

# Securing network infrastructure

Matsuzaki 'maz' Yoshinobu

<maz@iij.ad.jp>

# Our Goals

- Ensuring Network Availability
- Controlling Routing Policy
- Protecting Information
- Preventing Misuse
- Mitigating Attacks
- Responding to Incidents
- etc.

# Risks

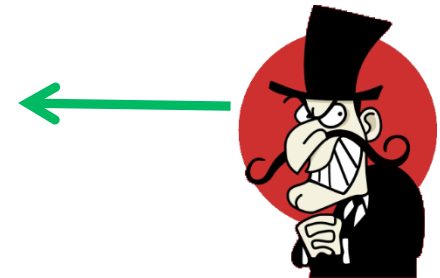
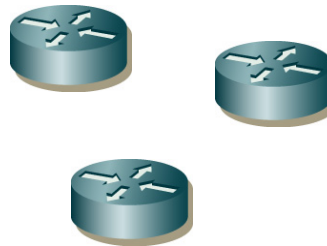
operations



remote access



- unauthorized access
- DoS
- route injection
- untraceable incident



attacker

# AAA server and remote access

- Authentication, Authorization, Accounting
  - tacacs, radius
- each operators has own login account
  - You can set privileges per tasks of the operator
- logging at AAA servers
  - where (device)
  - who (login account)
  - what (command)

# Remote Access to Devices

- in-band access
  - vty, snmp, ntp, etc...
  - IP reachability is required
  - useful for daily operations
- out-of-band access
  - serial console
  - workable without IP reachability
  - useful for restoration

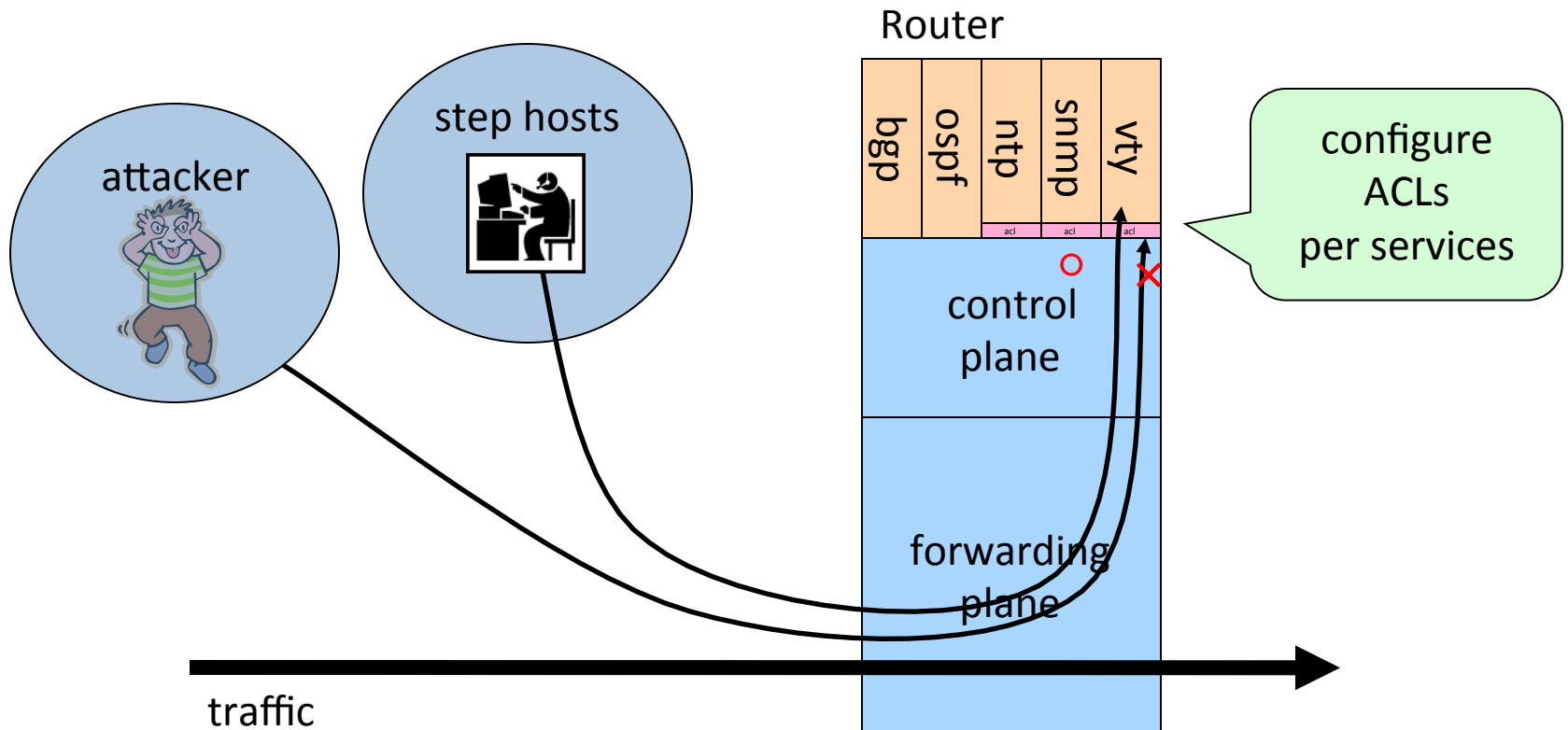
# Access Control for in-band access

- operations need to access remote devices
  - to manage the devices
- packet filtering on vty, snmp and etc
  - to protect devices from unauthorized access
  - allow access from trusted network only
    - source IP address based filtering

# step hosts

- are placed on a trusted network
- useful to enforce more restricted control
- each operations has own login account
- logging on step hosts
  - typescript of a VTY session
  - login/logout

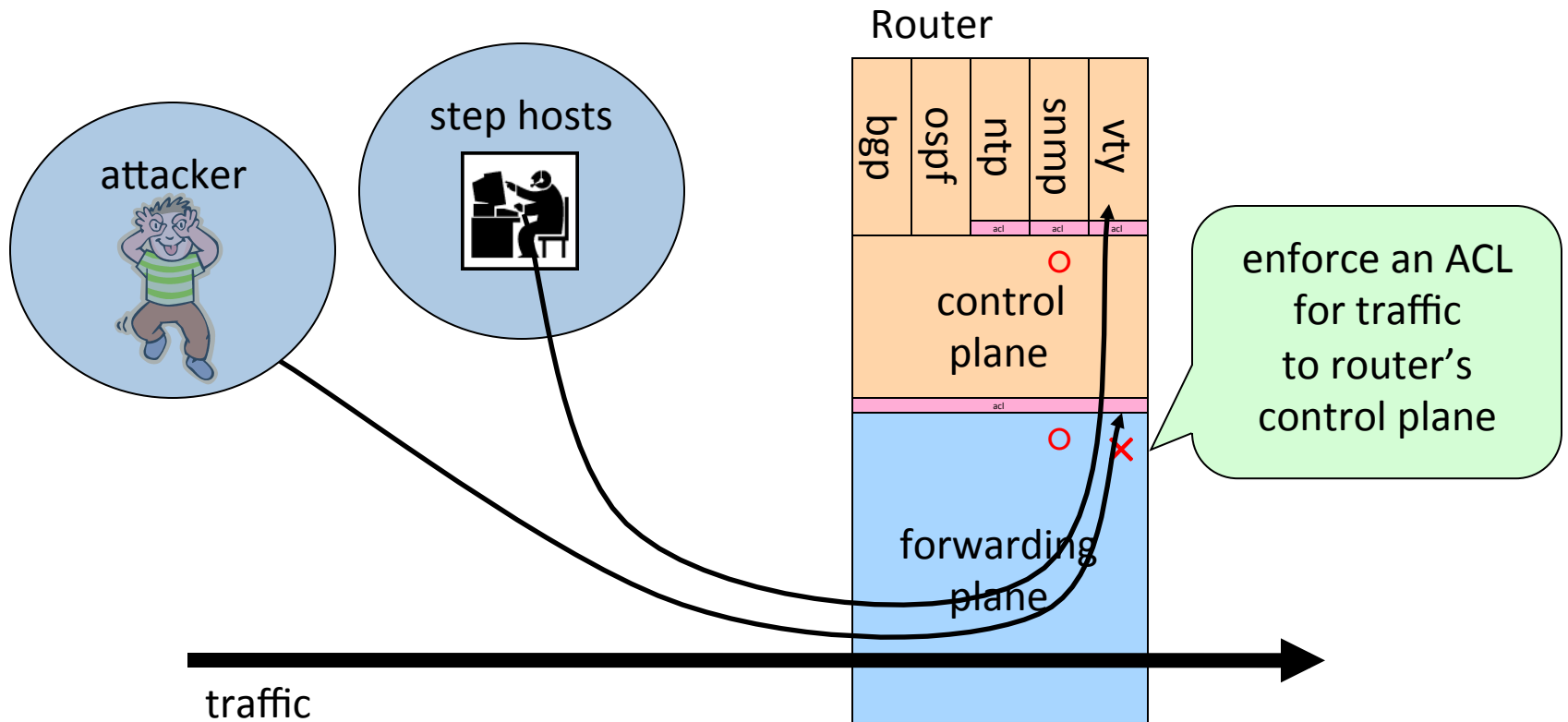
# access control per services





# Received/Router ACL (rACL)

access control against control plane

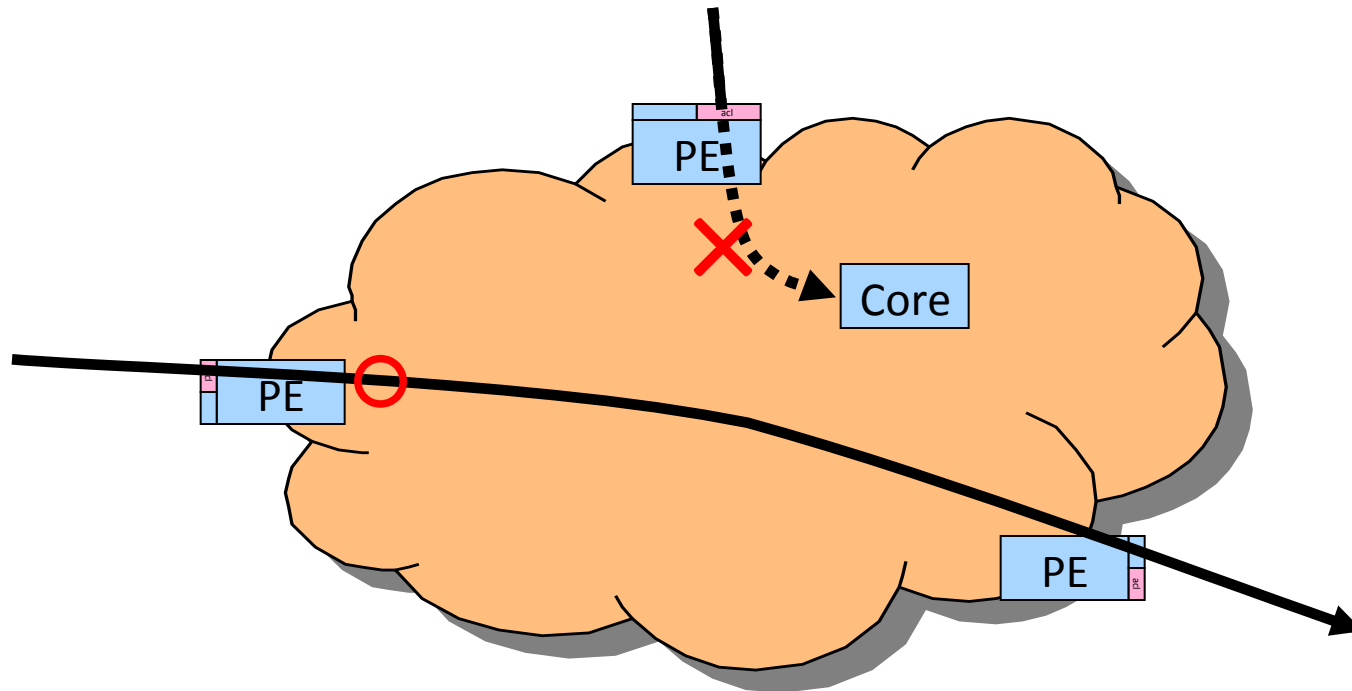


# infrastructure ACL

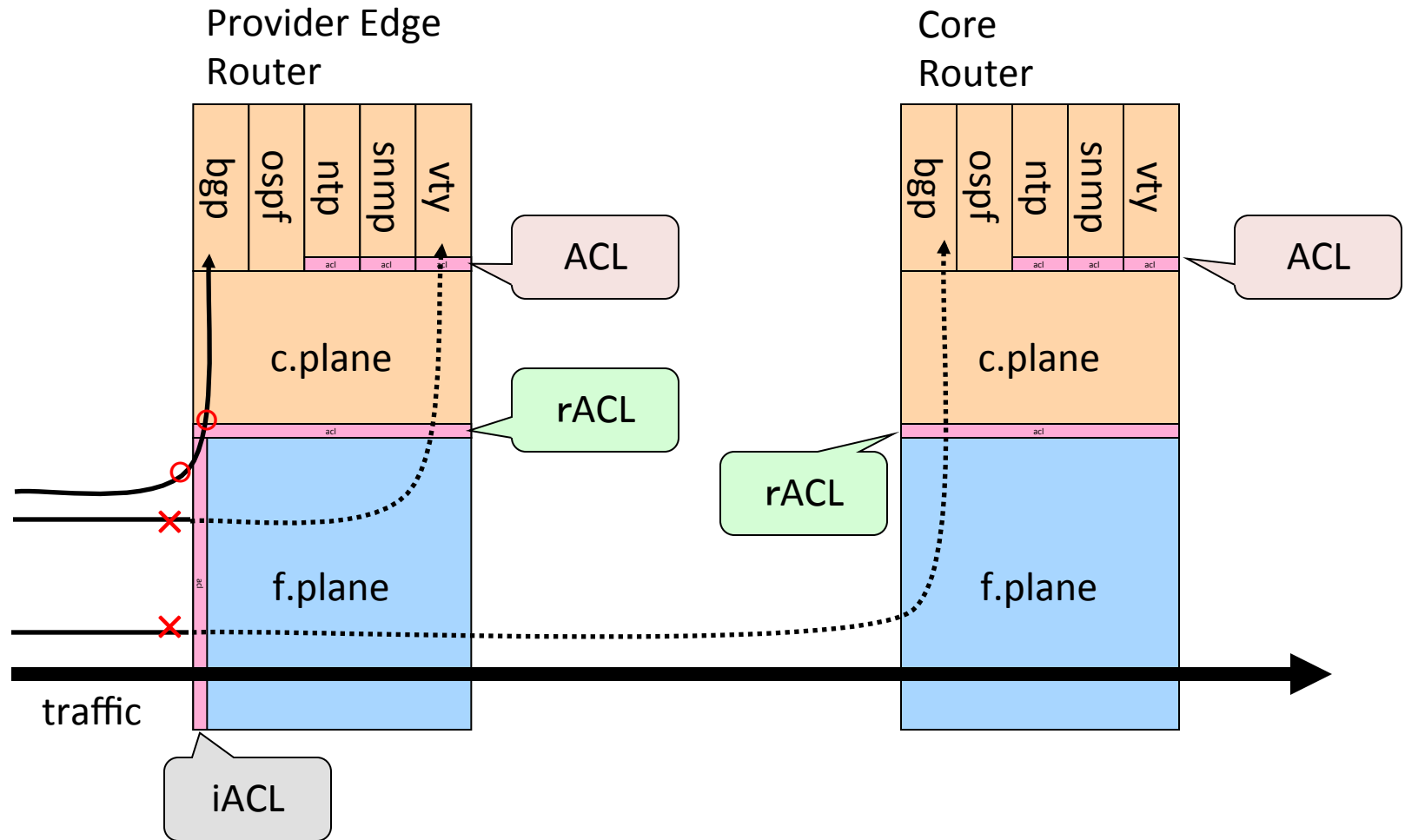
- to protect our management traffic
  - not too much
  - ping, traceroute to our devices should be workable
- deny packets from INFRA to INFRA on edge
  - INFRA: routers, step hosts and so on
    - these ip range should be stayed inside

# Infrastructure ACL (iACL)

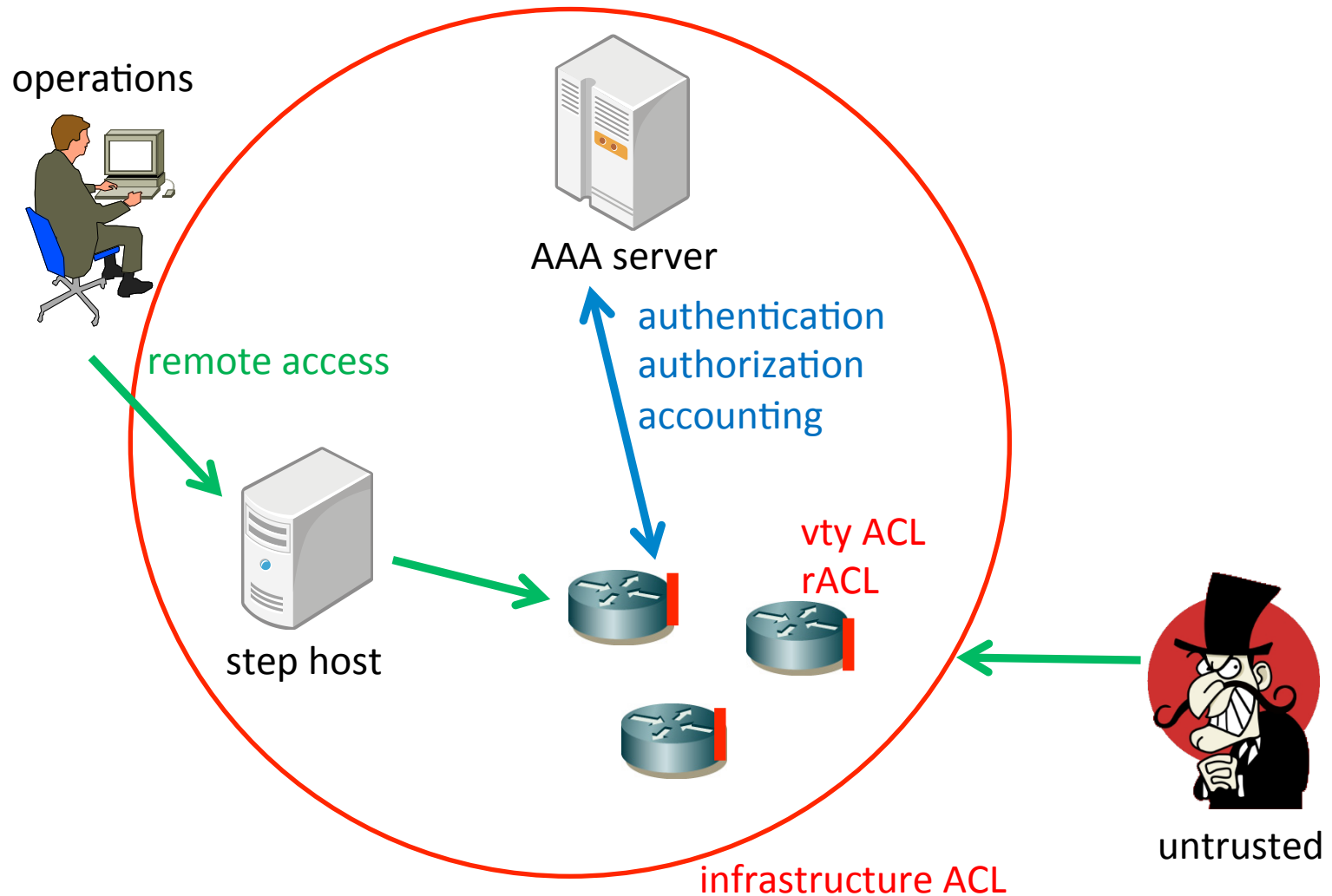
- enforce a policy on the network edge



# multiple ACLs to protect Devices



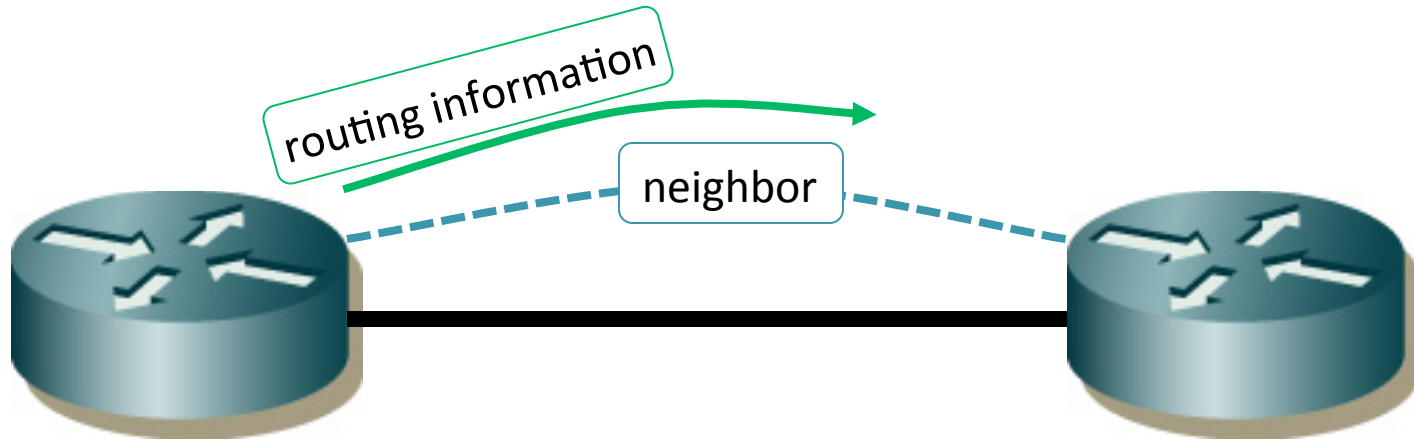
# protecting devices



# Protecting Routing

- To keep your network working
  - as you designed
  - as you configured
- Static Routing
  - mostly depends on design
- Dynamic Routing
  - possibility of remote attacks

# Routing Protocol



- Routers exchange routing information over a neighboring relationship.

# Threat Model for Routing

- Neighboring Relationship
  - Unexpected Neighboring
  - Shutdown by Someone else
  - Spoofed Neighbor
- Routing Information
  - Propagation of Wrong Information
  - Unintended Routing Policy
  - Hit a Hardware Limitation



# OSPF Neighbors

- Establishing a relationship among trusted neighbors only
- Disabled by default
  - Especially on a link to other parties (IX, customer)
    - to avoid unexpected neighbors
    - if you have to enable on these links, use 'passive' feature
  - Enabled where it is needed like backbone
- Authentication
  - MD5 authentication (OSPFv2, RFC2328)

# OSPF md5 configuration

cisco

```
interface <interface_name>  
ip ospf authentication message-digest  
ip ospf message-digest-key <keyid#> md5 <md5_key>
```

juniper

```
protocols ospf {  
  area <area#> {  
    interface <interface_name> {  
      authentication {  
        md5 <keyid#> key "<md5_key>";  
      }  
    }  
  }  
}
```

# BGP4 Neighbors

- Protecting TCP sessions
  - md5 authentication
- Peering with other parties
  - possibility of injection
  - needs more attention about routing information

# BGP md5 configuration

cisco

```
router bgp <as#>  
  neighbor <neighbor_ip> password <md5_key>
```

juniper

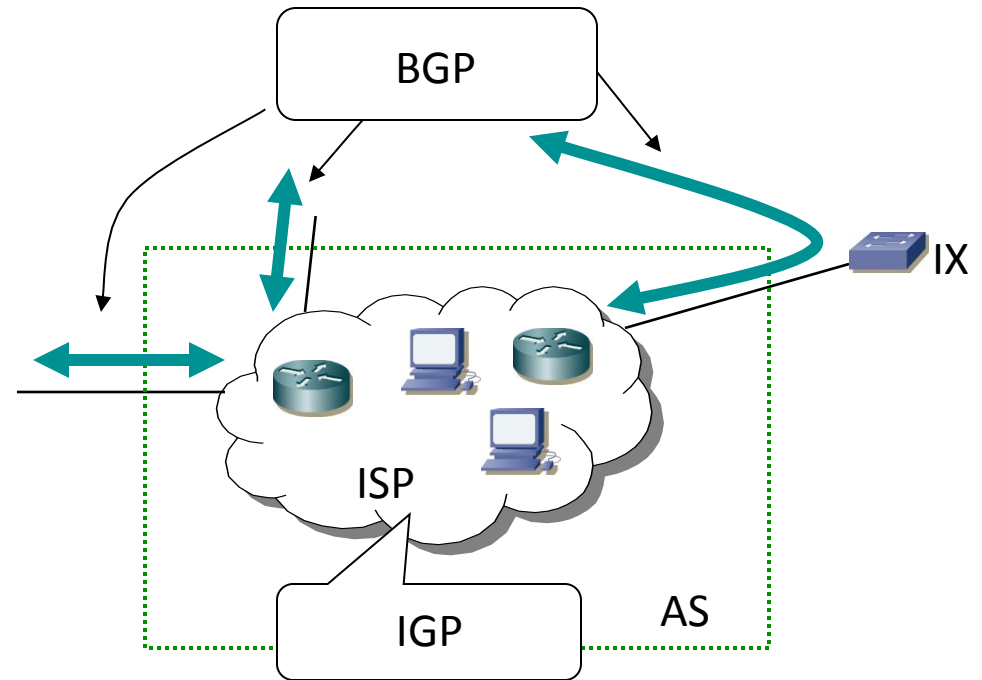
```
protocols bgp {  
  neighbor <neighbor_ip> {  
    authentication-key "<md5_key>";  
  }  
}
```

# Protecting routing information

- OSPF
  - mostly relies on neighboring
  - IGP should be used for internal purpose
    - should not be used to share routing information with your customers
- BGP
  - routing information is more problematic

# IGP and EGP

- IGP
  - OSPF, IS-IS, BGP
  - intra-AS
- EGP
  - BGP only
  - inter-AS



# BGP routing




- ISPs use BGP to carry routing information
  - full routes
  - customer routes
- need to peer with other parties
  - You know direct peering ASes
  - but not sure about ASes which are 2 or more AS hops away
  - You need to receive BGP announcements from such 'unsure' parties through peers

# critical routing information inside AS

- iBGP neighbor
  - usually loopback interface
  - /32 announcement by IGP
    - the most preferred
- BGP nexthop
  - typical BGP nexthop
    - IX segment
    - peering link
    - customer link
  - route filtering on eBGP sessions
    - needs care about more-specifics



# BGP UPDATE

- Prefixes + Path Attributes
- major attributes
  - AS Path 
  - local preference
  - MED 
  - nexthop
  - bgp community 
  - and so on

# exchanging routing information

- upstream
  - upstream announces full-route to us
  - we announce ourselves + customer
- peer
  - peer announces their selves + their customer
  - we announce ourselves + customer
- customer
  - customer announces their selves + their customer
  - we announce full-route

# BGP UPDATE from upstreams/peers

- risks
  - rogue announcement
    - default, own prefixes, private, linklocal, testnet
  - too many prefixes
- policy
  - accepts most routes up to /24
    - filter rogues
  - accepts basic routing control by AS Path
    - no MED, no BGP community
  - limit # of prefixes

# inbound route filter for upstreams/peers

- prefix filter
  - deny default (0.0.0.0/0)
  - deny private and other special prefixes
  - deny IJ prefixes
  - deny IX segments which IJ connects
  - accept prefixes up to /24
- resetting attribute
  - BGP community and MED
- prefix limitation
  - # of prefixes

# special-use prefixes [RFC5735]

- private
  - 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16
- shared
  - 100.64.0.0/10
- loopback
  - 127.0.0.0/8
- linklocal
  - 169.254.0.0/16
- testnet and benchmark
  - 192.0.2.0/24, 198.51.100.0/24, 203.0.113.0/24, 198.18.0.0/15
- IETF protocol assignments
  - 192.0.0.0/24
- Multicast
  - 224.0.0.0/4

# BGP UPDATE from customers

- risks
  - mis-announcement
- policy
  - accepts exact prefixes and AS Path which the customer asked for transit
  - accepts routing control
    - AS Path, MED and bgp community

# inbound route filtering for customers

- prefix filter
  - accept exact prefixes which the customer asked for transit
- AS Path filter
  - accept AS Path which the customer asked for transit
- resetting attribute
  - BGP community
    - overwrite communities which the network uses internally

# updating the route filter for customers

- customers ask you to update filters
  - prefixes and AS Path
- You should perform sanity check
  - registration check with IR(APNIC) and IRR
    - to avoid unauthorized announcement
  - aggregation check
    - to avoid unnecessary de-aggregation
    - ask the customer to set NO-EXPORT to localize the de-aggregation prefixes



# whois check

```
$ whois -h whois.nic.ad.jp '210.130.0.0/16 /e'
```

[ JPNIC database provides information regarding IP address and ASN. Its use ]  
[ is restricted to network administration purposes. For further information, ]  
[ use 'whois -h whois.nic.ad.jp help'. To only display English output, ]  
[ add '/e' at the end of command, e.g. 'whois -h whois.nic.ad.jp xxx/e'. ]

## Network Information:

[Network Number]	210.130.0.0/16
[Network Name]	
[Organization]	Internet Initiative Japan Inc.
[Administrative Contact]	JP00010080
[Technical Contact]	JP00010080
[Abuse]	abuse-contact@iij.ad.jp
[Allocated Date]	1996/10/03
[Last Update]	2007/06/28 10:26:08(JST)

## Less Specific Info.

-----

No match!!

## More Specific Info.

-----

Too many matches. Narrower expression, please.

```
$ whois -h whois.nic.ad.jp 'JP00010080 /e'
```

[ JPNIC database provides information regarding IP address and ASN. Its use ]  
[ is restricted to network administration purposes. For further information, ]  
[ use 'whois -h whois.nic.ad.jp help'. To only display English output, ]  
[ add '/e' at the end of command, e.g. 'whois -h whois.nic.ad.jp xxx/e'. ]

## Group Contact Information:

[Group Handle]	JP00010080
[Group Name]	IP Address Contact
[E-Mail]	nic-sec@iij.ad.jp
[Organization]	Internet Initiative Japan Inc.
[Division]	
[TEL]	03-5205-6500
[FAX]	
[Last Update]	2005/08/23 14:02:22(JST)

# check authority before announcing it

- One ISP received allocation from JPNIC in 2006
  - before using it, they realized parts of the prefix was announced by ISP-A ☹️
- This story was shared at JANOG meeting
  - I contacted the ISP-A to stop the announcement, and took 4 hours to fix. 😊
    - The ISP-A announced it, because one customer asked to route it through their AS
    - probably the ISP-A didn't perform sanity-test properly

# BGP announcement

- policy
  - Prefixes should be aggregated as possible
  - avoid any rogue announcement
- to upstreams/peers
  - its own and customers' prefixes
    - distinguished by BGP community
- to customers
  - full routes

# Outbound route filtering

- prefix filter
  - deny private and other special prefixes
  - deny unnecessary more-specifics
  - deny too specifics (/25 or longer)
  - permit any
- remove private AS from AS Path
  - remove-private-as

# BGP routing policy

- keep your policy simple
  - less trouble
- You should expect unexpected traffic flow change
  - one traffic engineering can break other traffic engineering
  - peering might help to get more stable traffic pattern

# config audit

- configuration files are periodically gathered
  - by in-house automated tool
- sanity check
  - filtering rules
  - routing configuration
  - and so on

# monitoring

- what's happened in the past
- syslog
  - to record messages from devices/software
- snmp
  - to monitor resources
- netflow
  - to monitor packet flows

# syslog messages



- Nov 9 15:19:14.390 UTC: config[65775]:  
%MGBL-SYS-5-CONFIG\_I : Configured from  
console by maz on vty0  
(2001:db8:120:100:e1dd:97f3:fd98:a51f)
- Nov 12 13:53:38 maz sudo: maz : user NOT  
in sudoers ; TTY=pts/3 ; PWD=/home/maz ;  
USER=root ; COMMAND=/bin/bash

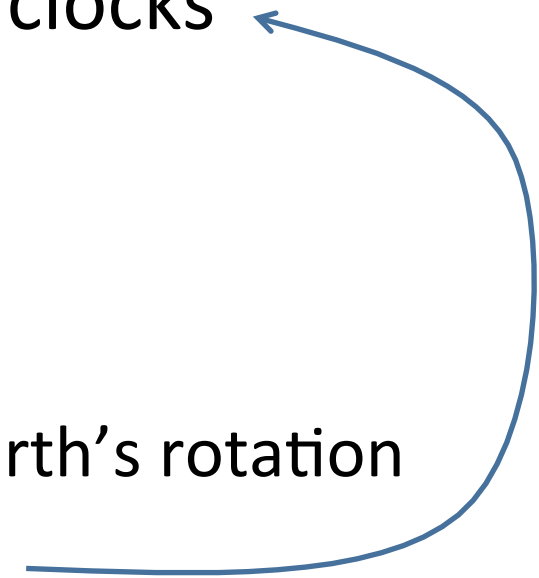




# synced timestamp

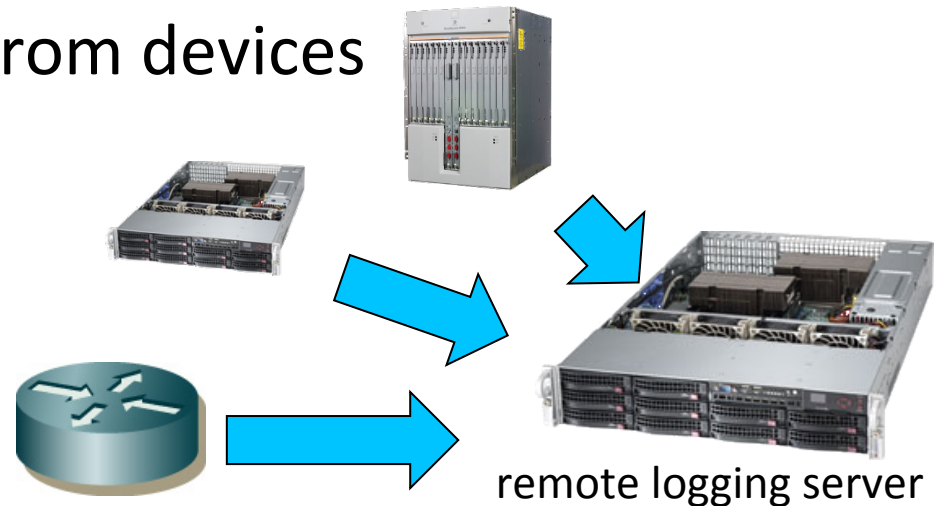
- makes log messages useful
  - to compare incidents among devices
  - to compare time-related events
- Use ntp to sync clocks
  - choose a proper clock source
    - national ntp server
    - stable clocks
      - ATOM, GPS

# clock = oscillation + counter

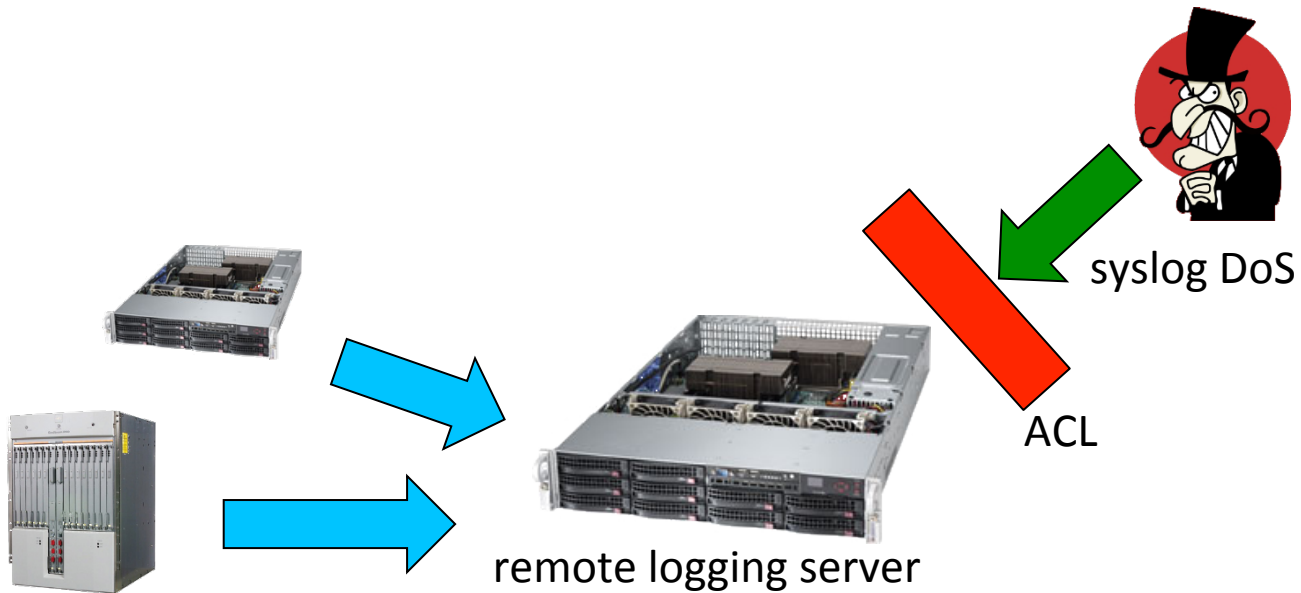
- TAI = weighted average of atom clocks
    - TAI: International Atomic Time
  - UTC = TAI + leap seconds
    - UTC: Coordinated Universal Time
    - leap seconds: to adjust clock to Earth's rotation
  - atom clocks are adjusted to TAI
  - localtime = UTC + timezone (+ summer time)
- 

# remote logging

- log messages could be modified/deleted
  - if the system is compromised
  - limited memory buffered log messages
- remote logging server
  - receive log messages from devices
  - syslog-ng
  - enough storage there



# protecting syslog

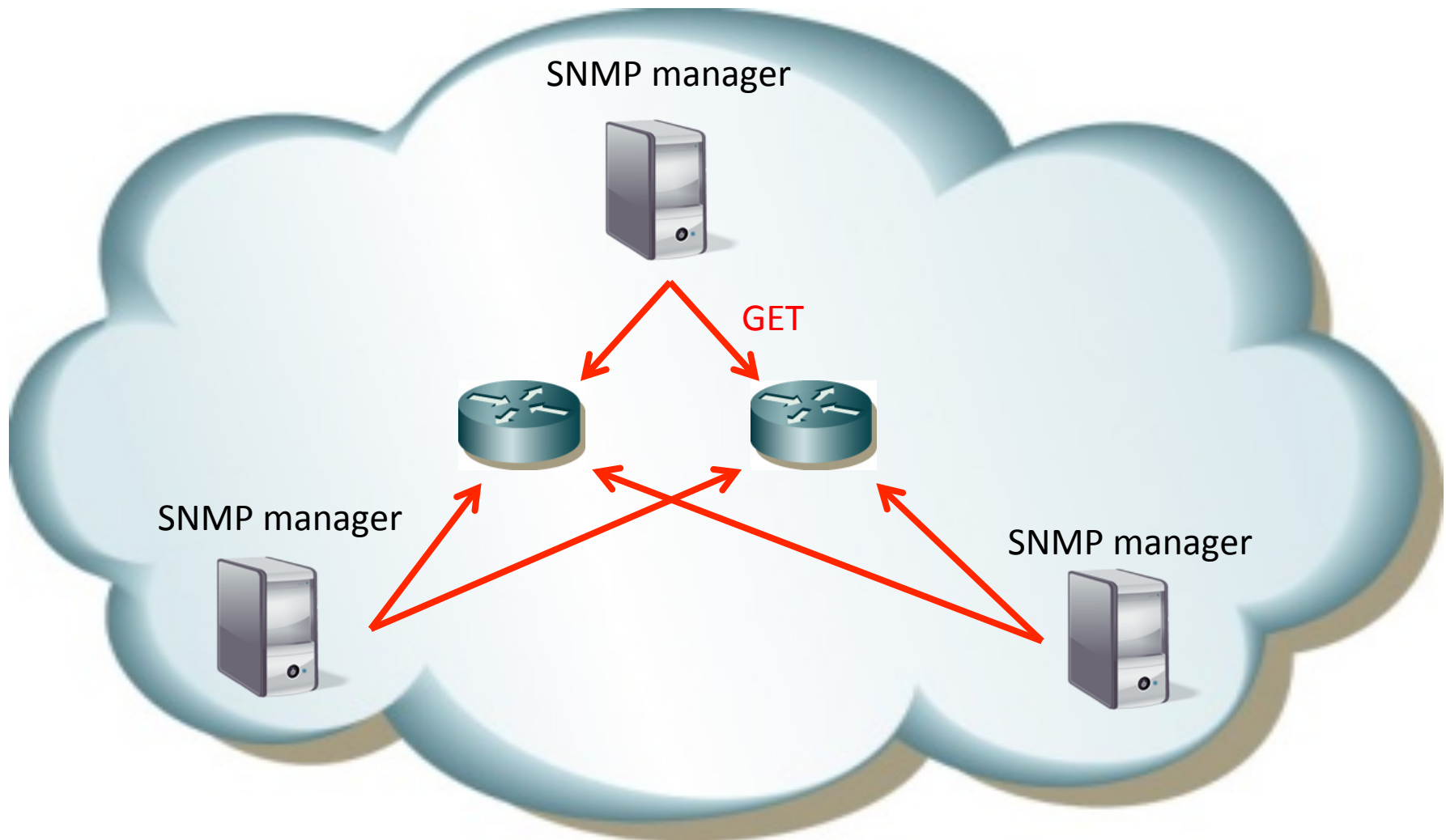


ensure the correctness of log entries

# snmp

- can read/write information and send a trap
  - use version 3, and set password
  - prevent 'write' function, or just disable it on agents
  - put ACL to prevent unauthorized access
- require a little disk space on snmp manager
  - useful to check **long-term trend**

# snmp monitoring system



# snmp MIB

- Management information base
  - MIB-2, IF-MIB, vender-specific MIB
  - you can get information if an agent supports the MIB you want
- you can specify the information by OIDs
  - ifHCInOctets = .1.3.6.1.2.1.31.1.1.1.6
  - ifHCOctets = .1.3.6.1.2.1.31.1.1.1.10

# snmp counters

- frequency of updating counters
  - depends on agents (0-30sec)
  - 5min is widely used as snmp polling time
- counter overflow
  - 32bit counters(ifIn/OutOctets) could wrap in 5.7min at 100Mbps
  - consider 64bit counters(ifHCInOctets) for 1Gbps or more interfaces



# useful information via SNMP MIBs

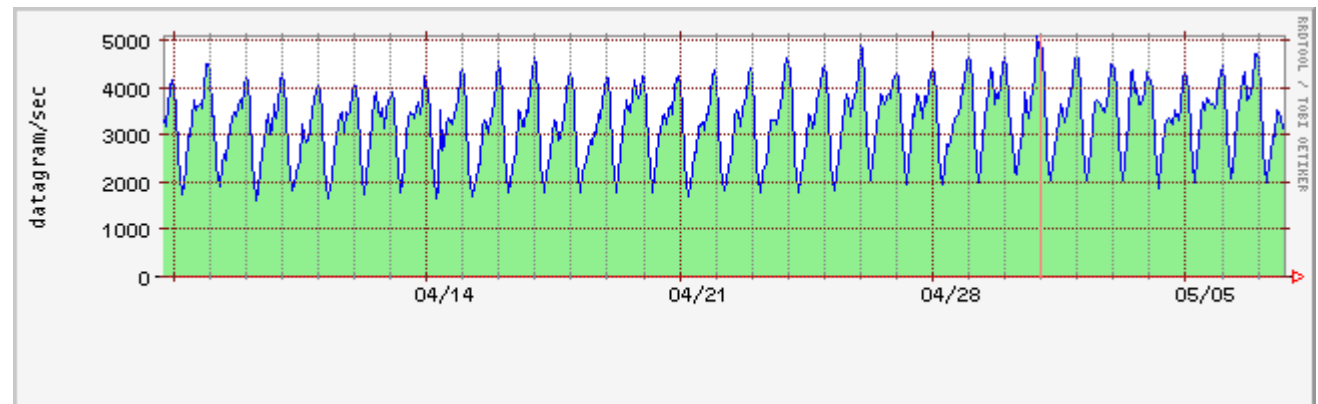
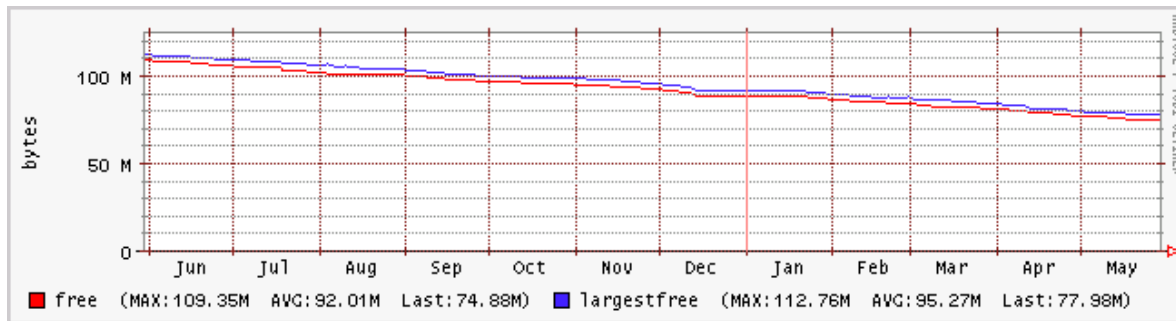
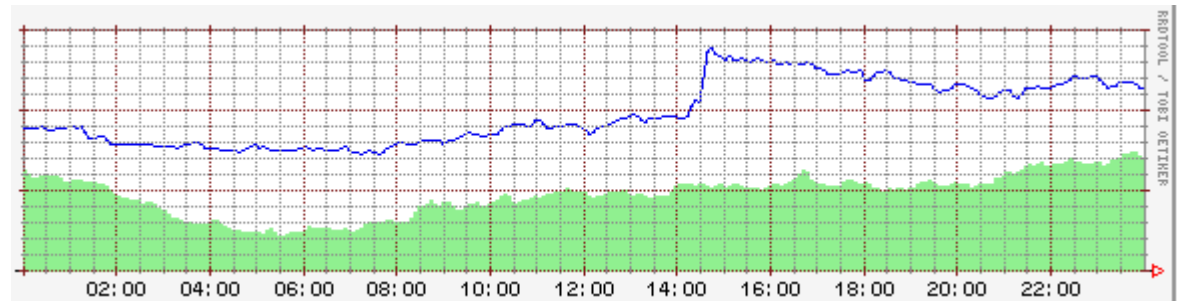
- interface
  - bytes, packets, errors
- system
  - cpu load
  - memory usage
  - temperature
  - icmp, udp
  - ntp

# snmp use case

- usage monitoring
  - bandwidth and traffic volume
- visualize
  - stackable graph
    - useful for multiple links between POPs
  - grouping
    - international links
    - IX

# visualize

- RRDtools



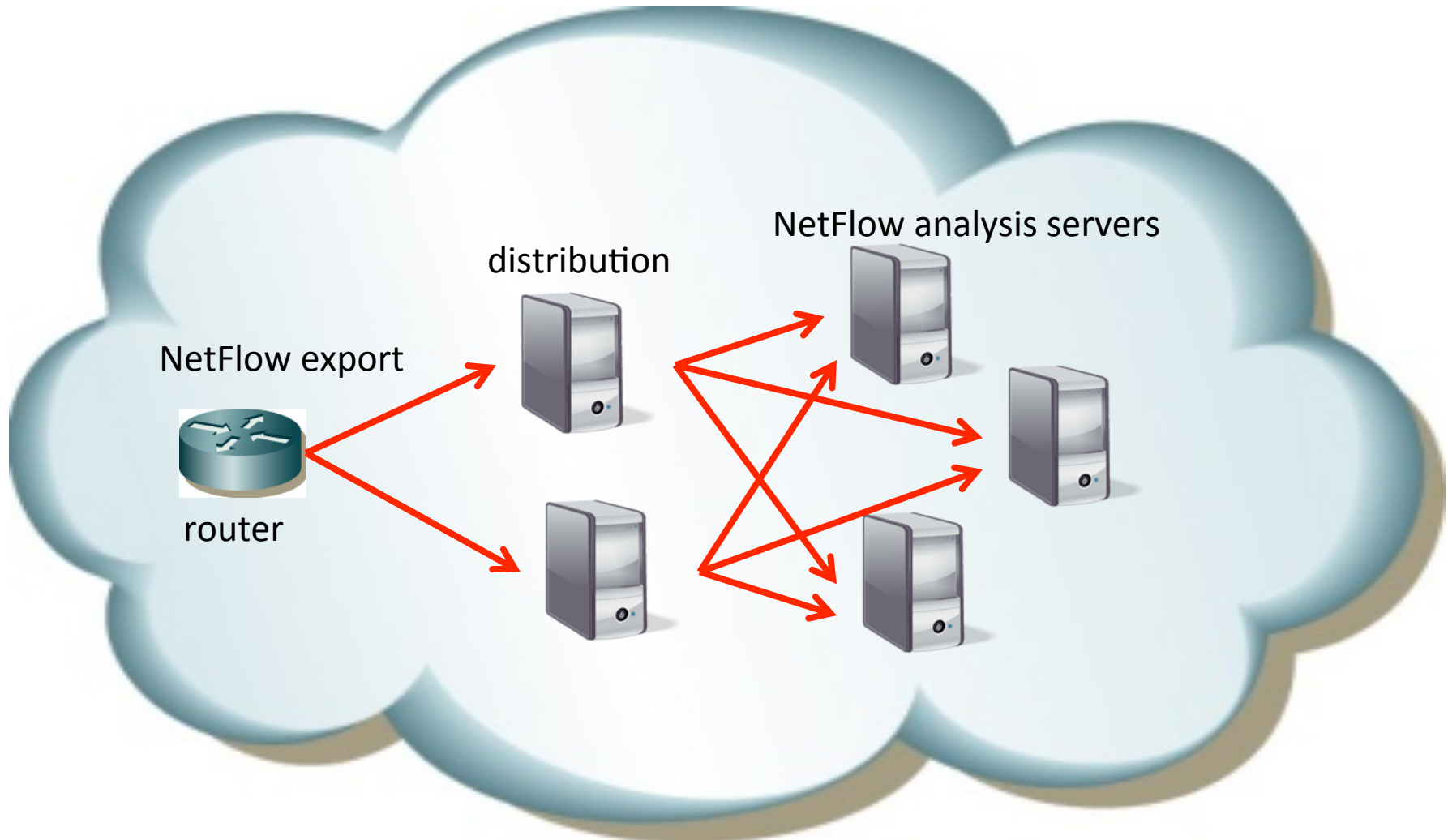
# netflow

- to monitor flow information
  - packet header
  - most routers support it
- require more storage
  - even with sampling, still need to expect huge data
  - not for long term monitoring
- useful for **analysis** and **anomaly detection**

# netflow and sampling

- sampled netflow is widely used
  - just to know trend
  - to reduce data
- margin of error
  - sampled netflow and actual traffic
  - depends on routers
  - worst case: 20%
- IJ uses magic number as sampling rate
  - $1/16382$

# netflow monitoring system

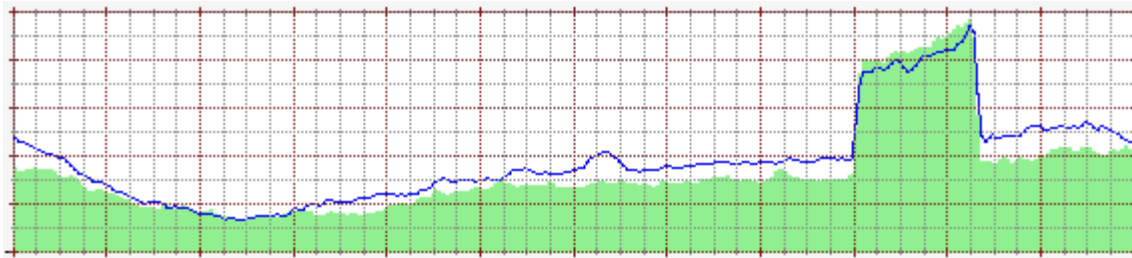


# netflow analysis

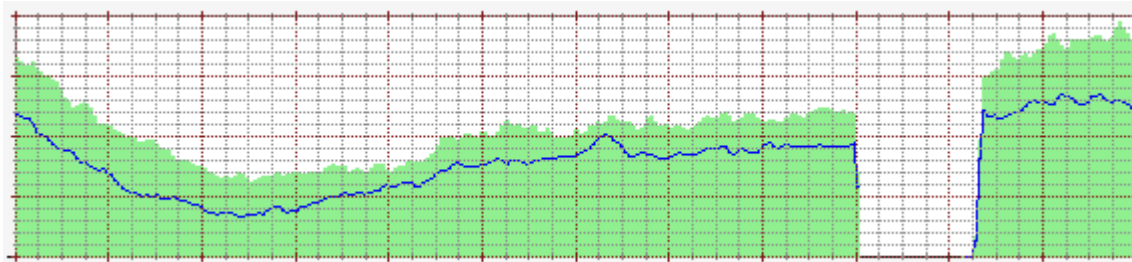
- combination of parameters
  - AS, IP address, protocol, port number
  - too many patterns to pre-generate every graphs
- Graphs
  - pre-defined graphs
  - dynamic graph system

# case 1: bps

- traffic was suddenly doubled on a link

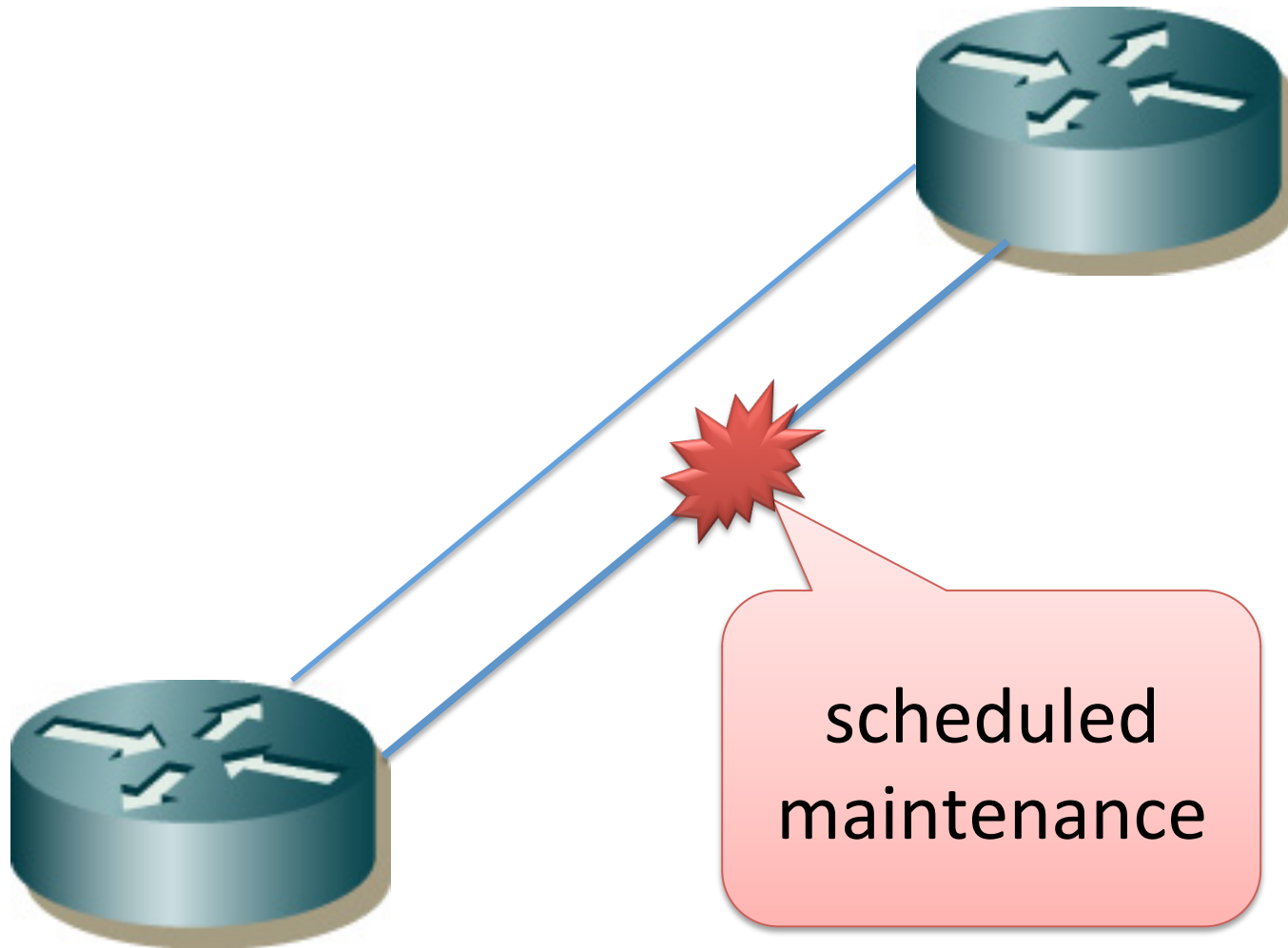


- also found a missing traffic

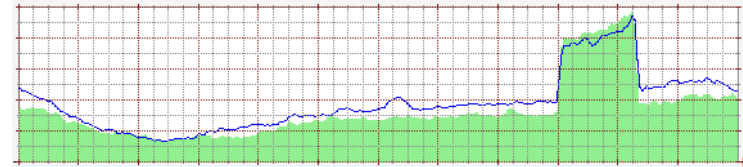
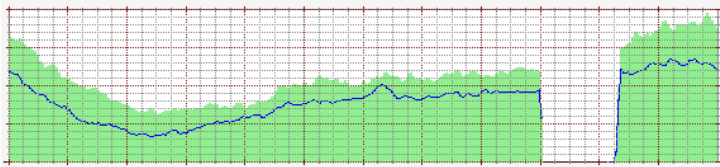




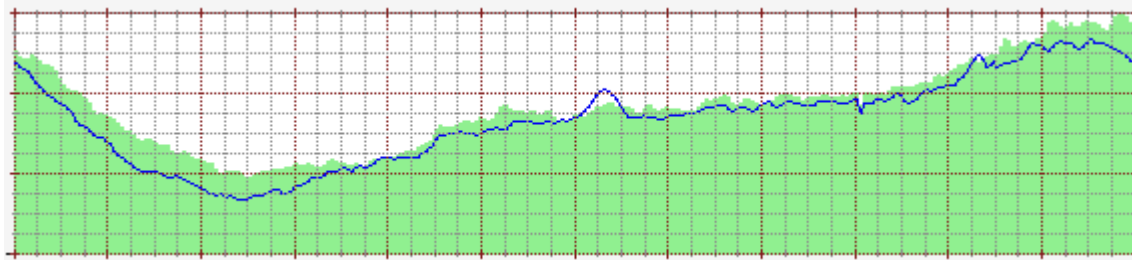
# case 1: 2 links between routers



# case 1: total traffic: bps

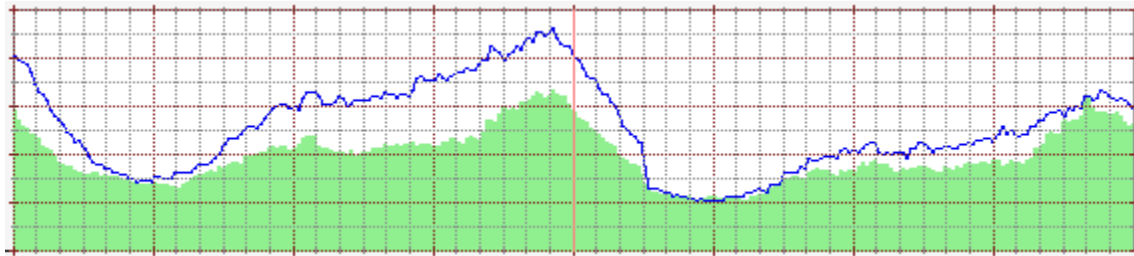


merge

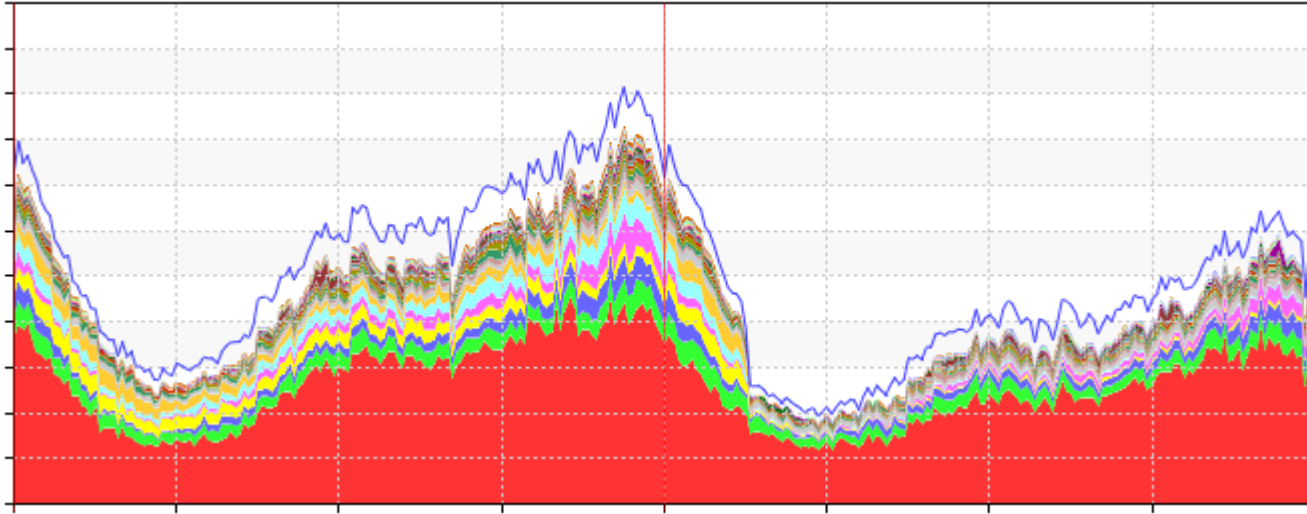


## case 2: bps

- traffic decreased
- There is no routing change in the network

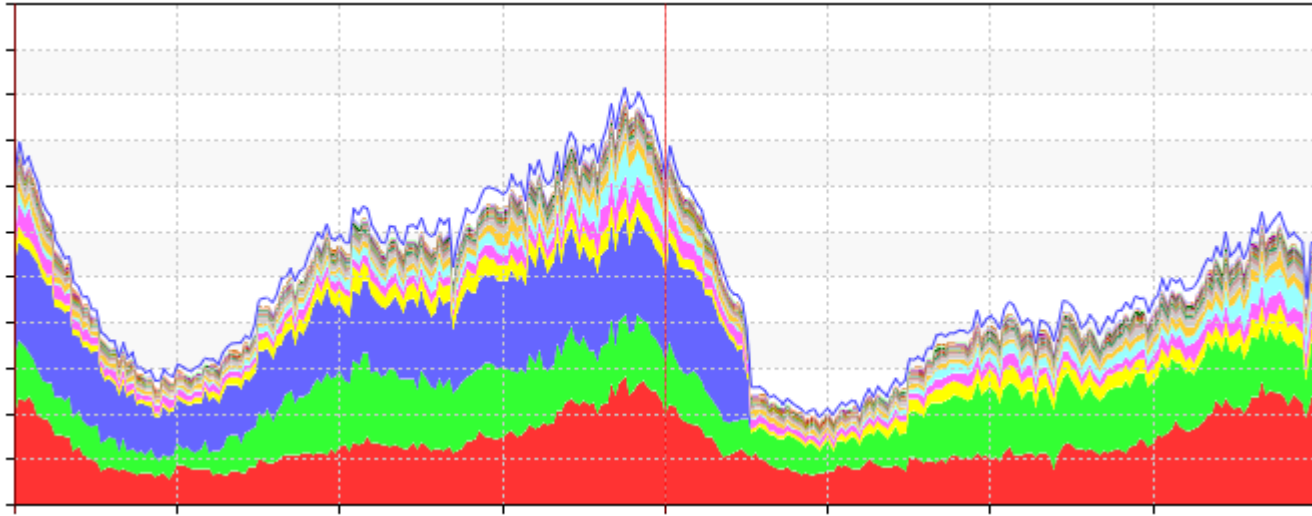


## case 2: netflow graph(dst AS)



- the dst AS based graph shows
  - missing traffic to several ASes
  - traffic to the other ASes also a bit decreased

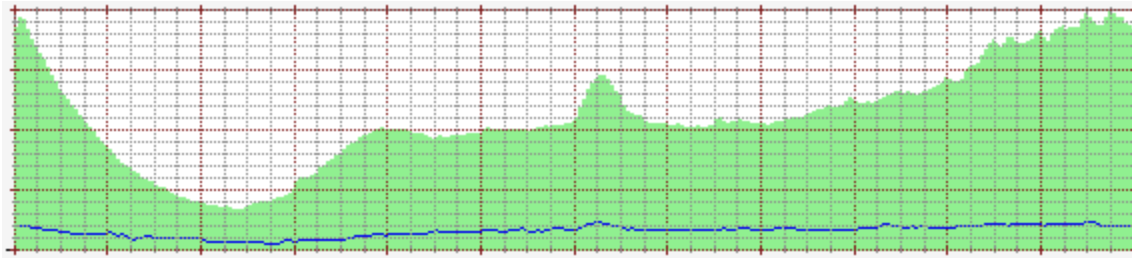
## case 2: netflow graph(src AS)



- traffic from a particular AS(blue) was gone
- probably something was happened on the AS(blue)
  - trouble or route change

# case 3: bps

- traffic looks stable



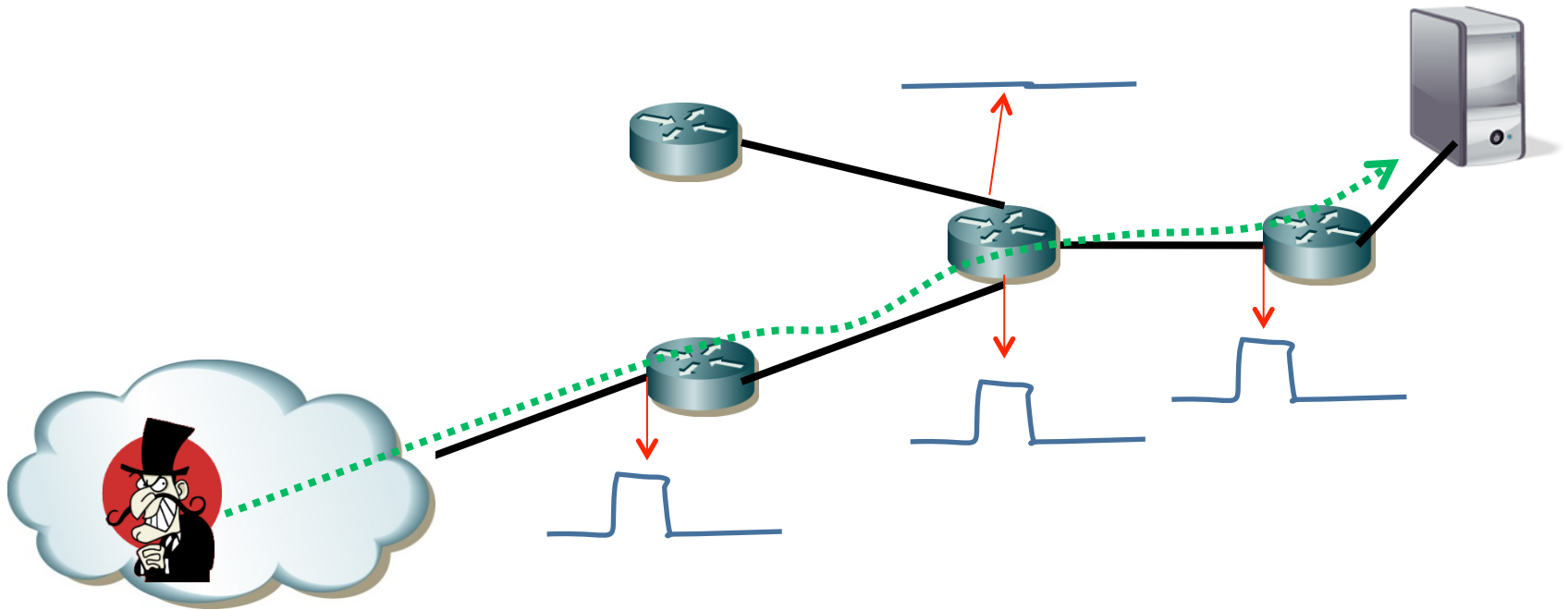
## case 3: pps

- pps(packets/sec) graph shows something anomaly



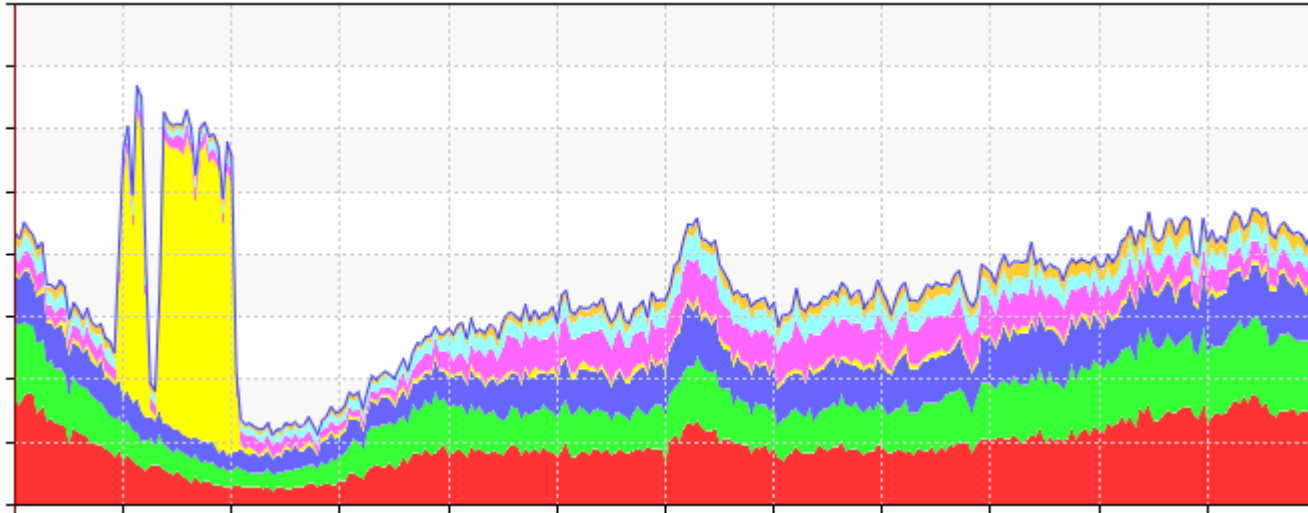
# traceback by a shape

- if the traffic pattern is enough characteristic, you can traceback to the inbound interface



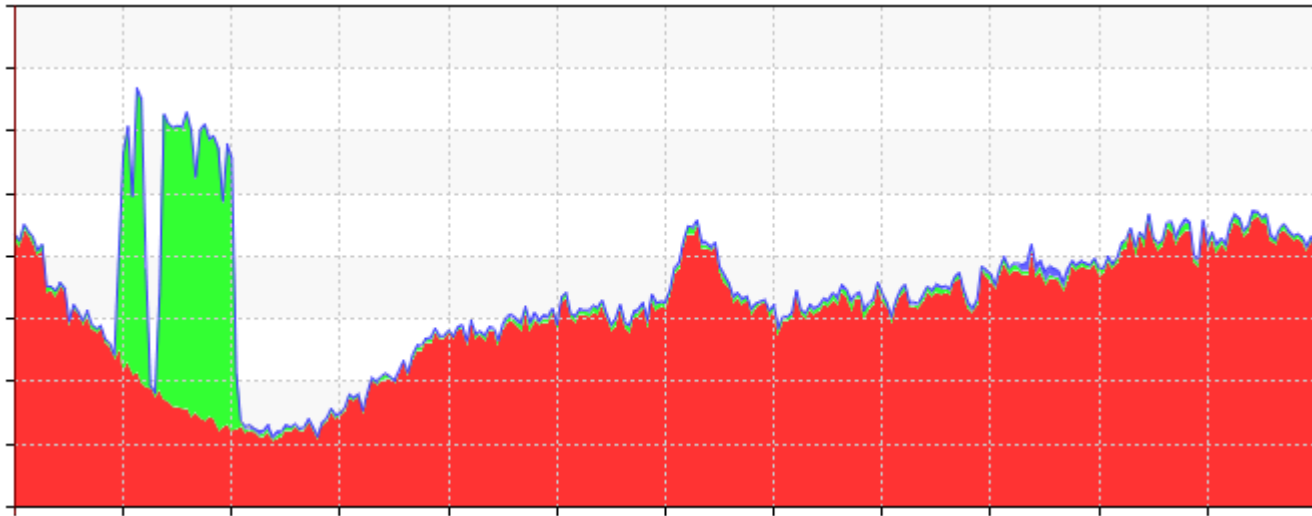


## case 3: netflow graph(dst AS, pps)



- according to dst AS based graph, the anomaly traffic was directed to a particular AS(yellow)

## case 3: netflow graph(protocol, pps)



- the traffic profile was mostly UDP

# monitoring and detection

- snmp is useful to check
  - trend
  - threshold
- netflow is useful to analysis
  - anomaly
  - change

# Operational Design

