anti IP spoofing technique

MATSUZAKI ‘maz’ Yoshinobu
<maz@iij.ad.jp>

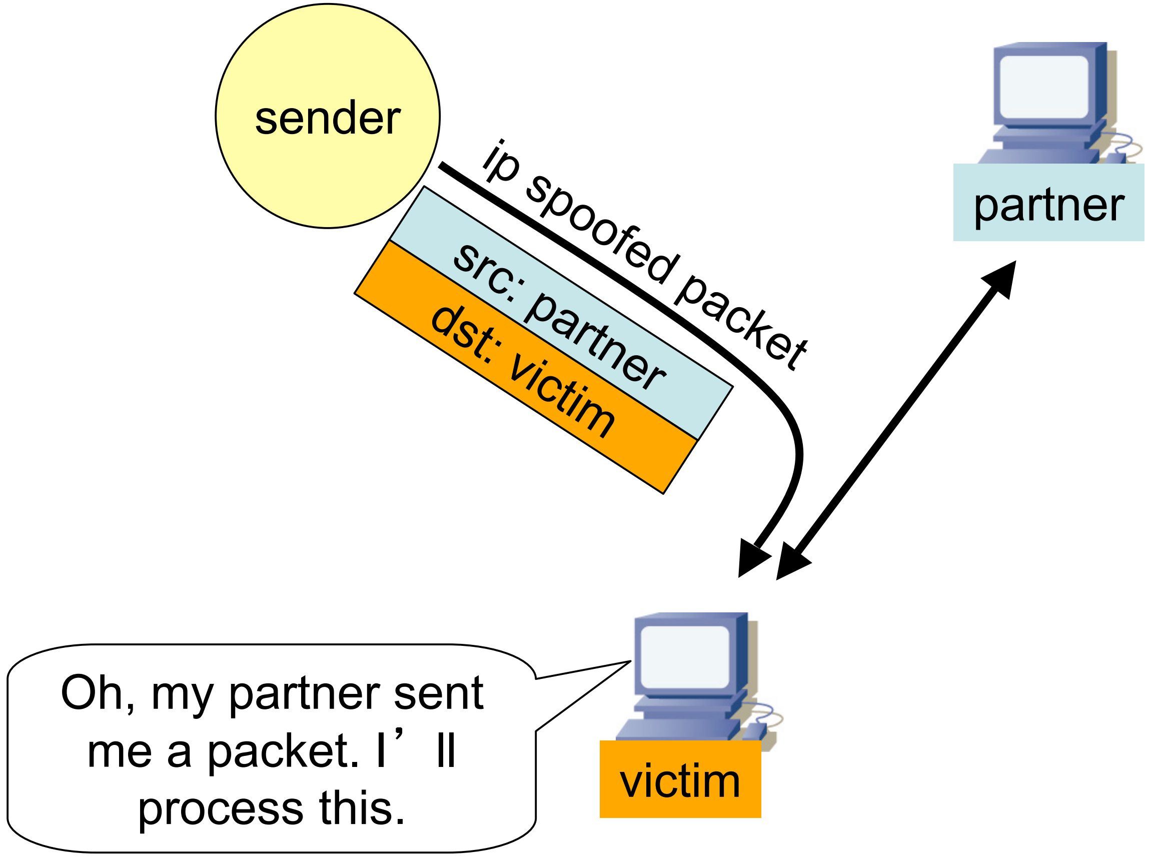
ip spoofing

creation of IP packets with source addresses other than those assigned to that host

Malicious uses with IP spoofing

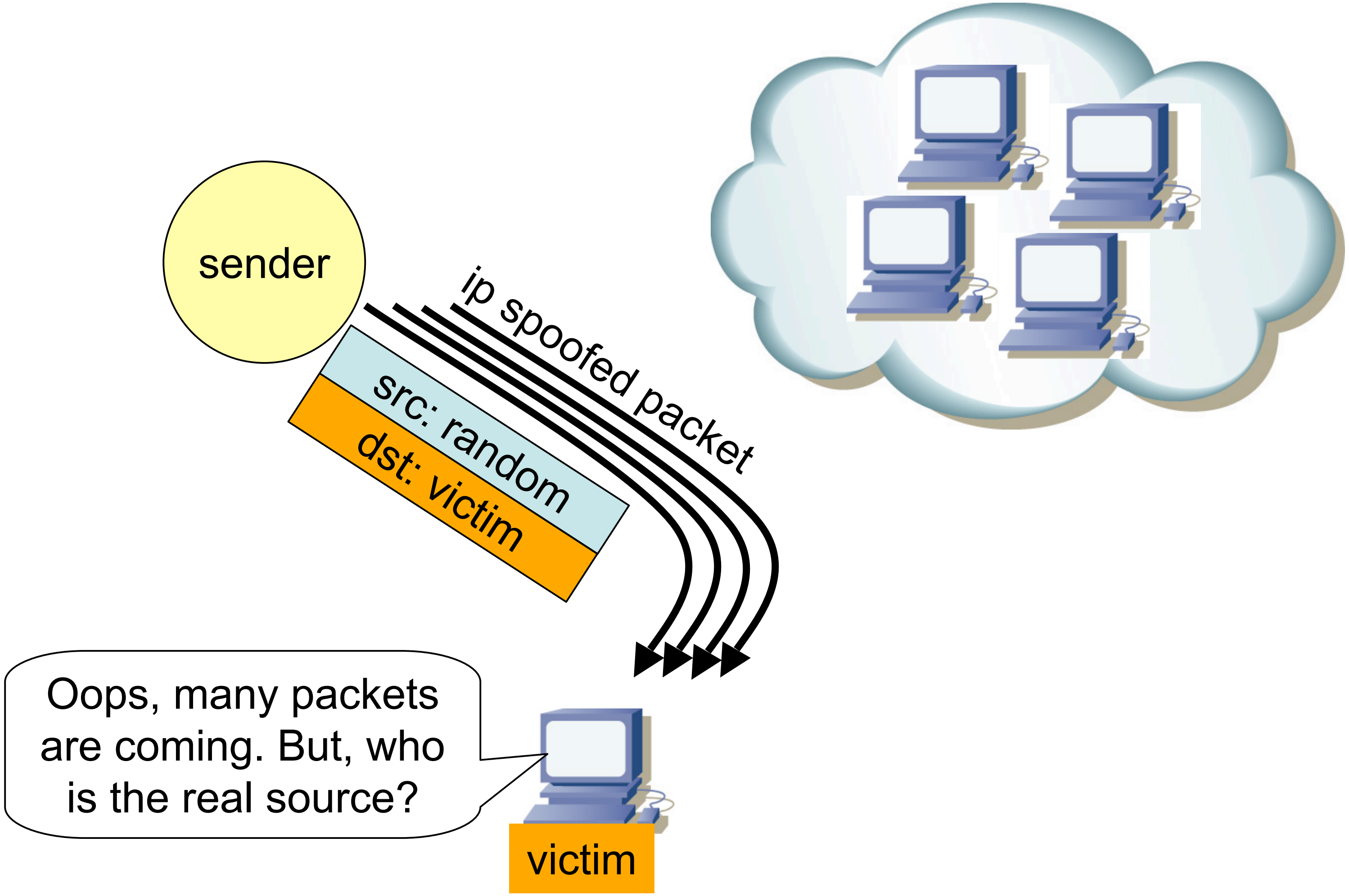
- impersonation
  - session hijack or reset
- hiding
  - flooding attack
- reflection
  - ip reflected attack

impersonation

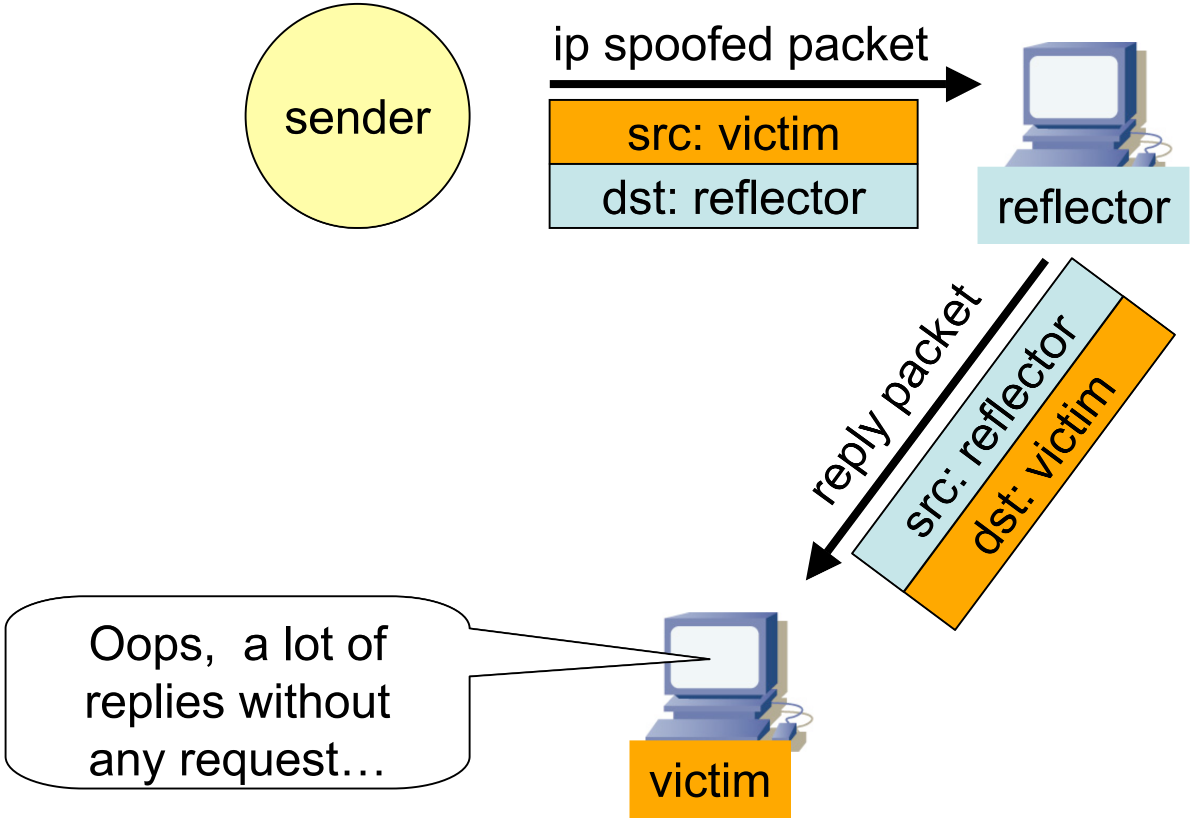


Oh, my partner sent me a packet. I' ll process this.

hiding


sender
ip spoofed packet
src: random
dst: victim
Oops, many packets are coming. But, who is the real source?
victim

reflection



Oops, a lot of replies without any request…

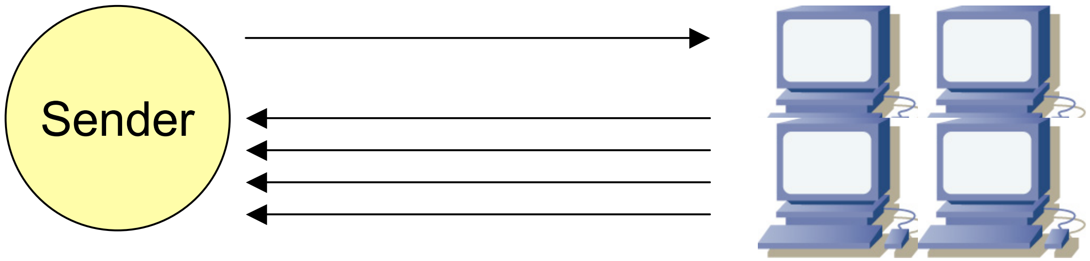
Copyright (C) 2006 Internet Initiative Japan Inc.

ip reflected attacks

- smurf attacks
  - icmp echo (ping)
  - ip spoofing (reflection)
  - directed-broadcast amplification
- **dns amplification attacks**
  - **dns query**
  - **ip spoofing (reflection)**
  - **DNS amplification**

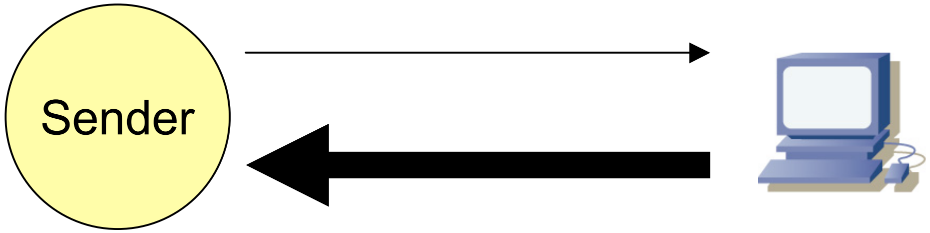
amplification

1. multiple replies



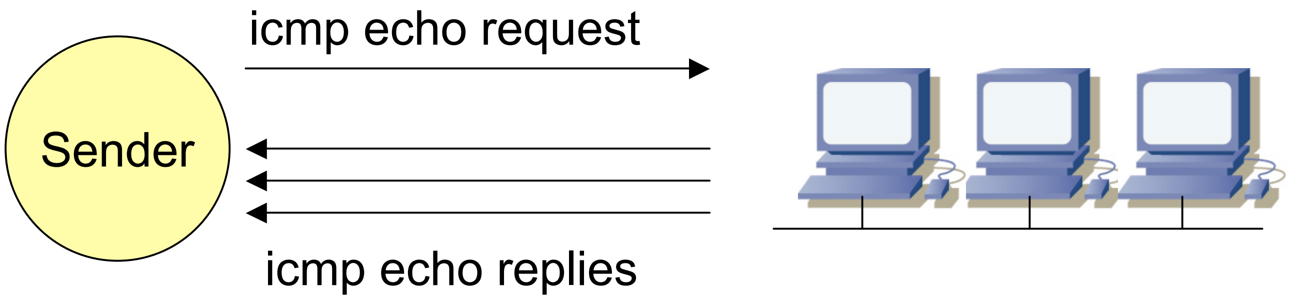
Sender

2. bigger reply

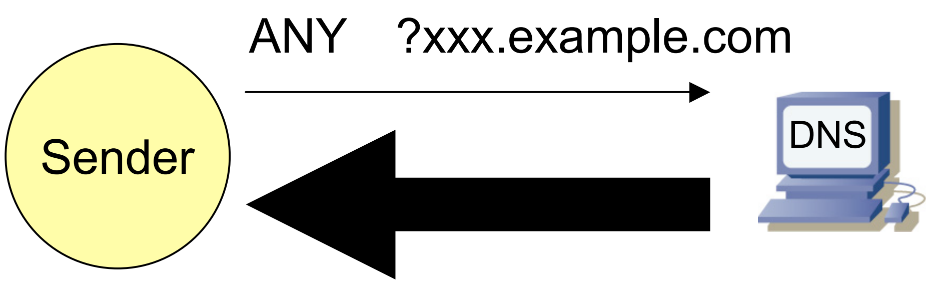


Sender

directed-broadcast amplification



DNS amplification



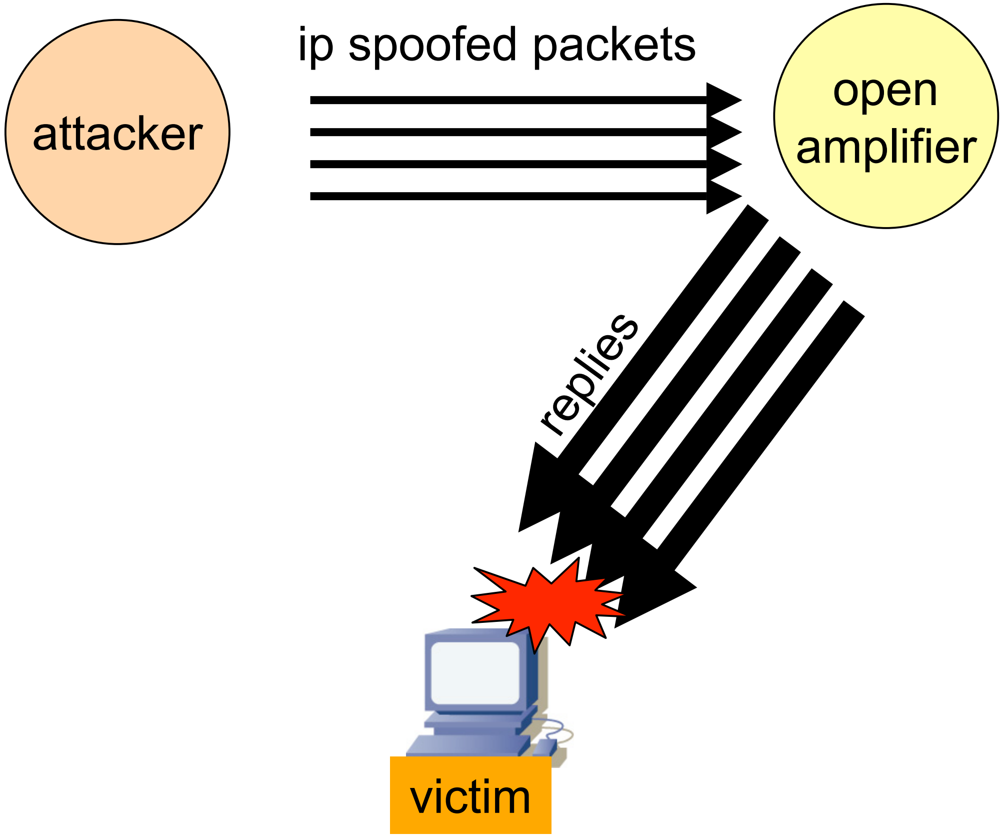
ANY ?xxx.example.com

Sender

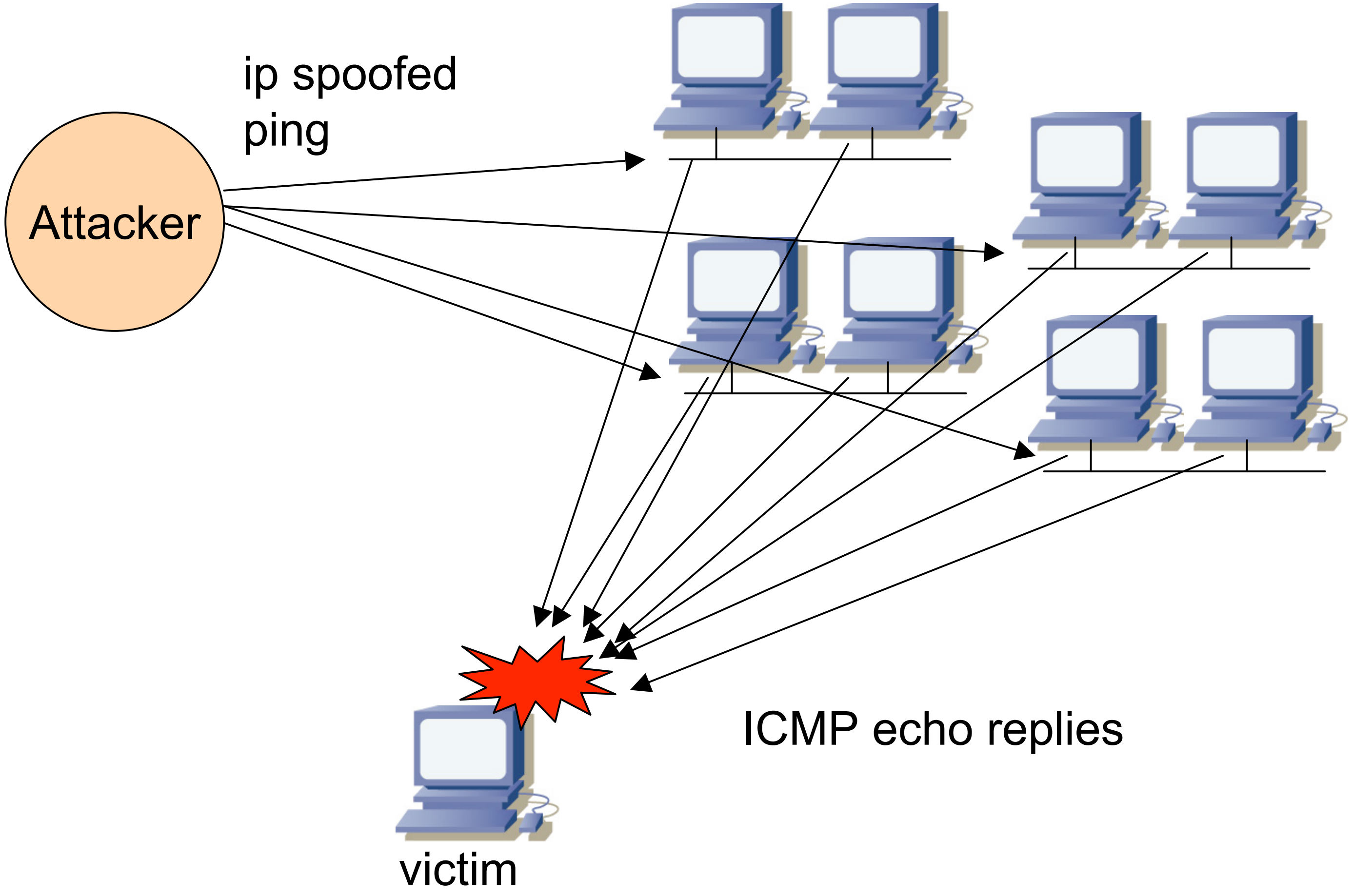
DNS

xxx.example.com IN TXT
XXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX

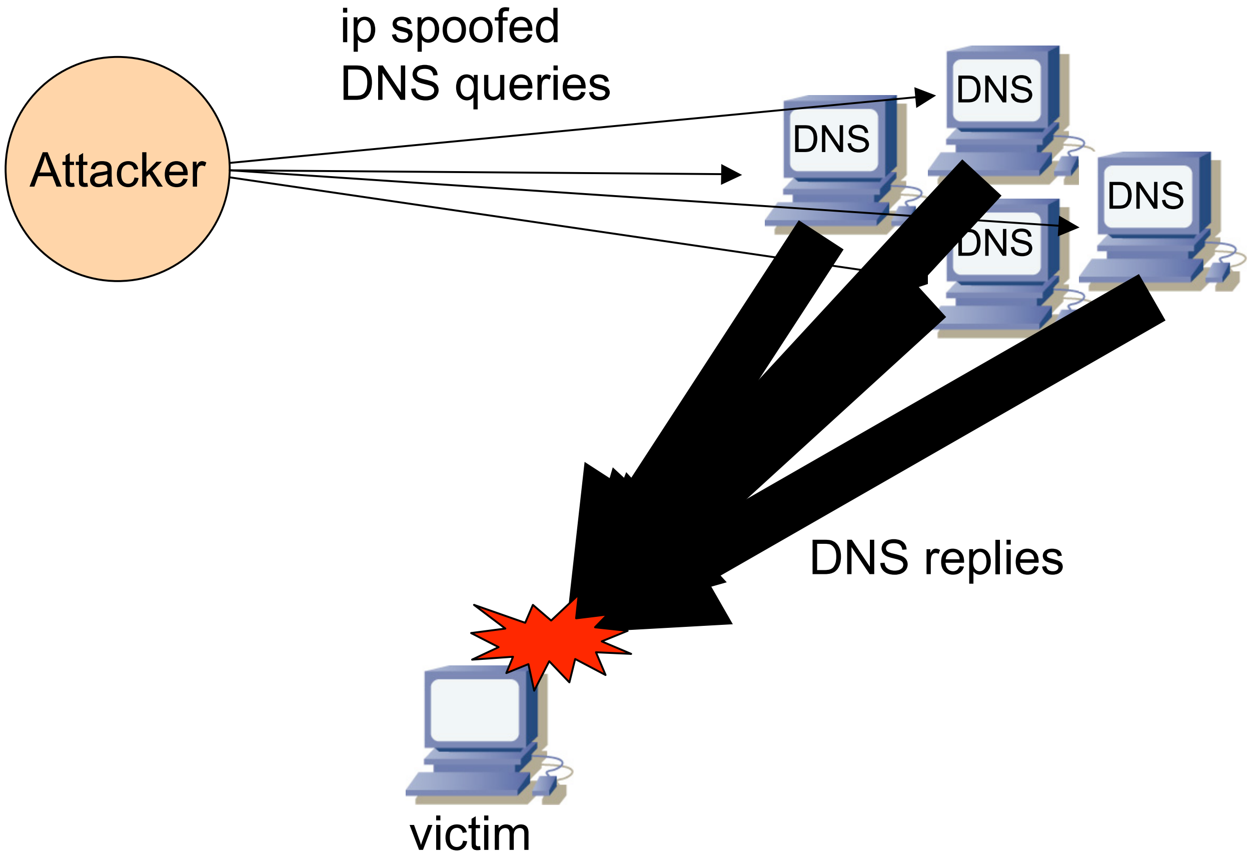
ip reflected attacks



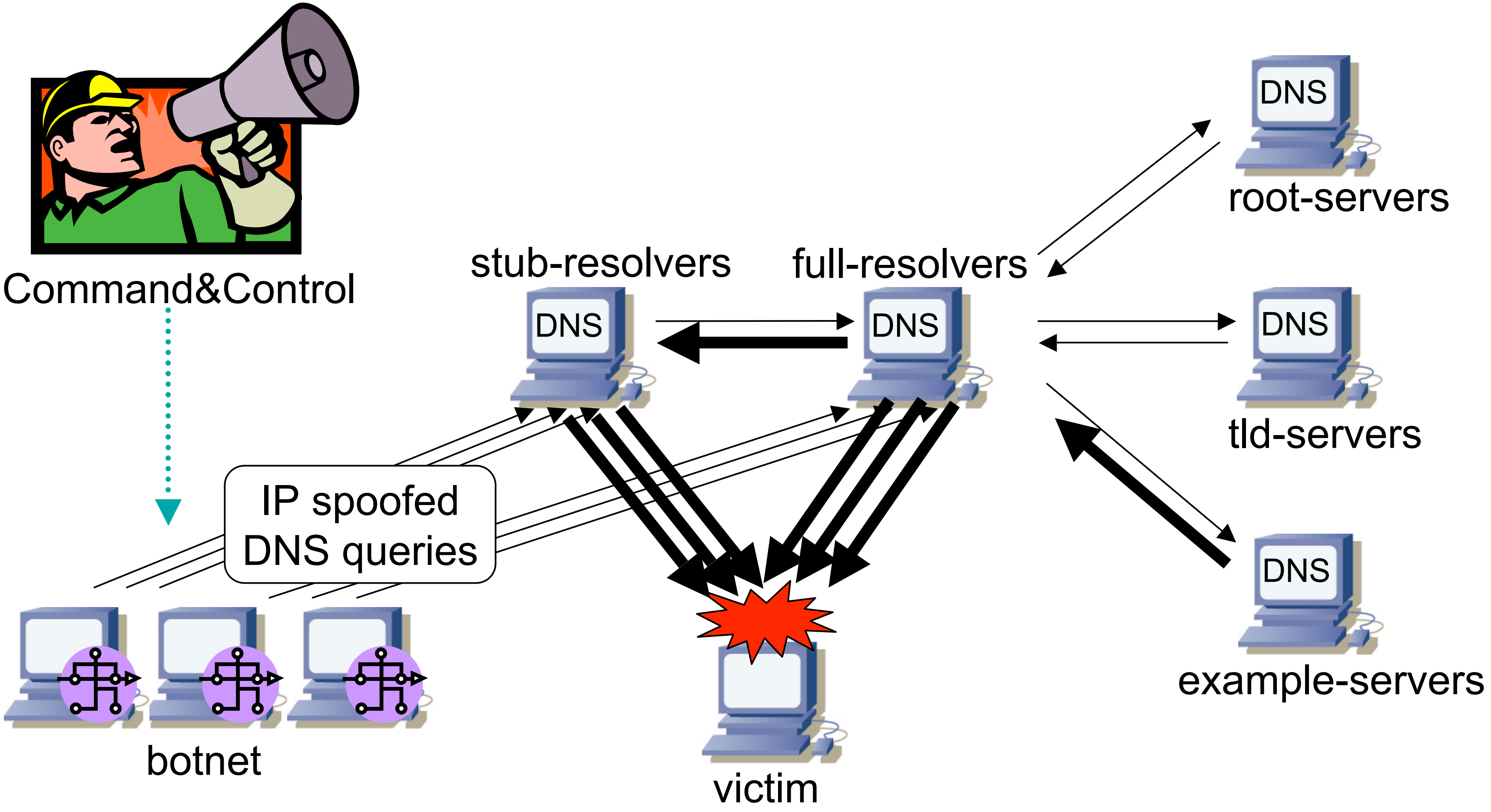
smurf attack



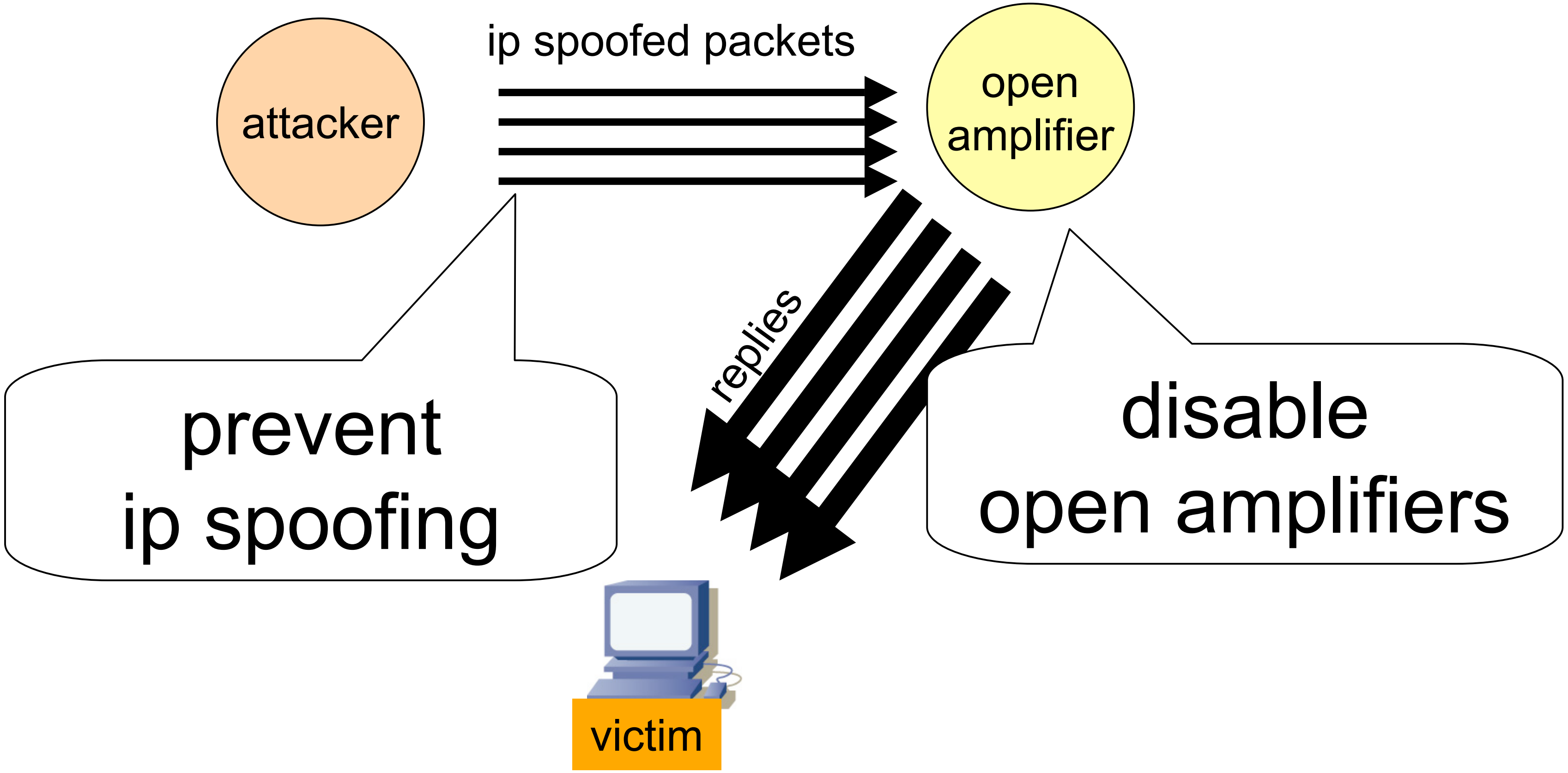
dns amplification attack



relations – dns amp attack



solutions for ip reflected attacks



prevent ip spoofing

disable open amplifiers

two solutions

- disable ‘open amplifier’
  - disable ‘directed-broadcast’
  - disable ‘open recursive DNS server’
    - contents DNS server should accept queries from everyone, but service of resolver (cache) DNS server should be restricted to its customer only.
- prevent ip spoofing!!
  - source address validation
  - BCP38 & BCP84

Source Address Validation

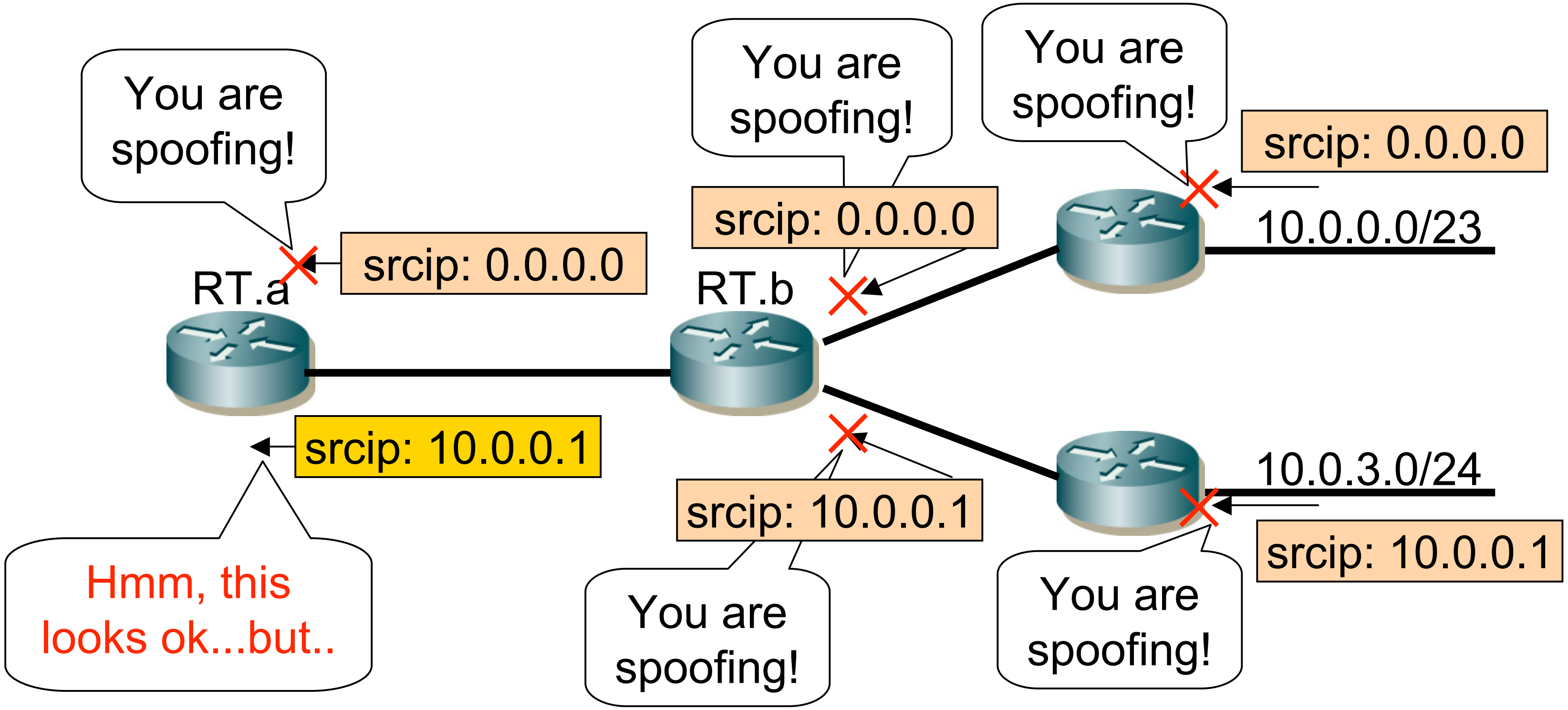
- Check the source ip address of ip packets
  - filter invalid source ip address
  - filter close to the packets origin as possible
  - filter precisely as possible

- If no networks allow ip spoofing, we can eliminate these kinds of attacks

our assumption

- ISP/network administrator assign ip address for their users.
  - dynamic or static
  - DHCP, connectivity service
- Users should use these assigned ip address as their source ip address.

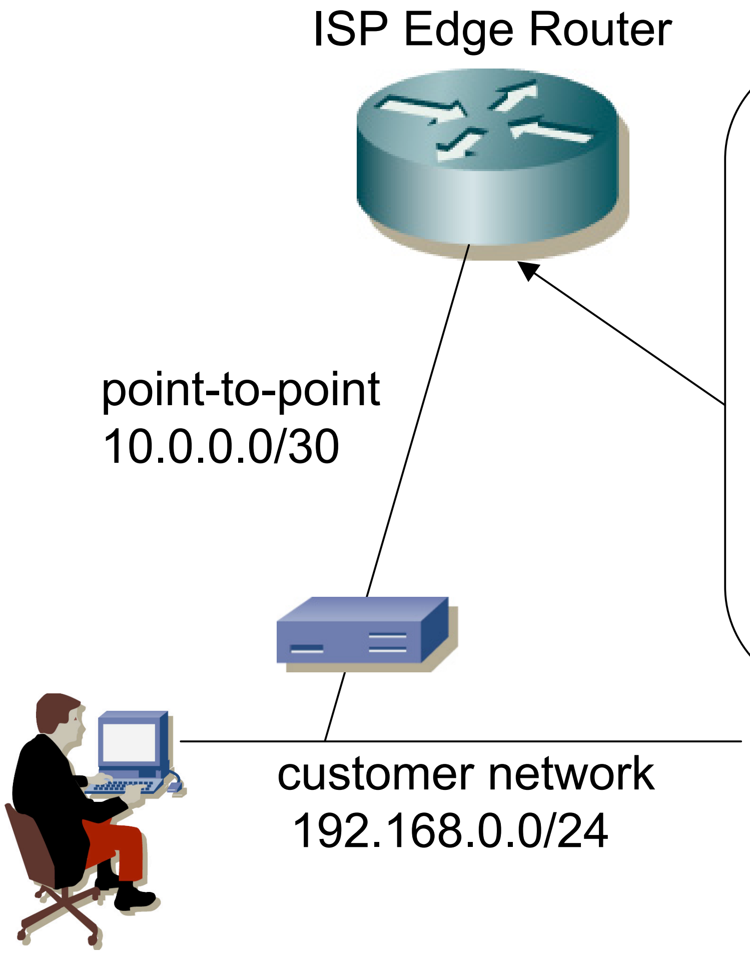
close to the origin



how to configure the checking

- ACL
  - – packet filter
  - – permit valid-source, then drop any
- uRPF check
  - – check incoming packets using ‘routing table’
  - – look-up the return path for the source ip address
  - – loose mode can’t stop ip reflected attacks
    - use strict mode or feasible mode

cisco ACL example



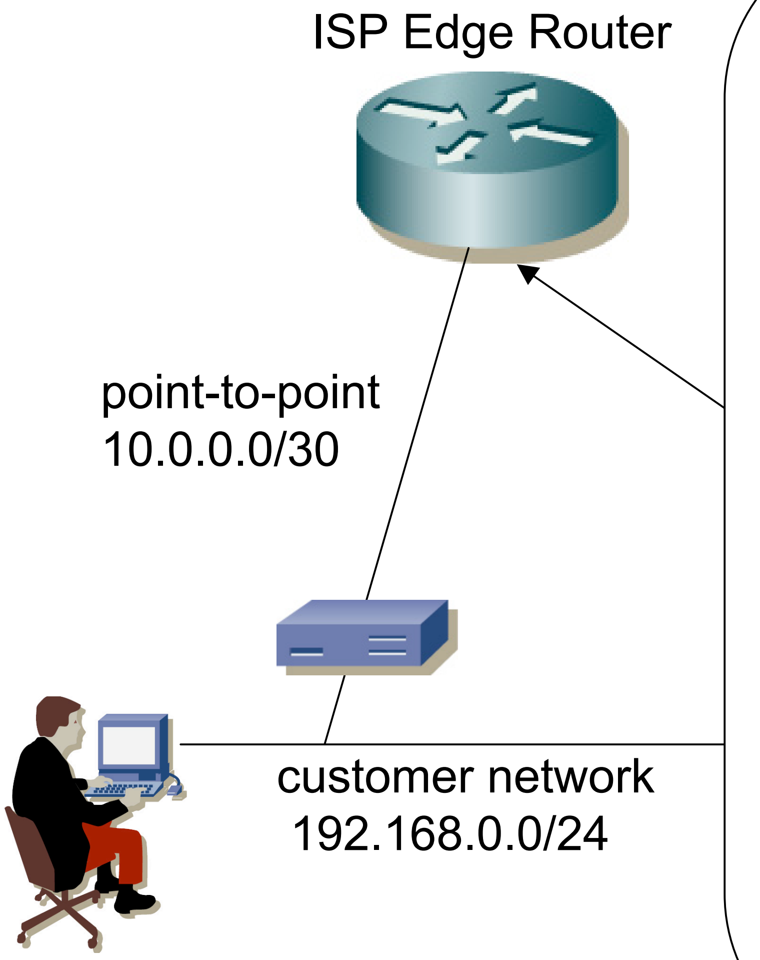
ISP Edge Router

point-to-point
10.0.0.0/30

customer network
192.168.0.0/24

```
ip access-list extended fromCUSTMER
 permit ip 192.168.0.0 0.0.255.255 any
 permit ip 10.0.0.0 0.0.0.3 any
 deny   ip  any any
!
interface Gigabitethernet0/0
 ip access-group fromCUSTOMER in
!
```

juniper ACL example



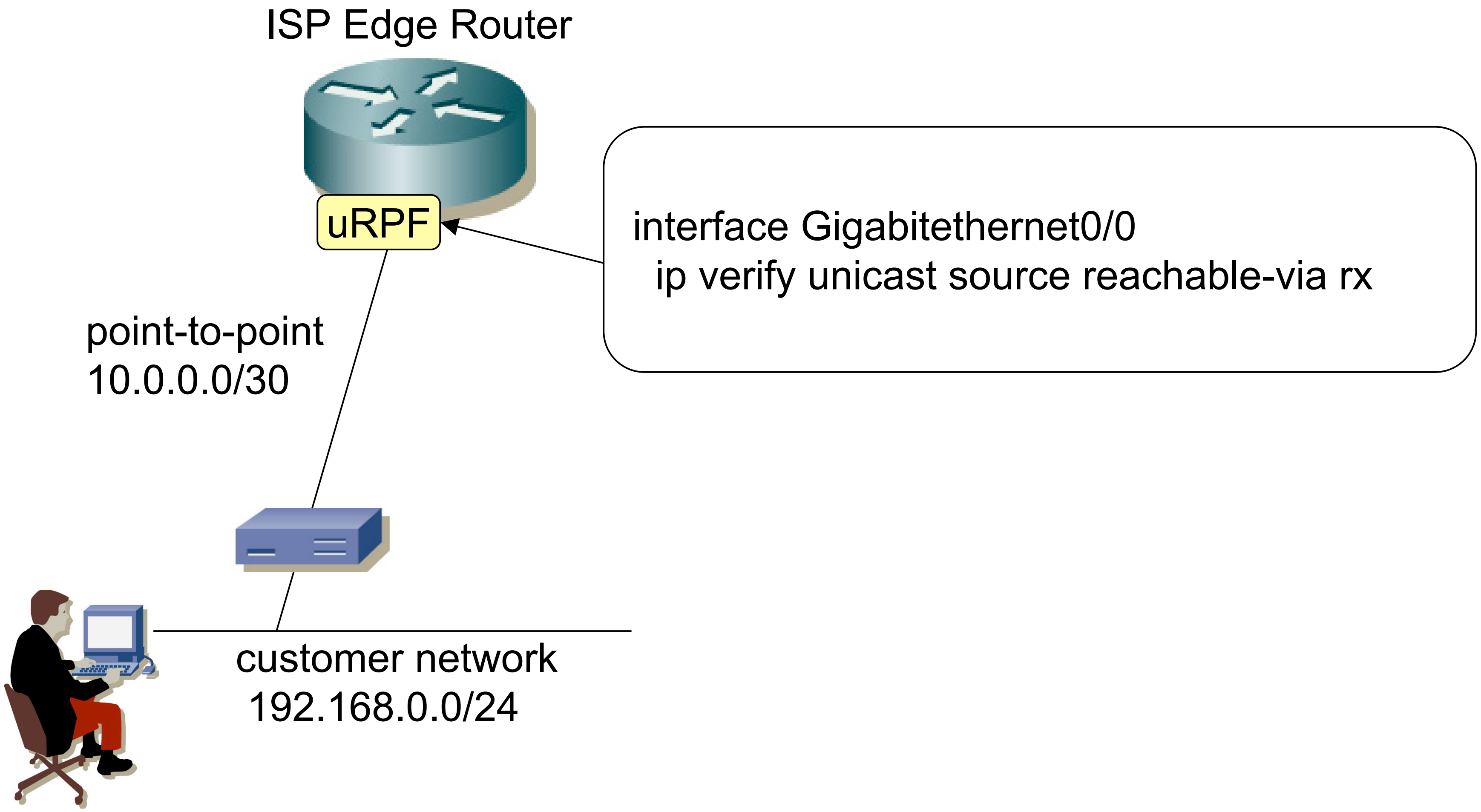
ISP Edge Router

point-to-point
10.0.0.0/30

customer network
192.168.0.0/24

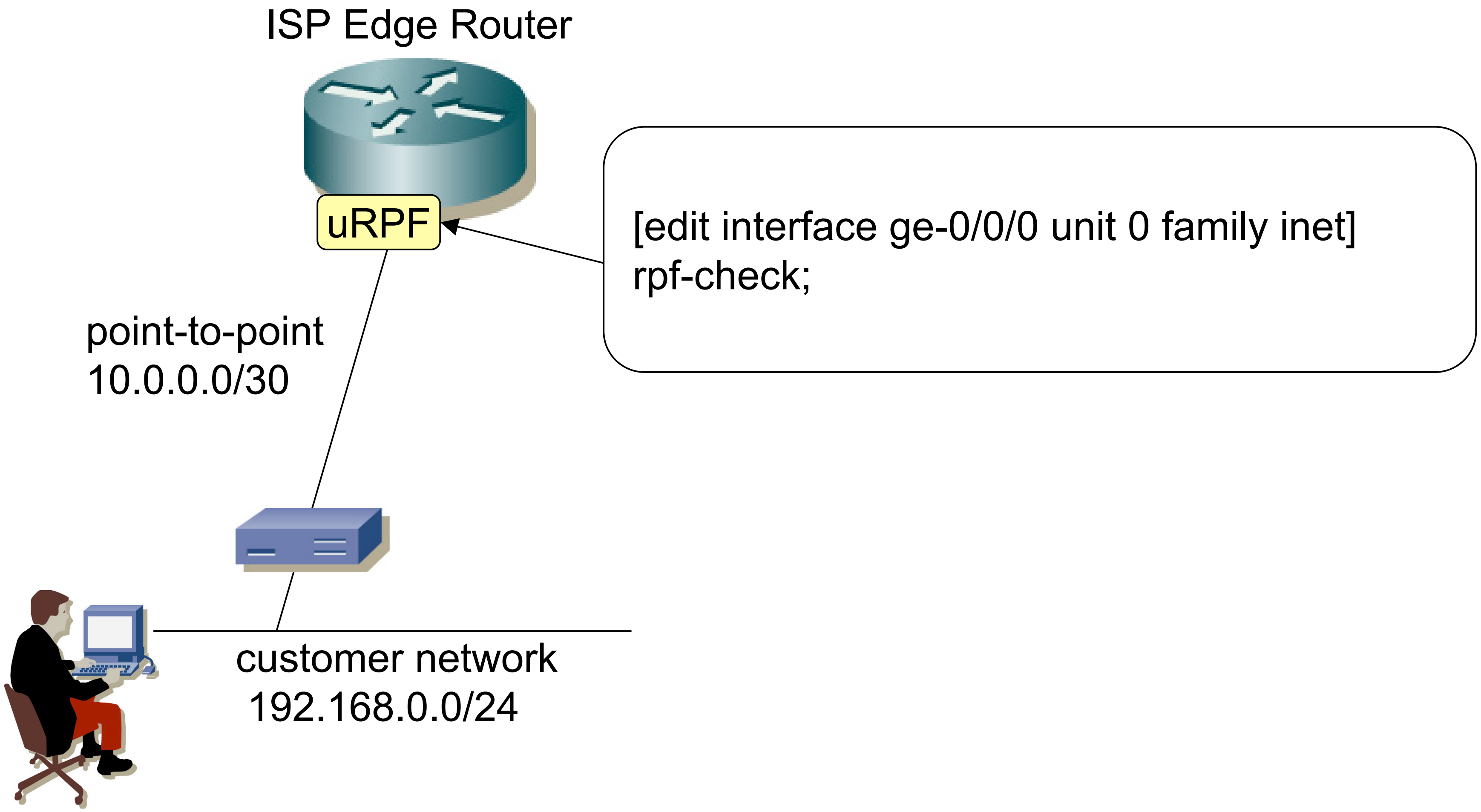
```
firewall family inet {
 filter fromCUSTOMER {
  term CUSTOMER {
   from source-address {
    192.168.0.0/16;
    10.0.0.0/30;
   }
   then accept;
  }
  term Default {
   then discard;
  }
 }
}
[edit interface ge-0/0/0 unit 0 family inet]
filter {
 input fromCUSTOMER;
}
```

cisco uRPF example



```
interface Gigabitethernet0/0
  ip verify unicast source reachable-via rx
```

juniper uRPF example



ISP Edge Router

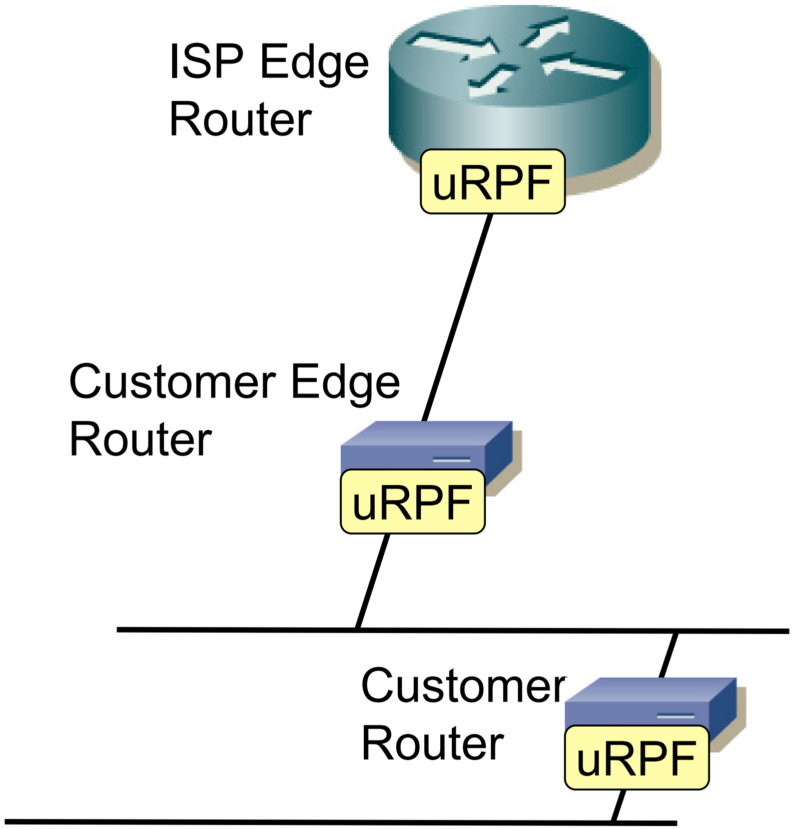
uRPF

```
[edit interface ge-0/0/0 unit 0 family inet]
rpf-check;
```

point-to-point
10.0.0.0/30

customer network
192.168.0.0/24

multistage verification



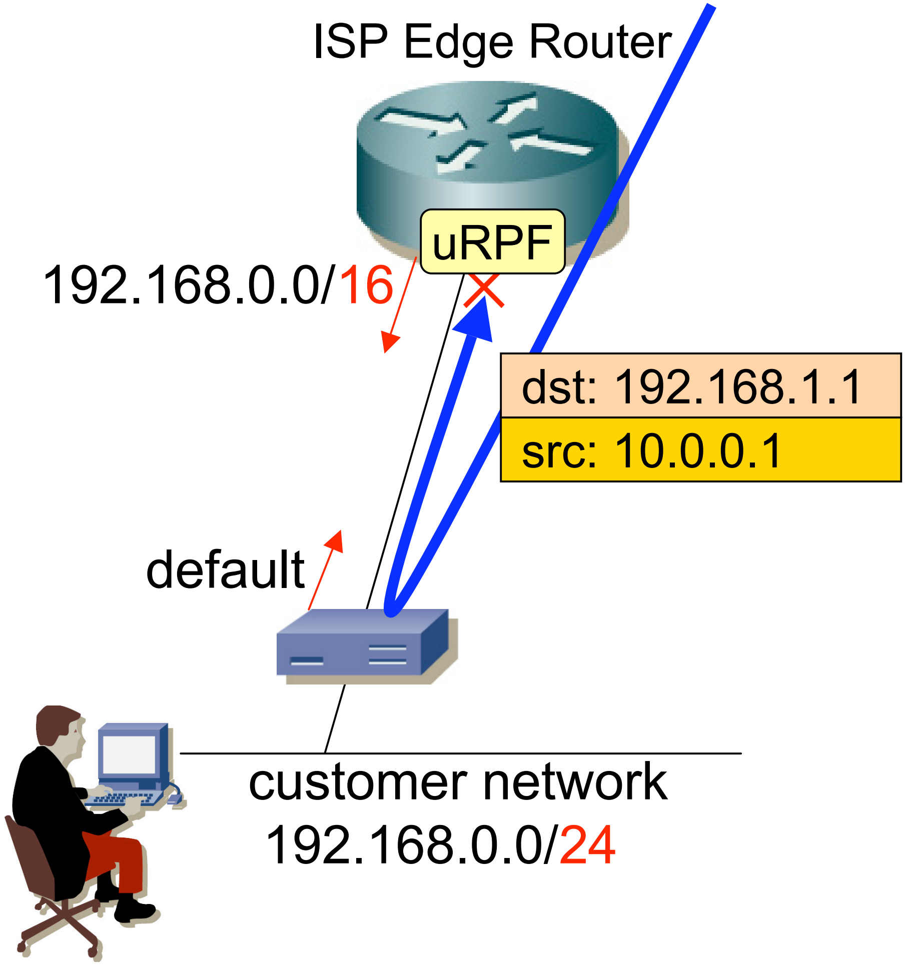
- customers know their network. ☺
- good for precise filter

- We can filter spoofed traffic at earliy stage.

uRPF - failures

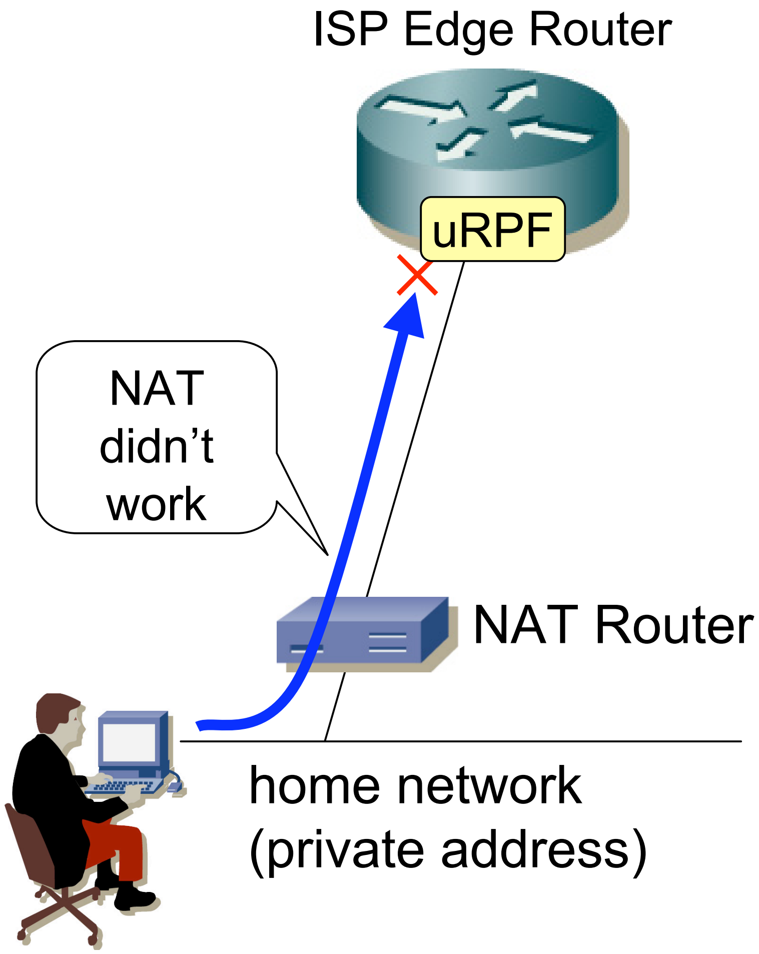
- common failures
  - – unused space
  - – private space
  - – wrong address
- asymmetric routing failures
  - – multi-connected network
  - – transit LAN
- special failures
  - – private/non-routed backbone network

unused space



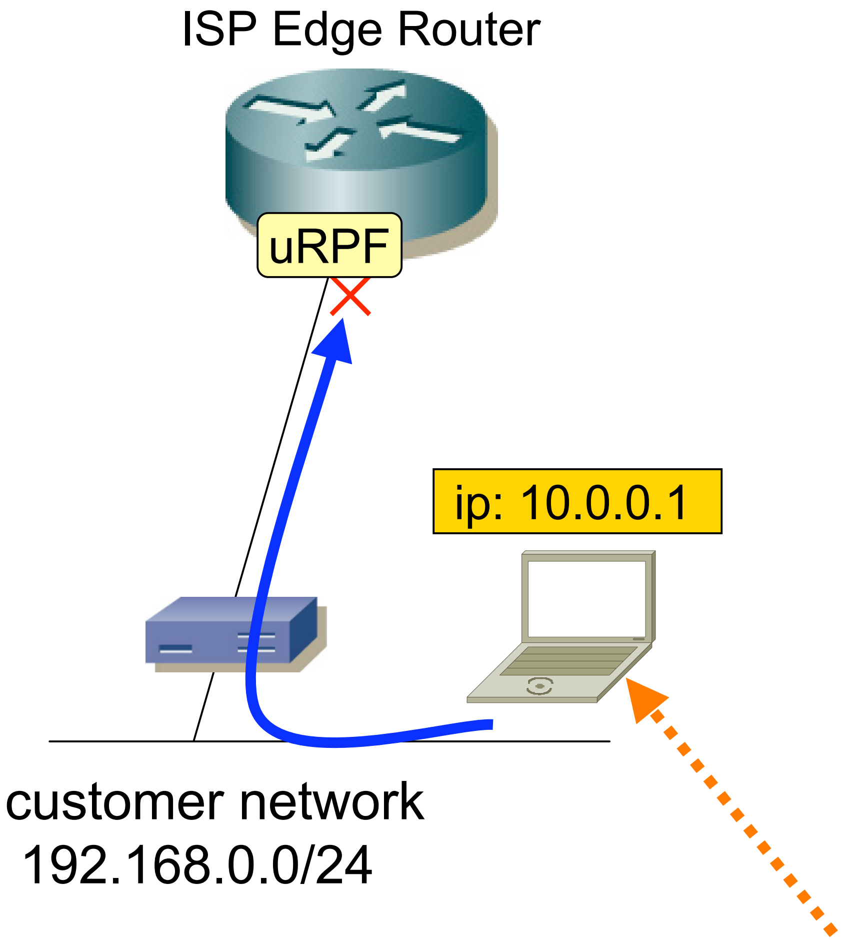
- if there is no filter, these packets keep looping until ttl expired....
- fix the routing!
- add null routes on the customer router

private space



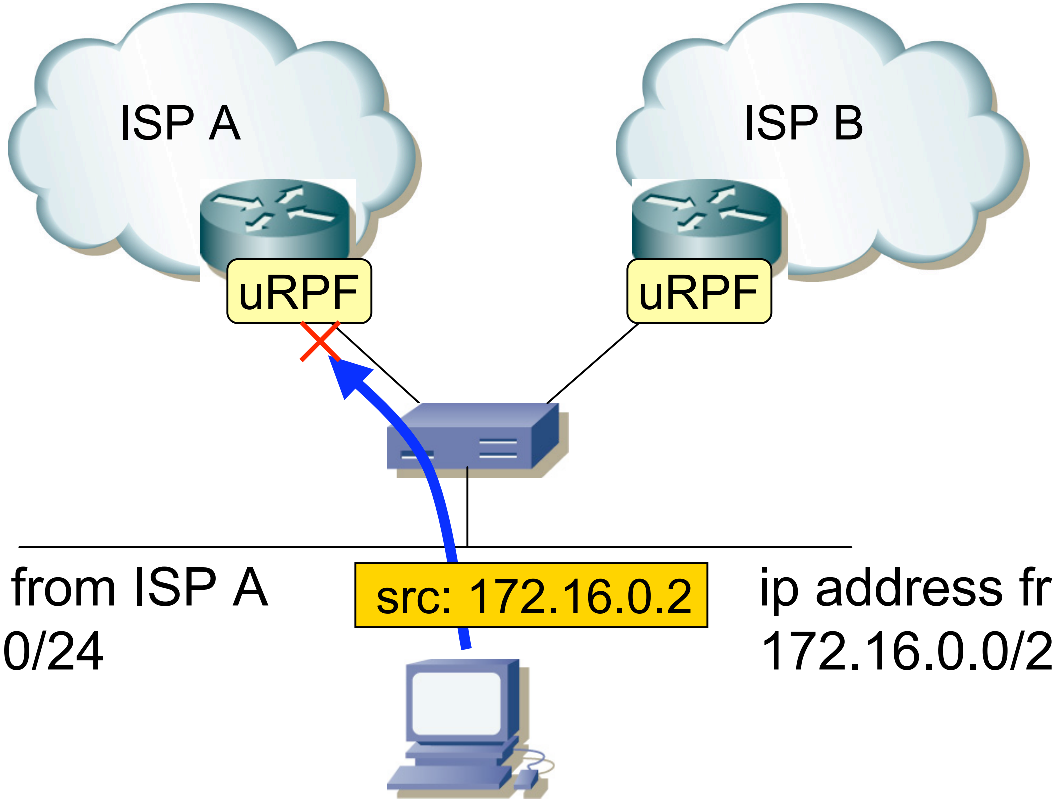
- usual case ☹
- bad implementation of NAT
- mis-configuration
  - router/firewall
  - network

wrong IP address



- mobile PC trying their old IP
- mis-configuration
  - typo
- just spoofing

multi-connected network



ISP A

ISP B

uRPF

uRPF

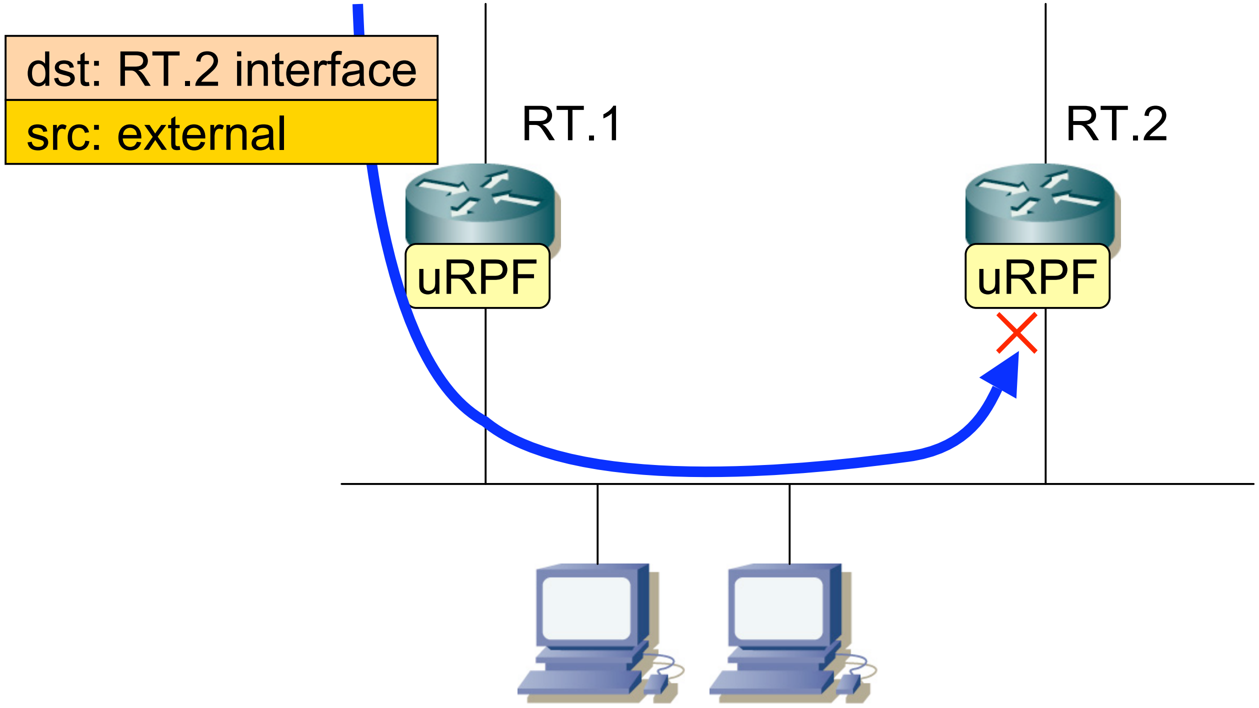
ip address from ISP A
192.168.0.0/24

src: 172.16.0.2

ip address from ISP B
172.16.0.0/24

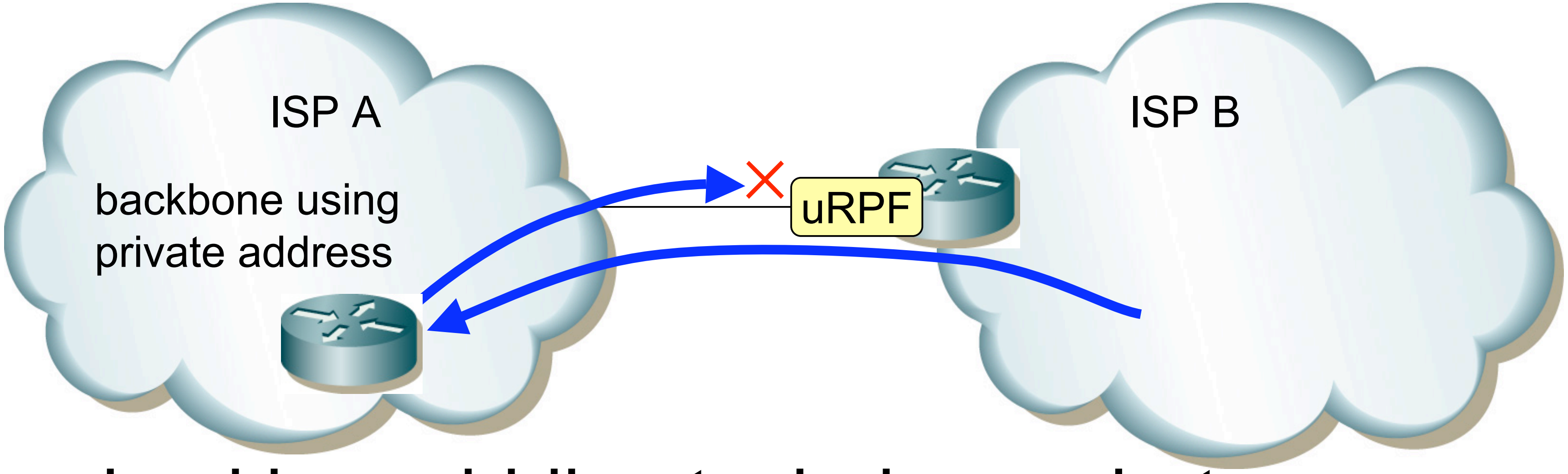
- PBR can fix this.

transit LAN



- packets to the router interface may filter

private/non-routed backbone



- backbone hiding technique... but
- icmp error messages will be filtered.
  - traceroute can’t show the ISP1’s network
  - this also breaks PMTUD

IIJ’s case

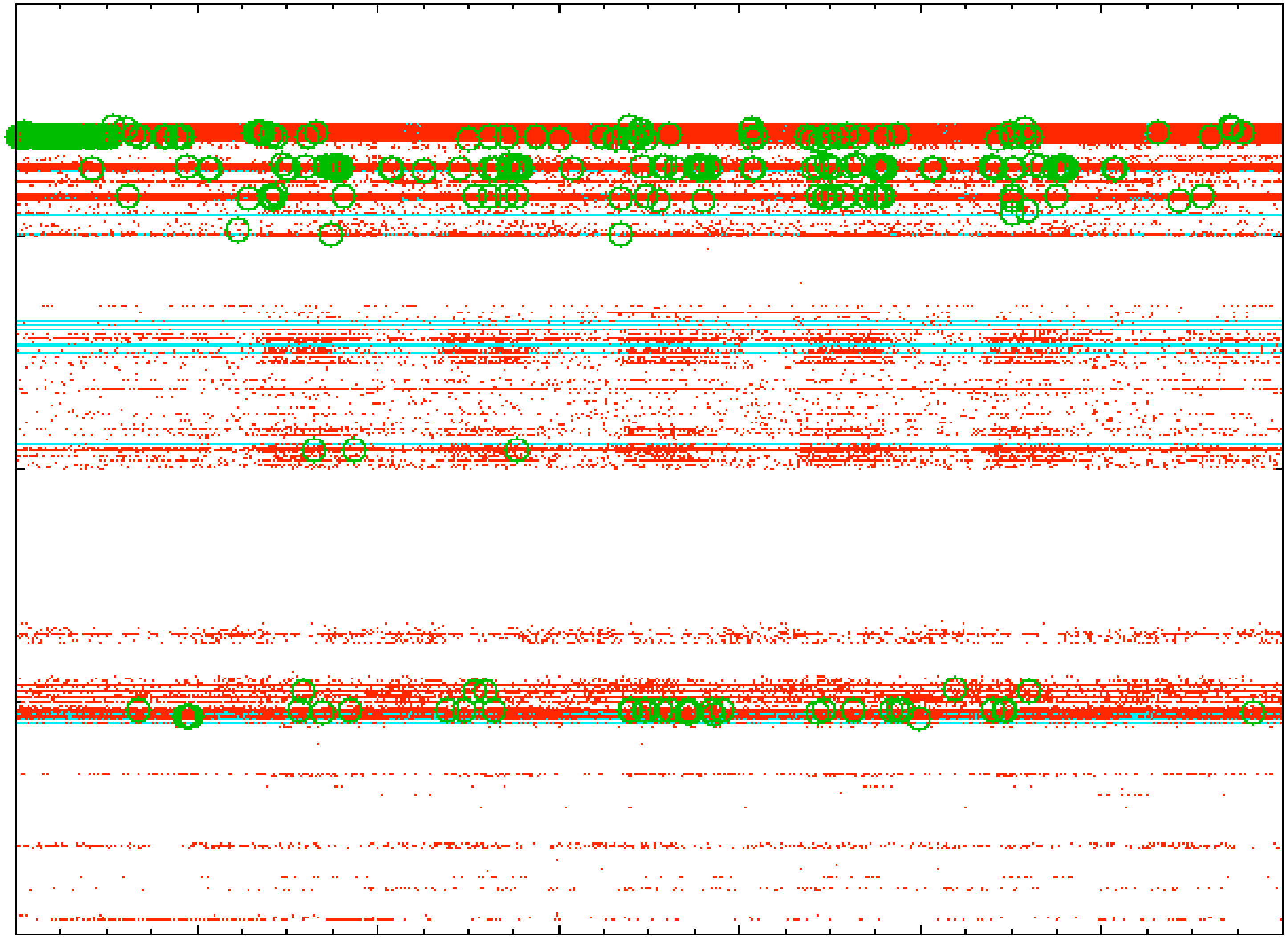
- discussion
- router capability
- policy
- problems

internal discussion

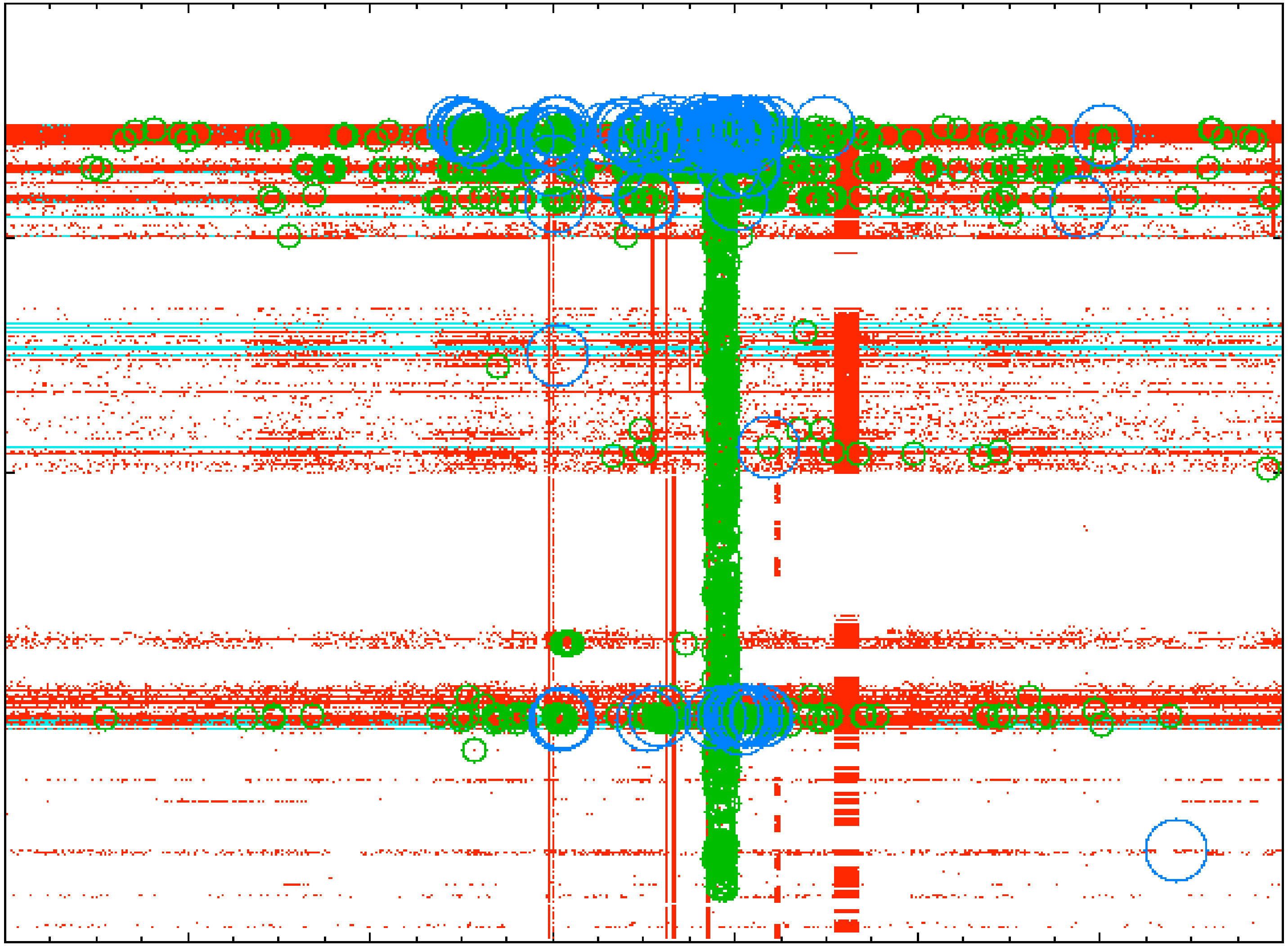
- Do we need anti-spoofing in our network?
  - We heard a rumor that attackers don’t use ip spoofing anymore in these days.

- Answer is YES.
  - ip spoofing is still used for attacks.
    - dns amplification attacks
  - preparation for new attacks using ip-spoofing

kubo graph #1



kubo graph #2



router uRPF capability #1

- Cisco
  - uRPF loose/strict mode

- Cisco 72xx, 75xx
  - software processing.... ☹
- Cisco sup2, sup720
  - hardware support for uRPF/ACL ☺
  - one uRPF mode per box ☹

router uRPF capability #2

- Cisco 12xxx GSR
  - depends on engine type of line card
  - E0,E1: software processing
  - E2: per physical interface, exclusion ACL
  - E3: loose mode only
  - microcode reload...

router uRPF capability #3

- Juniper T/M
  - works fine ☺
  - ‘feasible’ means ‘set of same length prefixes’

routing table

prefix	pref.
10.0.0.0/24	100
10.0.0.0/24	120

feasible

routing table

prefix
10.0.0.0/24
10.0.0.0/30

non-feasible

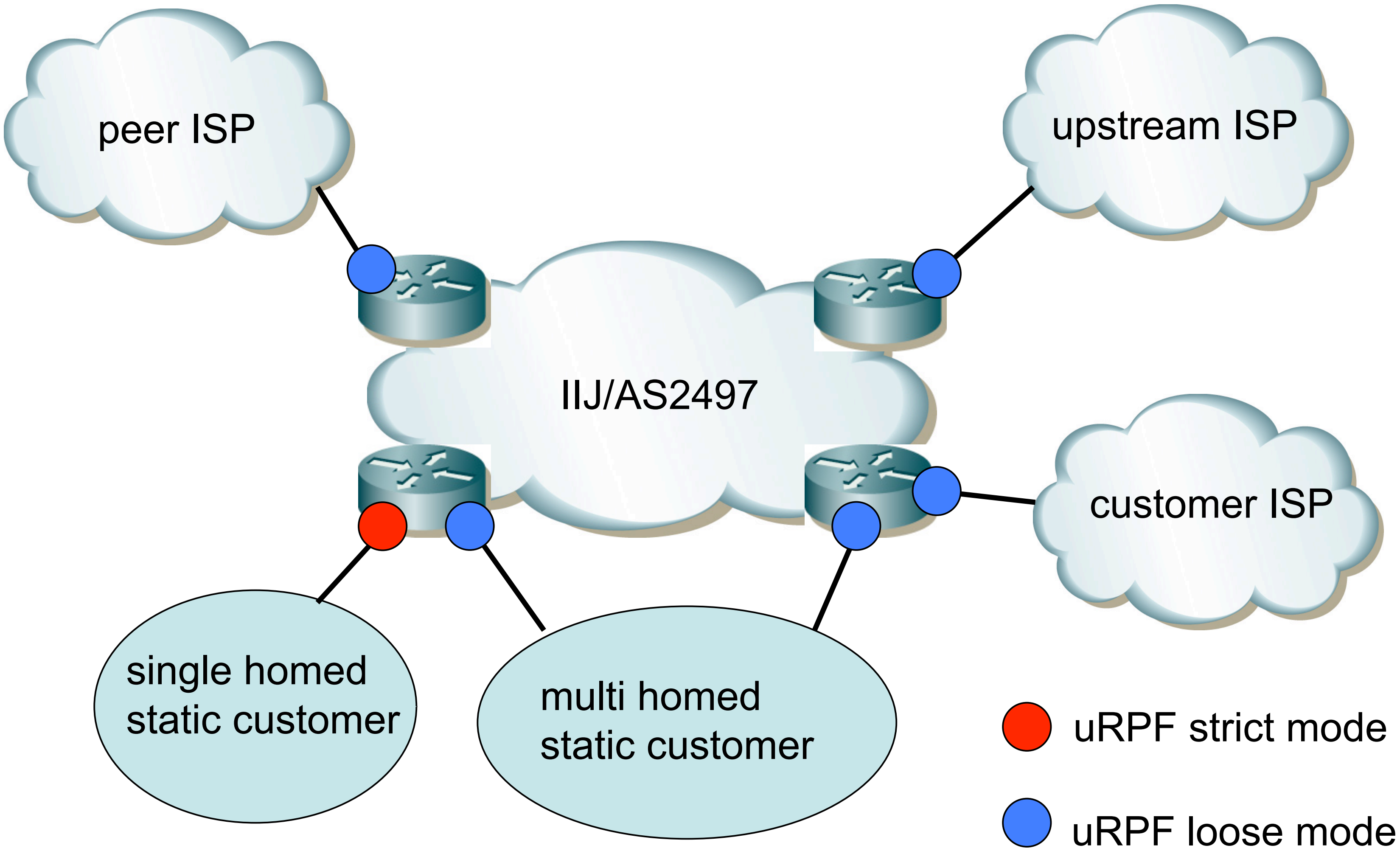
router uRPF capability

- Cisco
  - depends on box/linecard
  - uRPF strict/loose mode are supported
  - some boxes use software processing
    - additional 5~20% cpu load
- Juniper
  - works fine
  - need some hack to export cflowd data of discarded traffic

our initial choice

- single homed user
  - simple ☺
  - uRPF strict mode or ACL
- multihomed user
  - bgp customer(ISPs)
  - enterprise (need for redundancy)
  - uRPF loose mode
    - ・・・ something is better than nothing

IIJ’s policy


peer ISP
upstream ISP
IIJ/AS2497
customer ISP
single homed static customer
multi homed static customer
🔴 uRPF strict mode
🔵 uRPF loose mode

ACL and uRPF

- ACL
  - deterministic ☺
    - statically configured
  - maintenance of access-list ☹
- uRPF
  - easy to configure ☺
  - care about asymmetric routing ☹
    - strict mode is working well only for symmetric routing
    - loose mode can’t stop the ip reflected attack
    - there are few venders support of feasible mode

problems

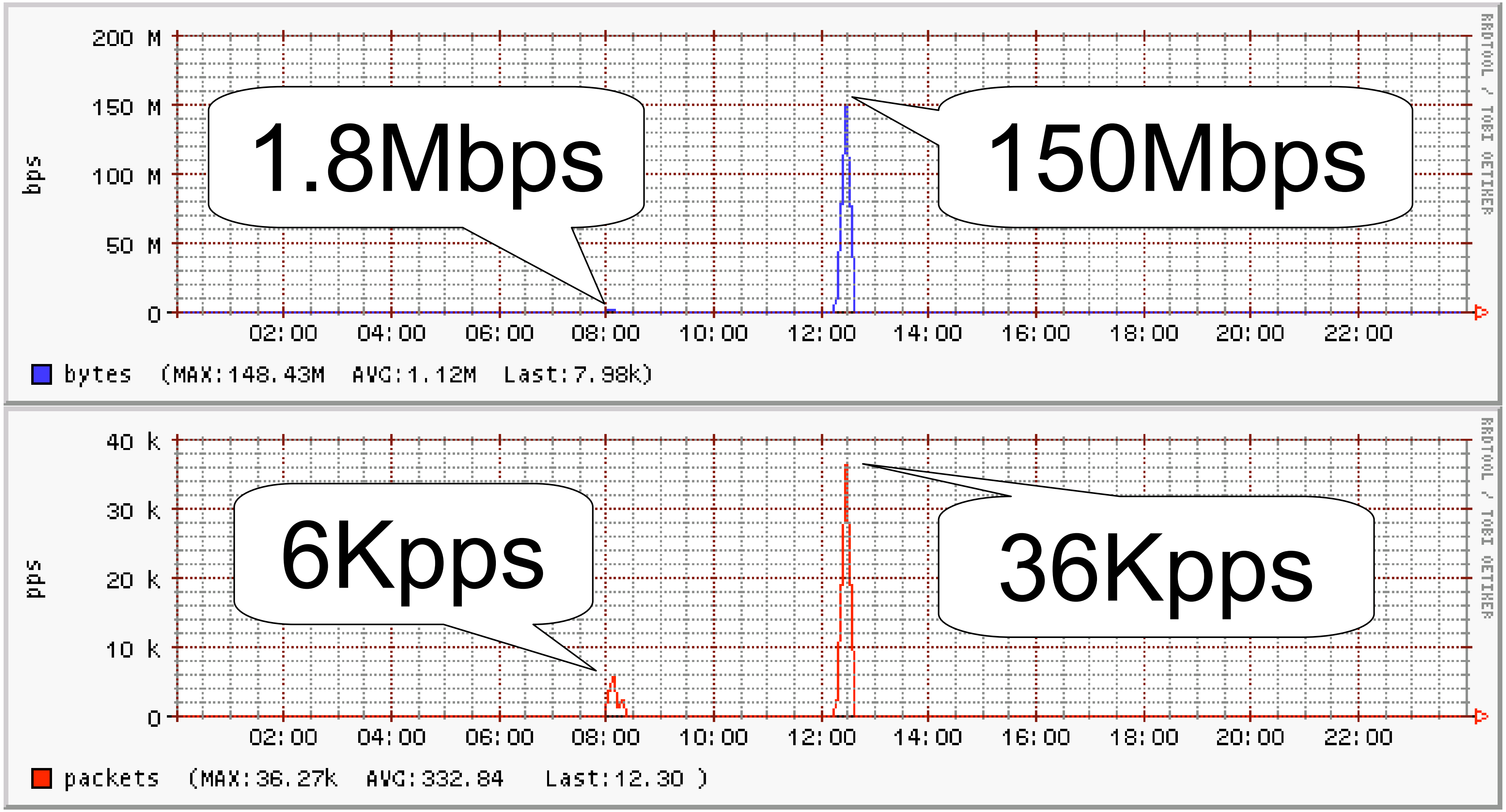
- uRPF/ACL works fine in most case. ☺
  - bug, device capability, performance...
- less confidence for uRPF
  - operations know uRPF, but never use it.
  - test it!
- unaware of Source Address Validation
  - why do we need this?

Why do we need?

- Source Address Validation do NOT protect your users from DoS/Attacks/Etc. directly.

- This reduce malicious activity.
  - sending ip spoofed packets from your network.
- If no networks allow ip spoofing, we can eliminate these kinds of attacks.

bogon traffic



1.8Mbps

150Mbps

6Kpps

36Kpps

please consider Source Address Validation in your network

END