of known hosts.  
louis@58.152.214.89's password:  
Permission denied, please try again.  
louis@58.152.214.89's password:  
Permission denied, please try again.  
louis@58.152.214.89's password:  
Permission denied (publickey,password).  
root@pi-deb8:~#  
root@pi-deb8:~#  
root@pi-deb8:~#  
root@pi-deb8:~#
```



Access the ssh via  
58.152.214.89:1037  
from public internet



# RG: iptables (nat translation table)

```
root@DD-WRT x86:~#
root@DD-WRT x86:~# iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source                destination
DNAT       tcp  --  anywhere               192.168.1.11
DNAT       tcp  --  anywhere               192.168.1.11
DNAT       udp  --  anywhere               192.168.1.11
DNAT       tcp  --  anywhere               192.168.1.11
DNAT       tcp  --  anywhere               192.168.1.11
DNAT       icmp --  anywhere               192.168.1.11
TRIGGER    0    --  anywhere               192.168.1.11

Chain POSTROUTING (policy ACCEPT)
target     prot opt source                destination
MASQUERADE 0    --  anywhere               anywhere
RETURN     0    --  anywhere               anywhere
MASQUERADE 0    --  192.168.20.0/24        192.168.20.0/24

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
root@DD-WRT x86:~#
```

□

```
tcp dpt:1037 to:192.168.20.100:22
tcp dpt:1035 to:192.168.20.100:1035
udp dpt:1035 to:192.168.20.100:1035
tcp dpt:webcache to:192.168.20.1:80
tcp dpt:ssh to:192.168.20.1:22
to:192.168.20.1
TRIGGER type:dnat match:0 relate:0

PKTTYPE = http
```

Port forwarding@RG  
Request via UPNP for  
TCP port 1035 and 1037



# Others

- Demo video on youtube
  - <https://is.gd/mn16ju>



- Seek for comment and usefulness in live situation