

# غیر فعال نمودن سرویس

## (Denial of Service)

# حملات DoS

- فایده ای برای مهاجم ندارد، بلکه باعث آزار قربانی می شود.
  - استثناء: مهاجم برای متوقف کردن حمله از قربانی باج می گیرد.
- مشکلی اساسی در اینترنت امروزه
- اکثر ویروس ها و کرم ها مرتکب حمله DoS می شوند.
- باید بین حملات "Slashdot Effect" و "DoS" یا "crowds" تمییز قائل شد.

# چه چیزهایی مورد حمله قرار می‌گیرند؟

- پهنای باند
- CPU
- قربانی را مجبور به محاسبات سنگین می‌کند.
- حافظه
- در حملات DoS، هزینه ارسال پیام توسط مهاجم **کمتر** از هزینه پردازش آن توسط قربانی است.

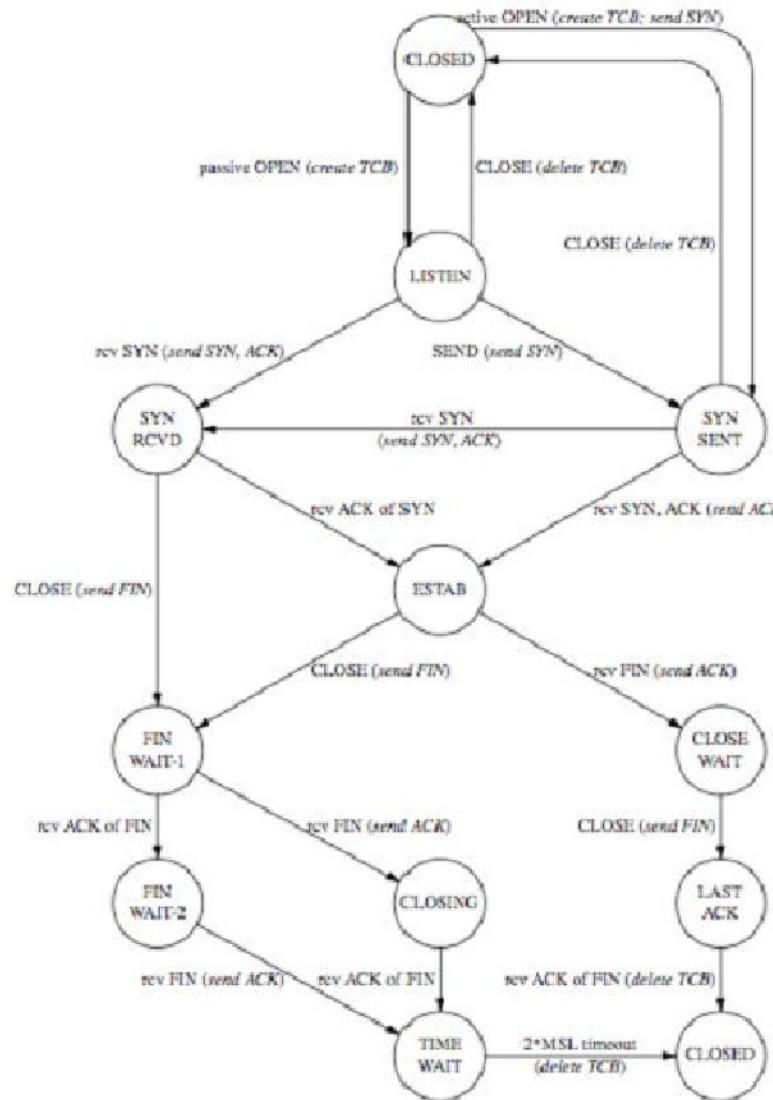
# DOS حملات نمونه

- حمله SYN
- حمله Ping of death
- حمله Smurf
- کرم ها
  - مصرف منابع در ماشین قربانی
  - کند کردن شبکه با استفاده از تعداد زیاد بسته
- اجرای پروسس ها در حلقه بی نهایت
  - Fork

# اولین حمله DoS در اینترنت

- مهاجم تعداد زیادی بسته SYN با آدرس های تقلبی برای قربانی ارسال می کند.
- قربانی بسته های SYN-ACK را به هاست های نامربوط ارسال می کند.
- هیچ بسته ACK ای به قربانی نمی رسد و اتصال به صورت نیمه باز باقی می ماند.

# TCP State Diagram



# SYN Flooding

- بسته SYN اتصال را به حالت SYN-RCVD می برد و بسته SYN-ACK را به کلاینت ارسال می کند.
- سیستم در این حالت می ماند تا بسته ACK را دریافت کند.
- تعداد اتصالات به یک پورت در حالت SYN-RCVD محدود است.
- بسته های SYN بعدی مربوط به آن پورت دور انداخته می شوند.
- مهاجم از تکنیک جعل آدرس مبدأ استفاد می کند.
  - اگر مهاجم آدرس یک هاستی که وجود ندارد را جعل کند، هیچکدام از بسته های ACK یا RST به قربانی نمی رسد.
- بنابراین پورت بلاک می شود.

# مکانیزم های دفاعی

- مقابله با جعل آدرس (Anti-Spoofing)
- استفاده از ساختمان های داده ای بهتر
- SYN Cookies

# Anti-Spoofing

- ایده ای ساده، اما باید در مقیاس گسترده ای پیاده شود.
- ISP ها بسته های outbound ای که دارای آدرس مبدأ غیر از شبکه داخلی هستند را فیلتر کنند.
  - هزینه این فیلترینگ بالاست.
  - باز هم به صورت محلی امکان spoofing وجود دارد.
  - باید به ISP ها اعتماد داشت!!

# استفاده از ساختمان های داده ای بهتر

- دلیلی برای اختصاص کامل حافظه مورد نیاز فقط برای یک بسته SYN وجود ندارد.
- حافظه فشرده تری اختصاص داده شود و برای اتصالات نیمه باز محدودیت قائل شد.
- می توان تعداد بیشتری اتصال را مدیریت کرد ولی باز هم مهاجم با افزایش نرخ می تواند مشکل ساز شود.

# راه حل عام

- عدم ایجاد حالت تا زمانی که مورد نیاز باشد.
- سرور حالت را رمز کند و به کلاینت ارسال کند.
- کلاینت حالت رمز شده را در بسته سوم (ACK) به سرور ارسال باز می گرداند.
- سرور با رمزگشایی حالت را بازیابی می کند و اتصال را برقرار می سازد.

# SYN Cookies

- توسط Dan Bernstein ابداع شده است.
- ایده کلی: تولید ISN سرور با استفاده از شمارنده زمان(T)، MSS و تابع رمزگاری ۲۴ بیتی
  - $T = 5\text{-bit counter incremented every 64 secs.}$
  - $L = \text{MAC}_{\text{key}}(\text{SAddr}, \text{SPort}, \text{DAddr}, \text{DPort}, \text{SN}_C, T)$  [24 bits]
    - key: picked at random during boot
  - $\text{SN}_S = (T \cdot \text{mss} \cdot L)$  ( $|L| = 24 \text{ bits}$ )

- کلاینت صادق با (  $\text{ACK} (\text{AN}=\text{SN}_S, \text{SN}=\text{SN}_C+1)$  جواب می دهد.
- هنگامی که بسته ACK از کلاینت رسید، با استفاده از تابع رمز ۲۴ بیتی صحت اطلاعات بررسی شده و اتصال ساخته می شود.

# CPU DoS

- استفاده از SYN Cookies به پردازش زیادی برای محاسبات رمزی احتیاج دارد.
- مهاجم می تواند زمان CPU را تلف کند.
- احتیاج به راهی برای محدود کردن نرخ درخواست ها وجود است.

# روش شناسائی حمله SYN

- با استفاده از تعداد **SYNACK – FIN** یا **SYN/FIN** در ارتباط عادی TCP
- در سمت کلاینت یا سرور قابل انجام است
- در بعضی از ارتباطات تعداد **RST** نیز بر تعداد **SYN/FIN** تاثیر می گذارد.
- **RST منفعل**: در صورت دریافت یک بسته tcp به یک پورت که وجود ندارد(در طرف سرور اتفاق می افتد)
- **RST فعال**: در صورت درخواست سرور یا کلاینت برای قطع ارتباط ارسال می شود(مثلا با زدن Ctrl+D برای قطع ارتباط telnet)
  - داده های بافر شده دور ریخته می شود.
- پی ساید تعداد **SYN-RST** را هم در نظر گرفت

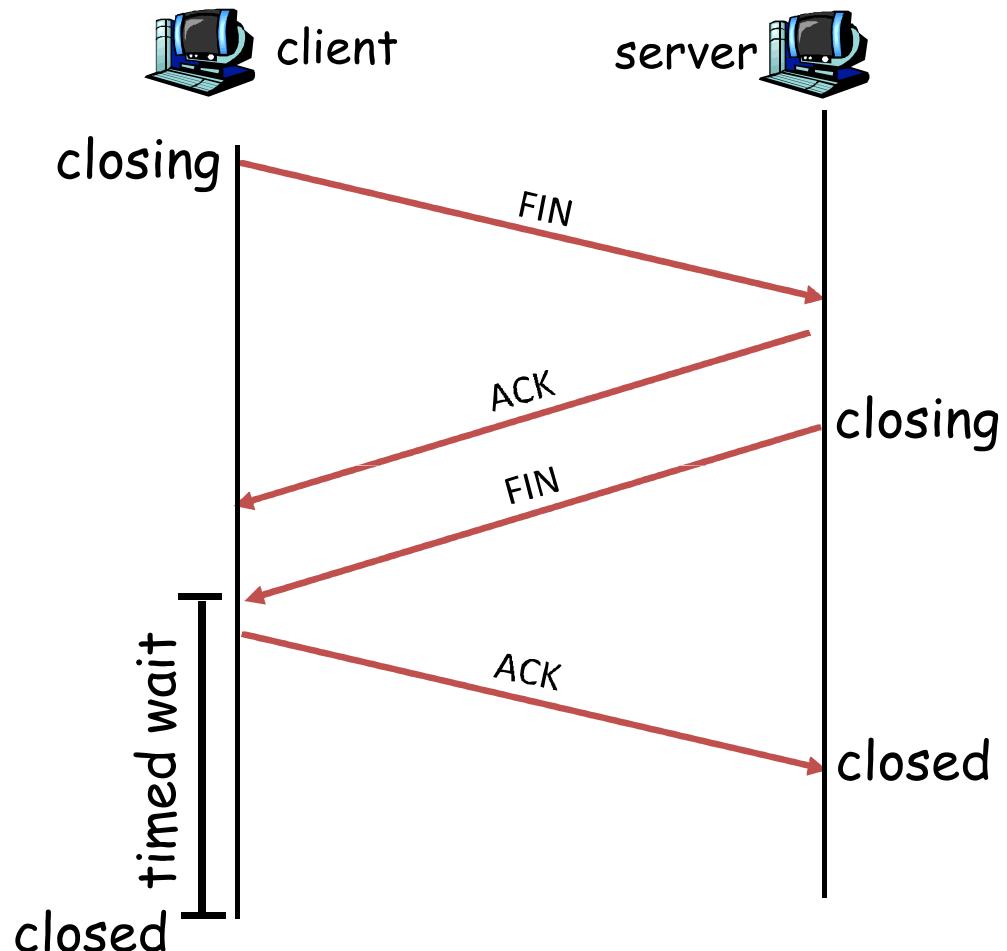
# مدیریت ارتباط در TCP

قدم ۱: کلاینت با ارسال FIN درخواست قطع ارتباط می‌دهد.

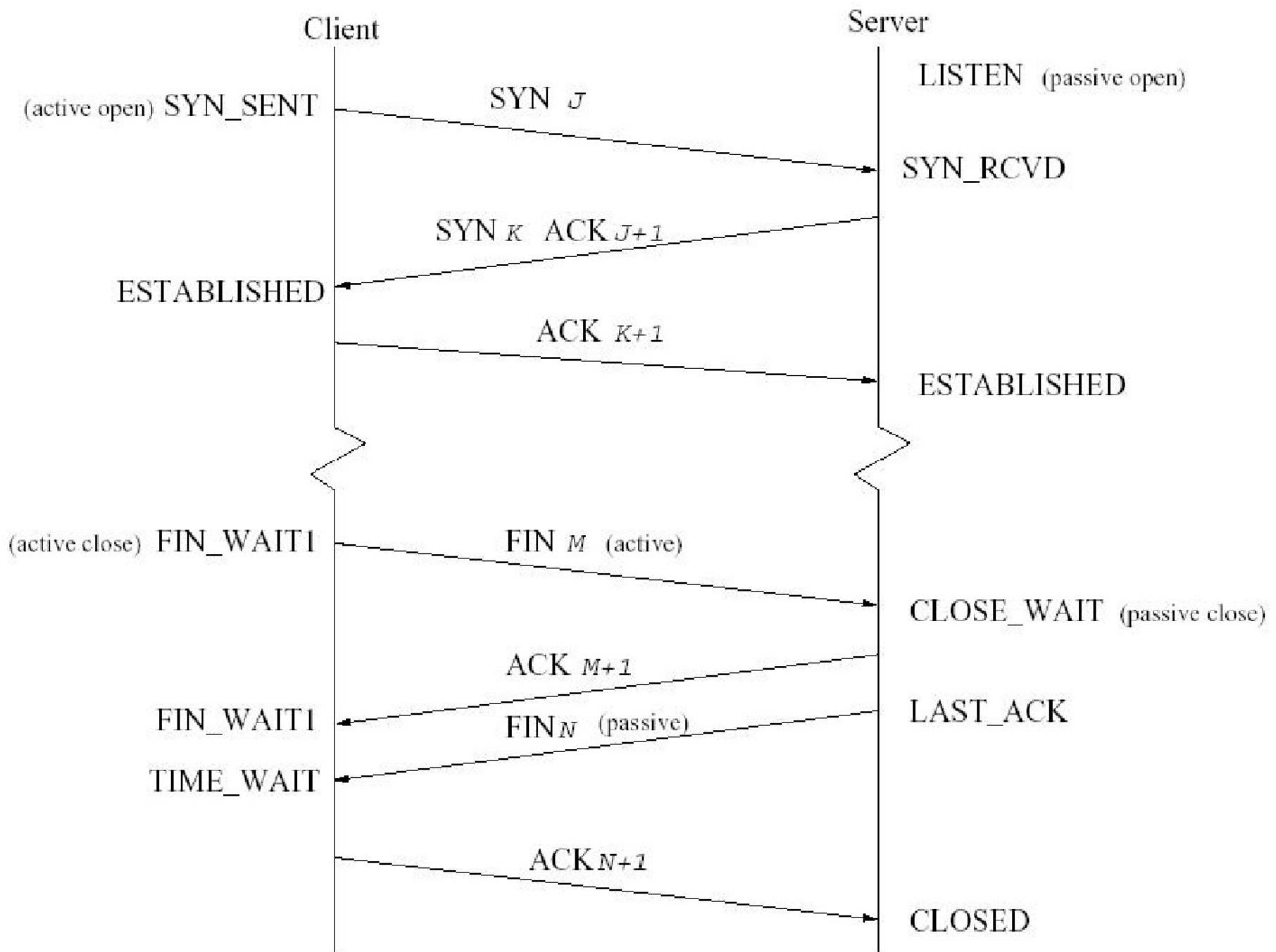
قدم ۲: سرور بعد از دریافت FIN و ارسال ACK به حالت قطع کردن ارتباط می‌رود.

قدم ۳: کلاینت بعد از دریافت FIN، یک ACK ارسال کرده و به حالت time-wait می‌رود تا بعد از قطع ارتباط timeout

قدم ۴: سرور بعد از دریافت ACK ارتباط را می‌بندد

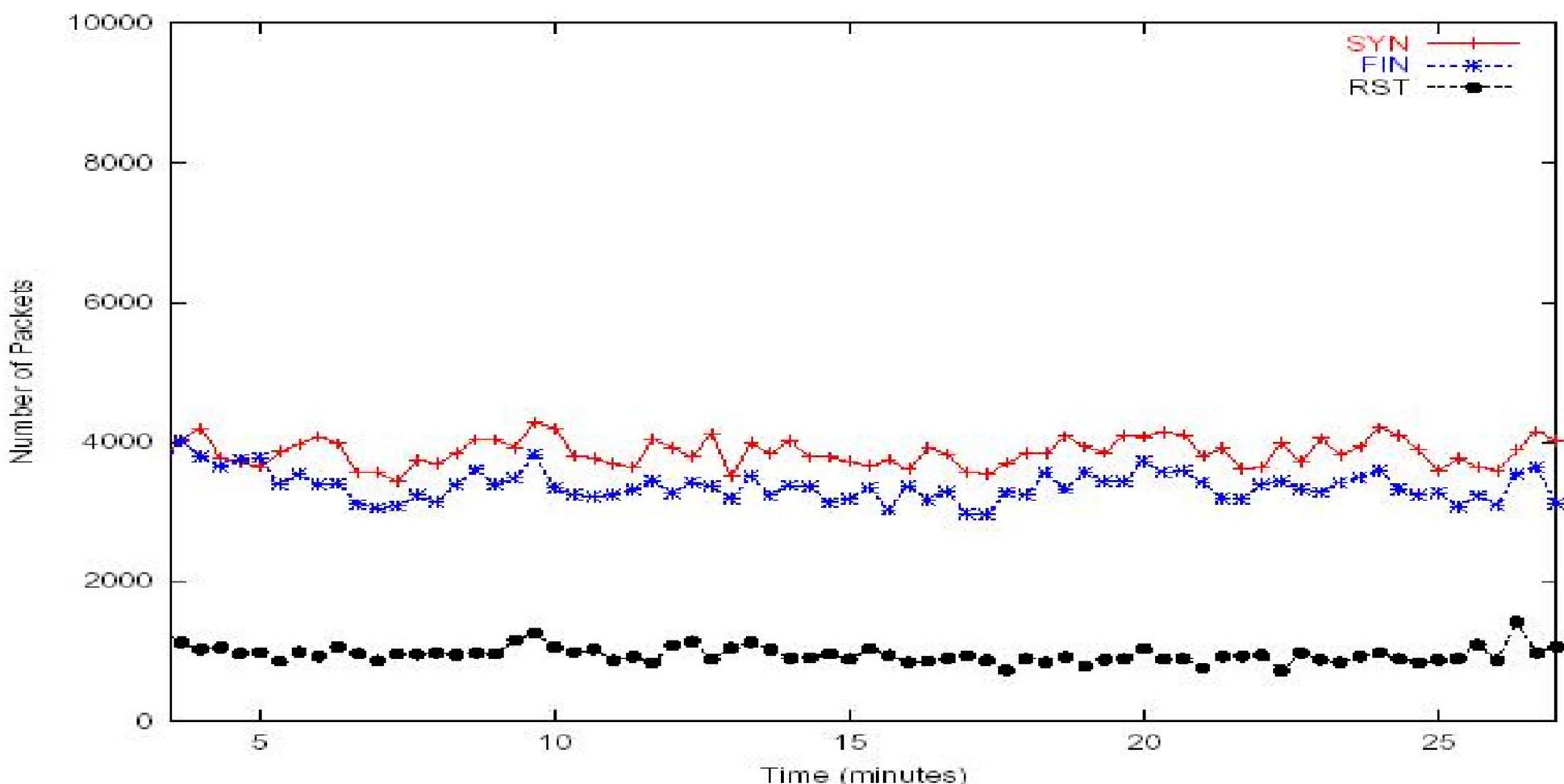


# SYN – FIN Behavior



# رفتار SYN – FIN

- به طور کلی هر SYN یک FIN دارد.
- در مورد فعال بودن و نبودن RST ها نمی توان نظر داد ولی معمولاً 75% از RST ها فعال هستند.
- تعداد  $\text{FIN} + .75 * \text{RST}$  تقریباً باید برابر تعداد SYN باشد



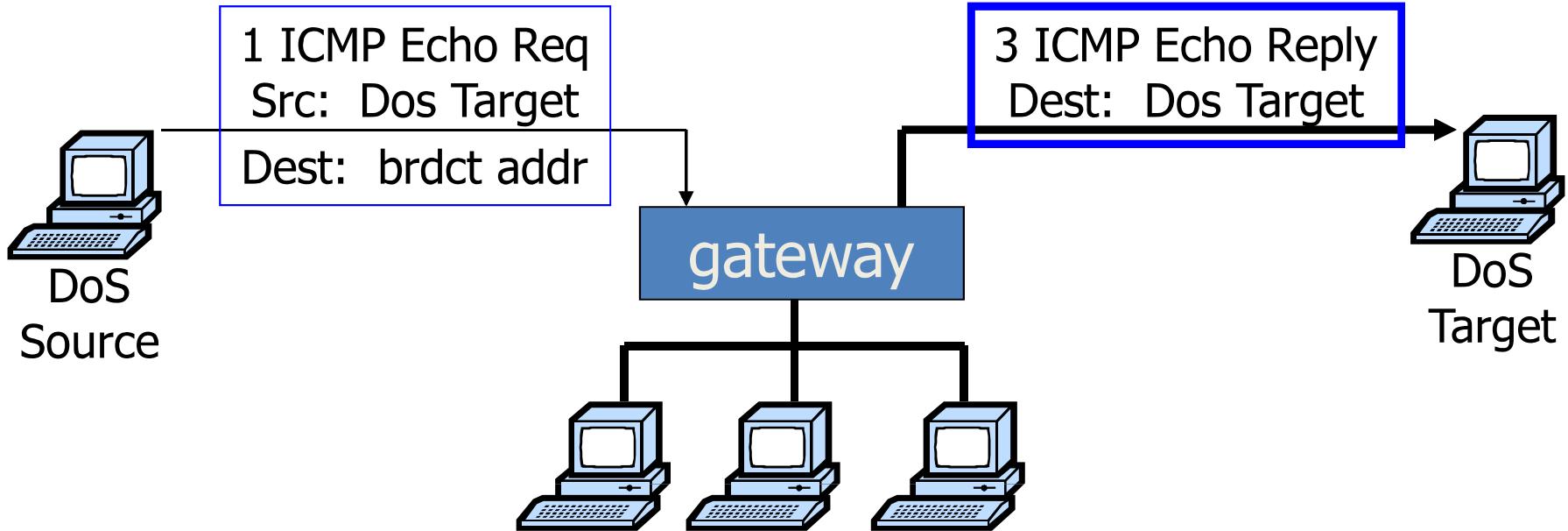
## مشکل نفوذ شناسائی با SYN-FIN

- ارسال FIN های اضافی با پورت و IP متفاوت
- از بین بردن عرض باند

# Ping of death حمله

- ارسال بسته های ICMP بزرگ که قبل 1996 باعث crash کردن تجهیزات می شد..
- ارسال بسته های بزرگتر از ۶۵۵۳۶ بایت باعث fragmentation می شود که توسط ماشین مقصد باید به هم متصل شوند
- تعداد زیاد باعث خرابی مقصد می شود.

# Smurf حمله



- ارسال ping به آدرس broadcast با آدرس مبداء قربانی
- تمامی host هایی که این ping را می گیرند یک ICMP reply به قربانی می فرستند
- Ping reply ها می توانند باعث overload شدن قربانی شوند

دفایع: جلوگیری از دریافت ping با آدرس broadcast

# Bayesian Filter

- Bayesian Fitler برای جلوگیری از اسپم استفاده می شود.
- مهاجمین با ارسال اسپم های خاص قصد مصرف زمان CPU توسط Bayesian Filter را دارند.
- در نتیجه سایت ها فیلترینگ رو خاموش می کنند.
- ایمیل های اسپم به راحتی عبور می کنند.

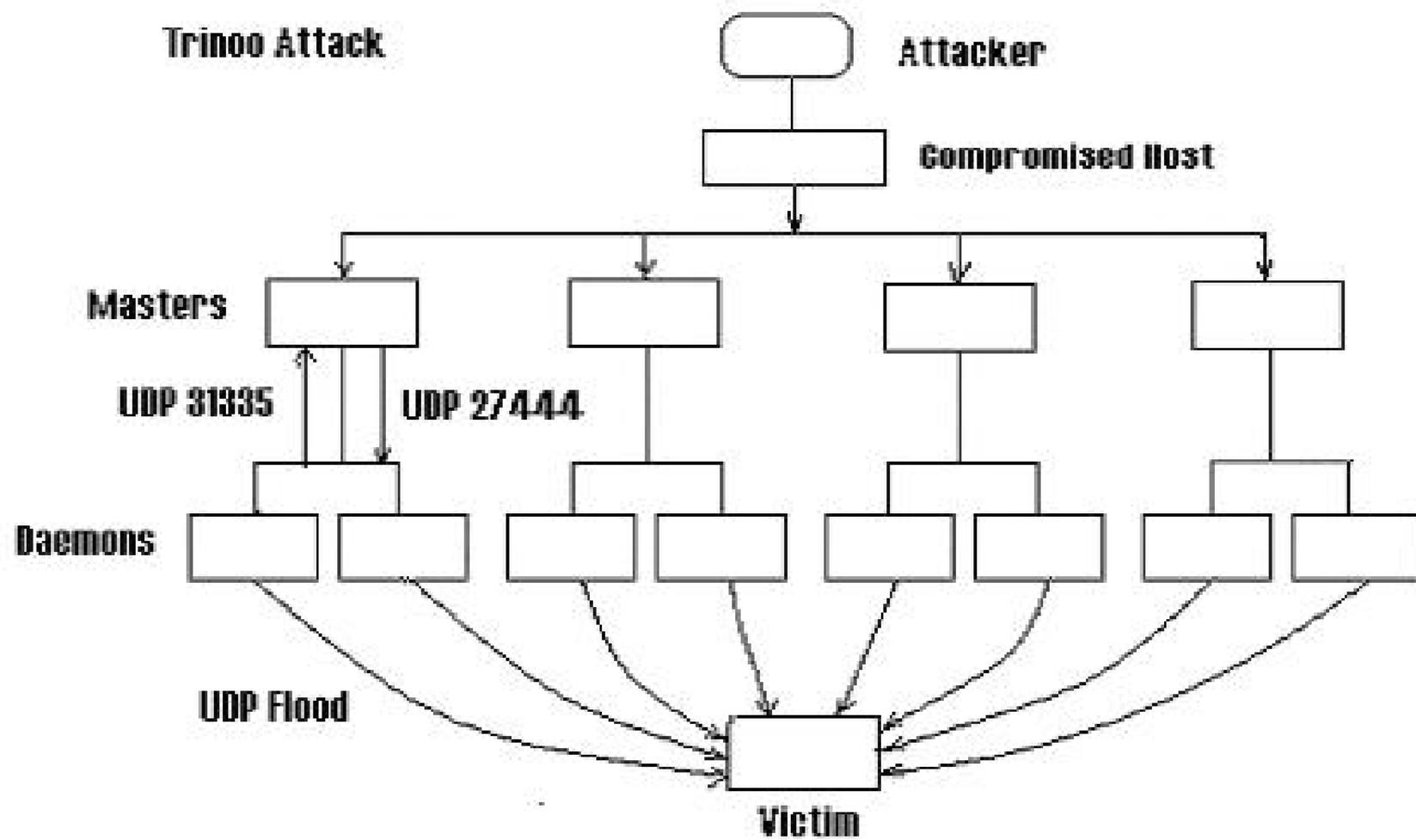
# Reflector Attacks

- مهاجم یک بسته کوچک با آدرس مبدأ جعل شده به یک سرور می فرستد، مخصوصاً DNS.
- سرور یک بسته بزرگتر به عنوان پاسخ تولید می کند.
- این پاسخ به آدرس جعل شده فرستاده می شود.
- بنابراین مهاجم با هزینه کم یک حمله DoS ایجاد می کند و پنهان می شود.

# Distributed Denial of Service (DDoS)

- معمول ترین نوع DoS در اینترنت امروزه
- مصرف کردن پهنهای باند شبکه
- استفاده از شبکه بزرگی از zombie bot ها یا bot هاست های "دستور و کنترل" به bot ها دستور می دهند.
- اغلب استفاده از پروتکل IRC برای کanal های کنترل
- نمونه های جدید از پروتکل های P2P استفاده می کنند.

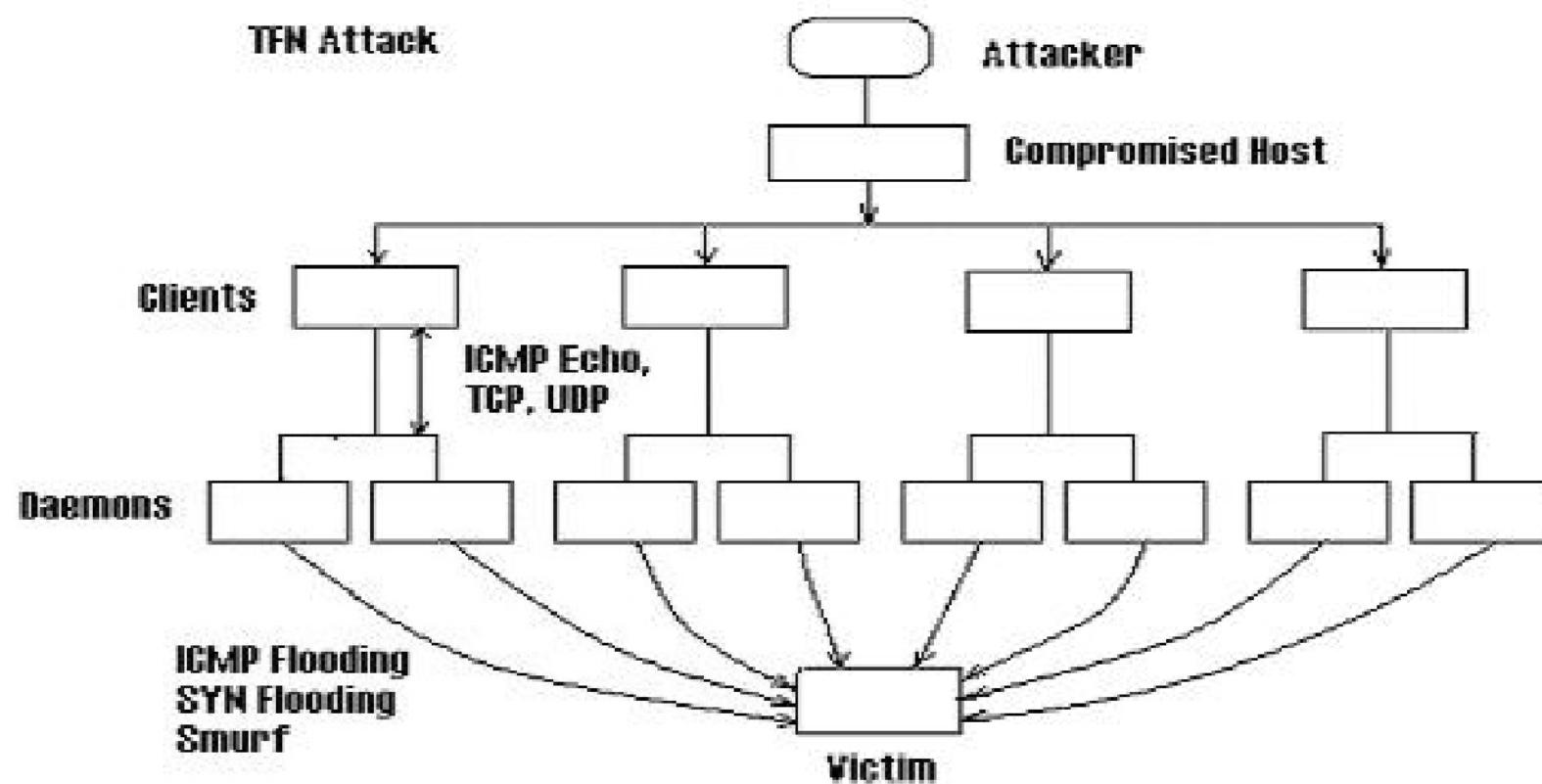
# Trinoo



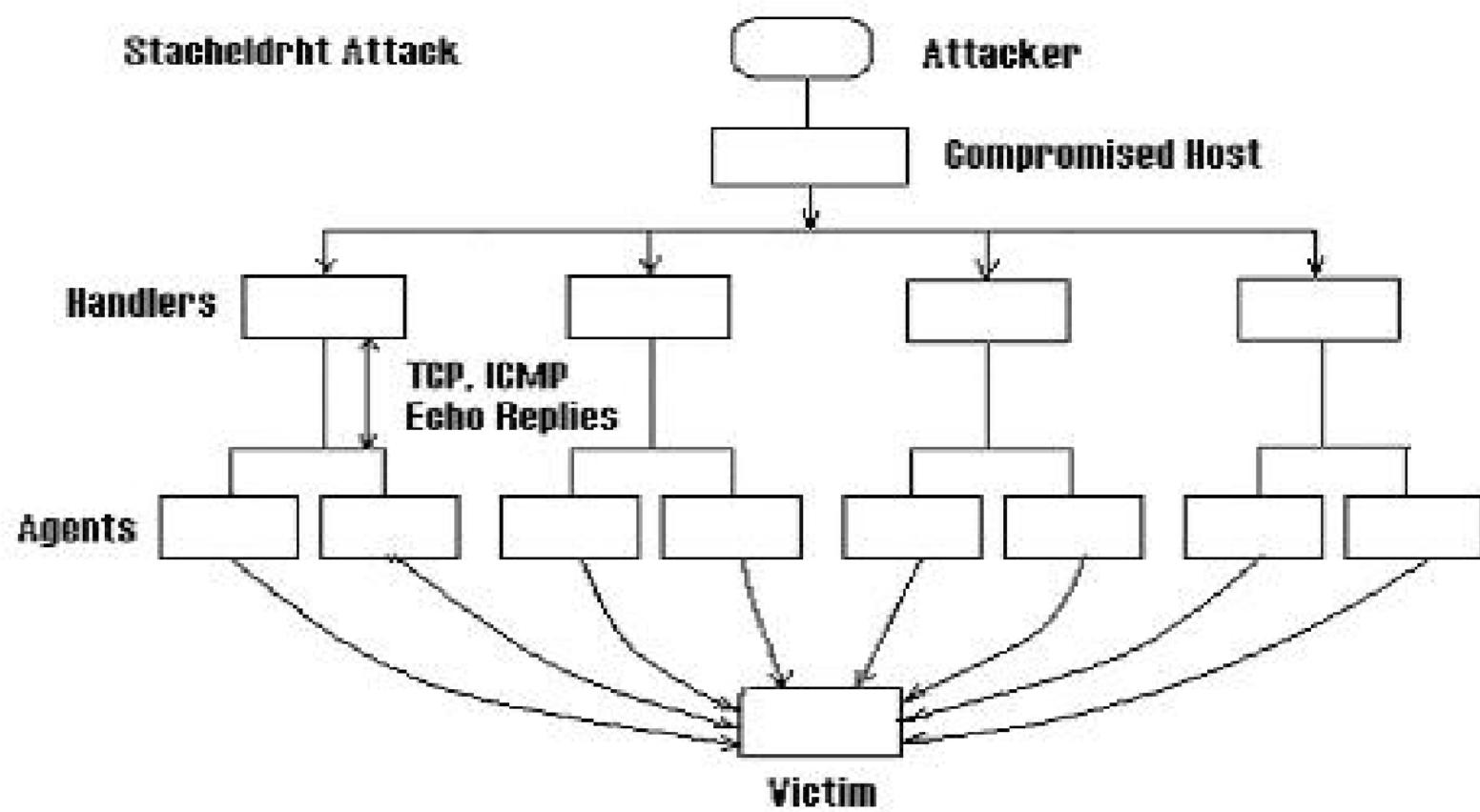
# مثال هائی از Trinoo-DDoS

- شامل masters and Daemones هست.
- از UDP DoS Flood برای حمله استفاده می کند.
- مراحل زیر انجام می شود:
- قدم ۱: حمله کننده با استفاده از یک کامپیوتر که به صورت غیر قانونی تحت کنترل در آمده است، مجموعه ای از کامپیوترها را که می توان نفوذ کرد را تهیه می کند.
- قدم ۲: سپس با استفاده از یک script و به صورت خودکار به آنها نفوذ شده و آنها به master یا daemon تبدیل می شوند. Daemon ها کامپیوترهایی هستند که مستقیماً حمله می کنند. هر گروه از Daemon ها تحت کنترل یک master هستند.
- قدم ۳: برای حمله attacker یک پیام به master ها می فرستد و هم هادستور حمله صادر می کند.

# TFN/TFN2K



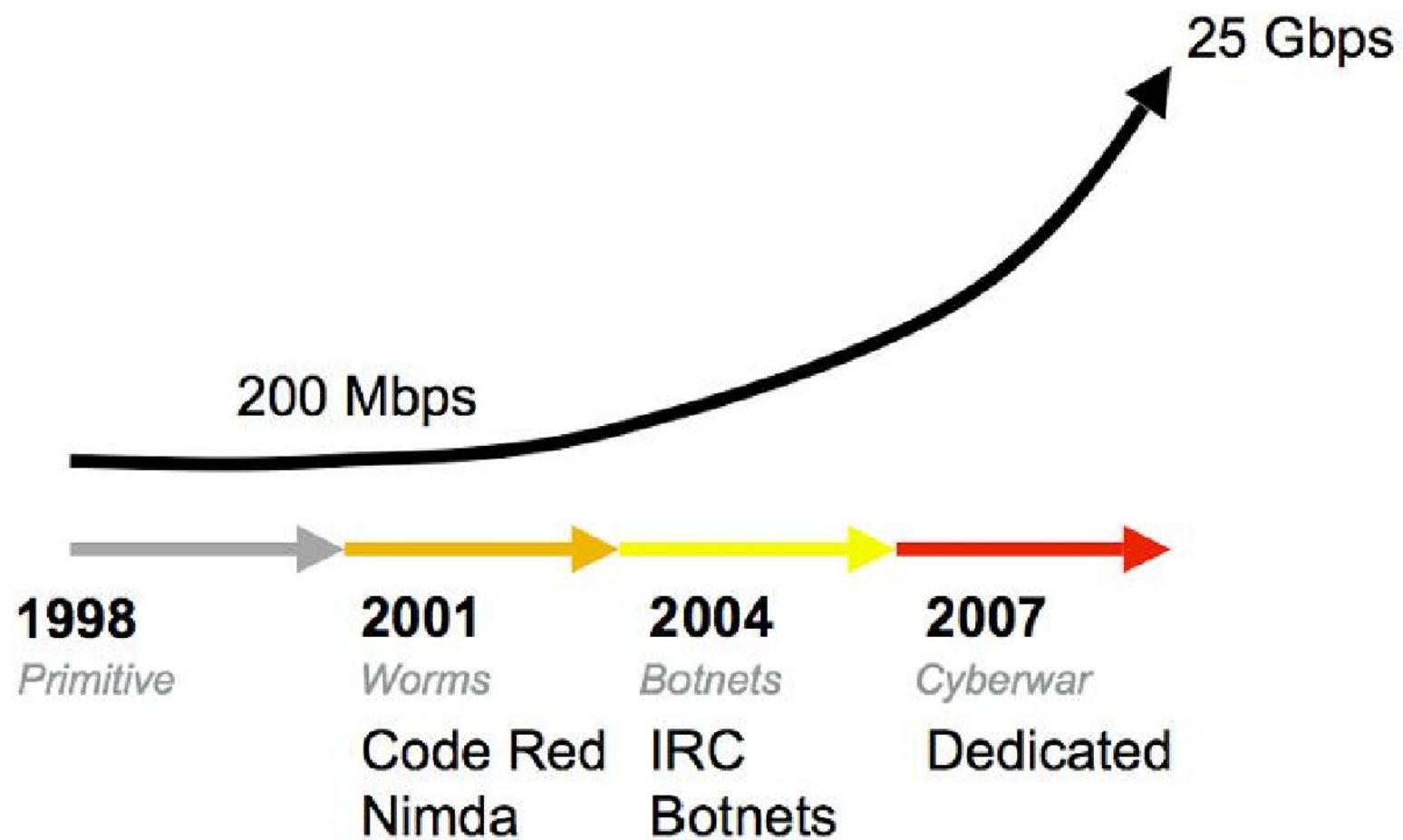
# Stacheldraht(سیم خاردار)



# Address Spoofing

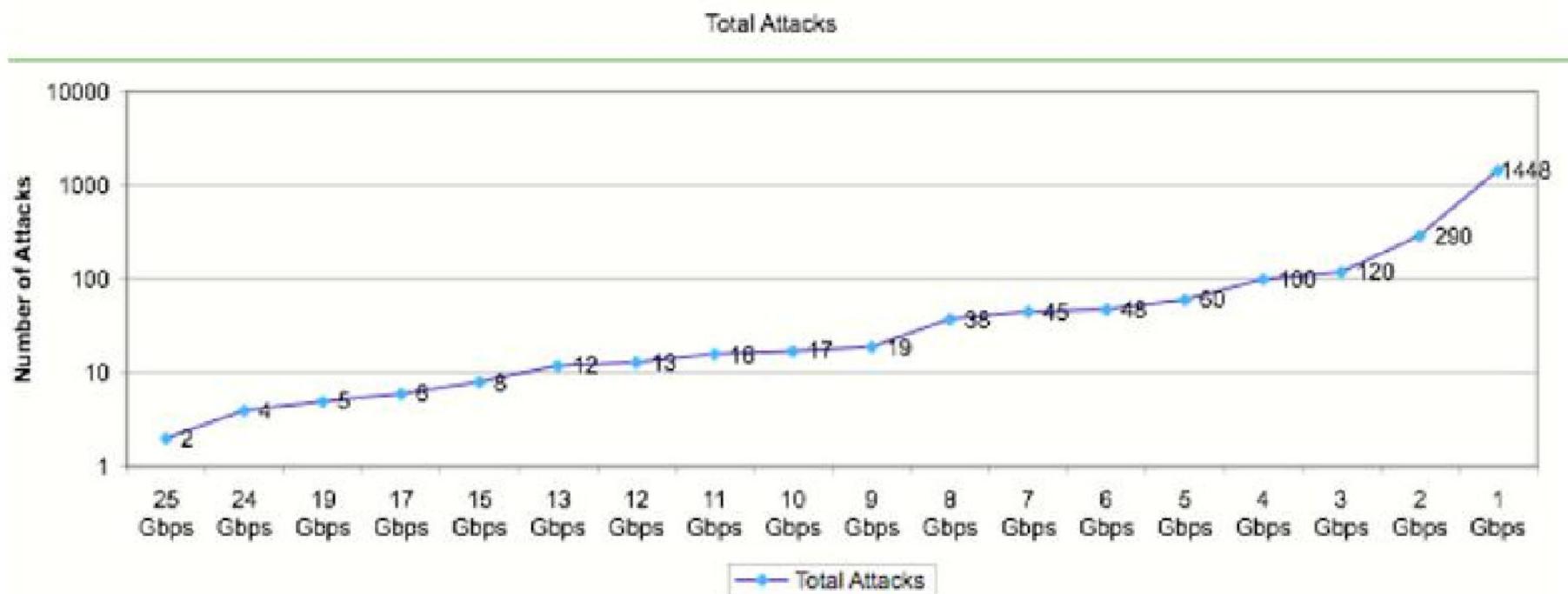
- نمونه های اولیه از جعل آدرس استفاده می کردند.
  - کار فیلترینگ و پیدا کردن مهاجم سخت تر می شود.
  - بنابراین مکانیزم های دفاعی روی پیدا کردن مهاجم تمرکز داشتند.
- خیلی از حملات جدیدتر از جعل آدرس استفاده نمی کنند.
  - چرا که فیلترینگ و پیدا کردن مهاجم کارایی لازم را ندارد.
  - مدیر شبکه توانایی برخورد با ۱۰۰۰۰ bot را ندارد.
  - پیدا کردن ماشینی که bot باشد، سودی به مدیر ندارد چرا که فرد مسئول ماشین مورد نظر مورد سوء استفاده قرار گرفته است.
  - خیلی از روتراها قابلیت فیلتر کردن ۱۰۰۰۰ IP را ندارند.

# تاریخچه DDoS



# حجم حملات DDoS در اینترنت

- انجام ۳ حمله ۱Gbps در یک روز
- انجام ۲ حمله ۲۵Gbps در یک روز به مدت ۳۵ دقیقه



# حملات DDoS روی کشور استونی



- کشور استونی در همسایگی روسیه دارای زیر ساخت شبکه ای خوبی است.
- بعد از مشکلاتی که با روسیه پیدا کرد، در سال ۲۰۰۷ مورد حمله DDoS قرار گرفت.
- بیشتر سایت های حکومتی و بانکی غیر فعال شد.

# مبدأ حملات استونى



# مکانیزم های دفاعی

- Puzzle
- Black-Hole Routing
- فیلترینگ آنومالی
- Pushback
- IP traceback

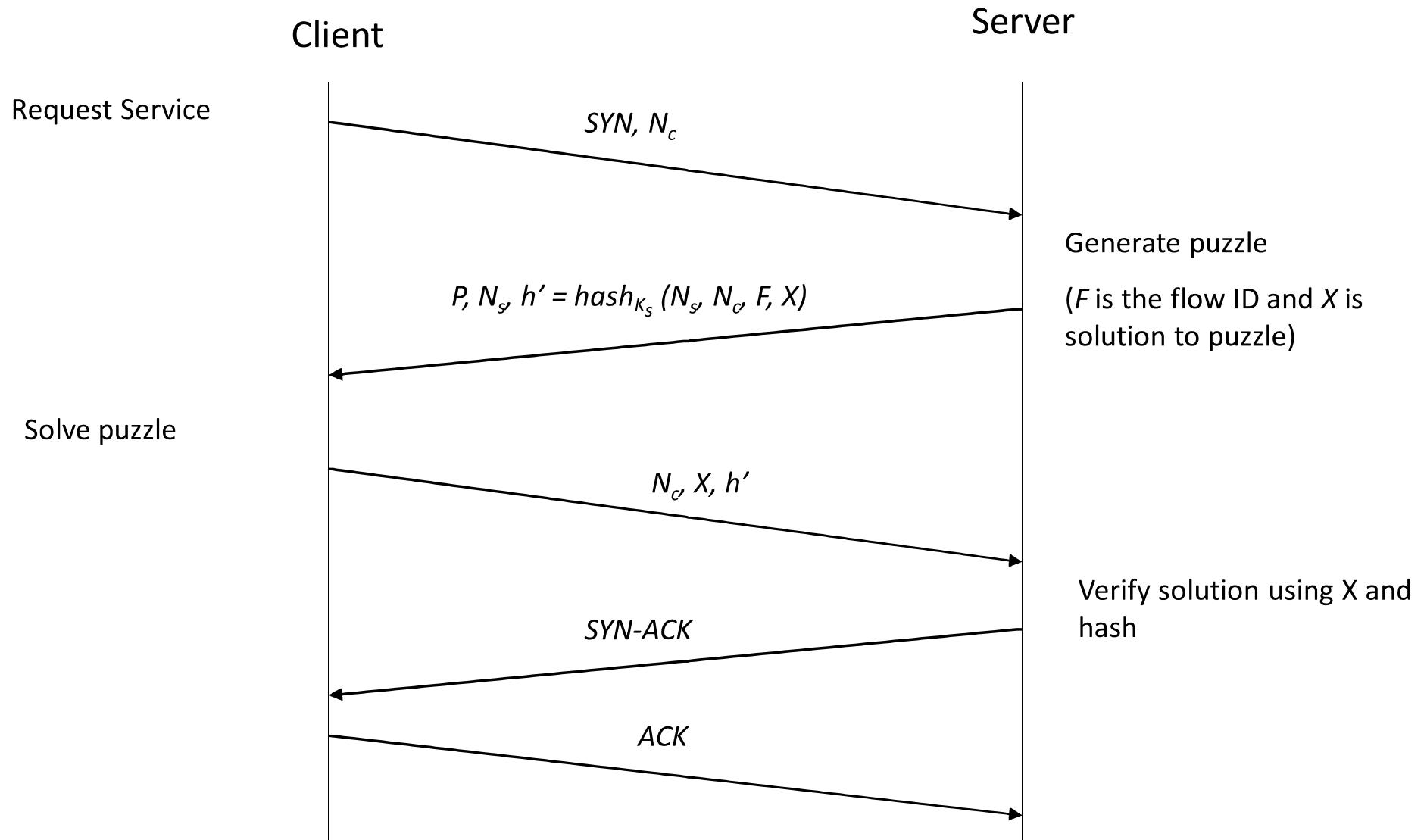
# Puzzles دفاع

- ایده کلی: ساخت پازلی که حل آن پر هزینه و تأیید آن کم هزینه باشد.
- قبل از انجام کاری پر هزینه، کلاینت را مجبور به حل پازل کنیم.
- برای کلاینت های مجاز مشکلی ایجاد نمی شود، بلکه کار مهاجمین را سخت می کند.

# Hash Puzzle

- دو عدد تصادفی  $n$  و  $x$  را تولید می کنیم.
- تاپل  $\langle n, h(x), y \rangle$  را به کلاینت می فرستیم.
  - $h$  یک تابع در هم سازی است.
  - $y$  همان عدد  $X$  است که  $n$  بیت کم ارزش آن صفر شده باشد.
- کلاینت باید  $X$  را پیدا کند.
- سرور باید جواب کلاینت را تأیید کند تا اتصال برقرار شود.

# Hash Puzzle



# Hash Puzzle

- تنها راه کلاینت برای پیدا کردن X استفاده از brute force است.
- پیدا کردن X به طور متوسط  $2^{n-1}$  عملیات می برد.
- تأیید کردن آن فقط 1 عملیات صرف می کند.

# Black-Hole Routing

- تنظیم مسیریابی ISP به قسمی که تمام ترافیک به سمت قربانی را به سمت sinkhole منحرف کند.
- این مسئله باعث می شود سایت های دیگر نیز از حمله در امان باشند.
- بیشتر حملات DDoS عمر کوتاهی دارند.

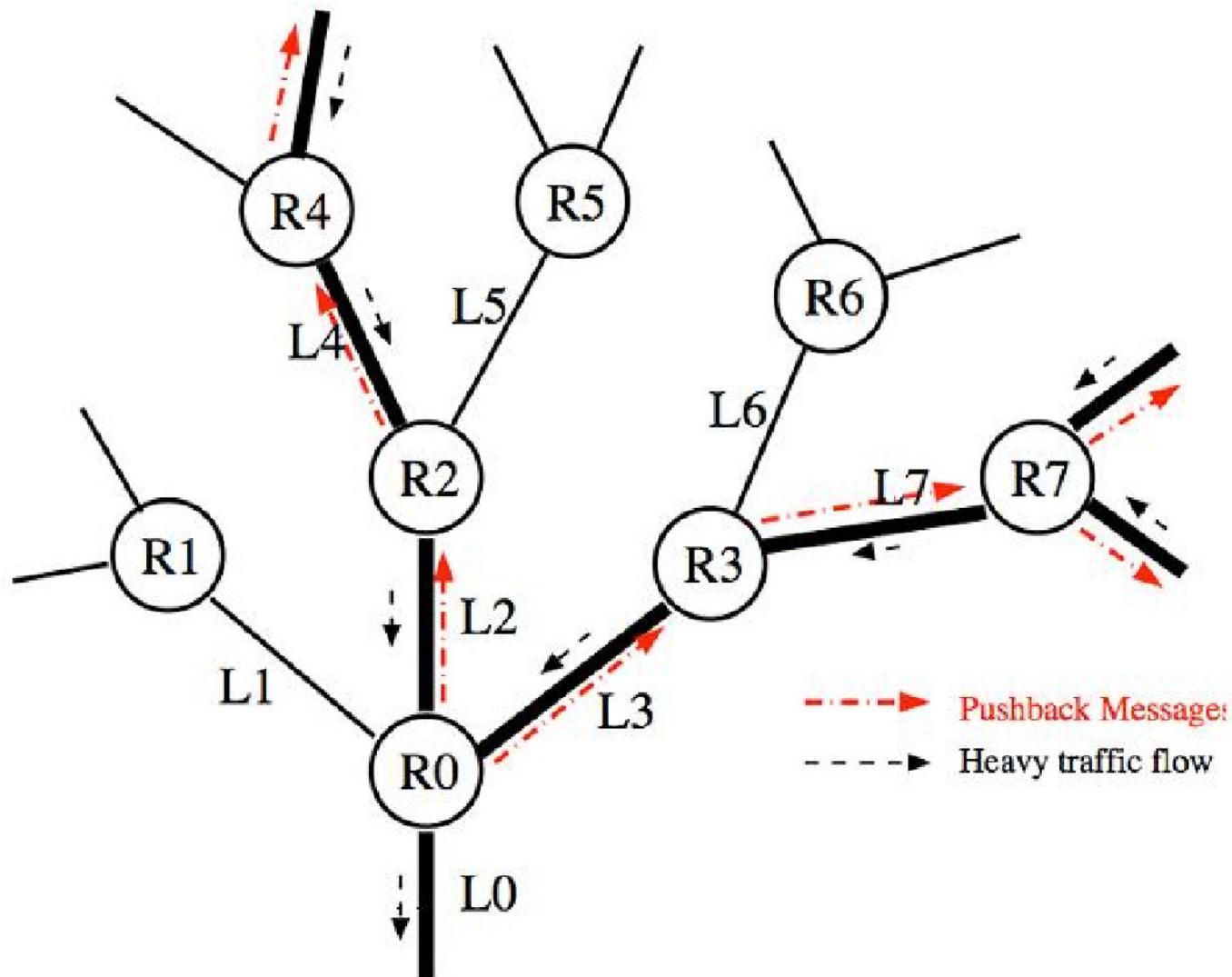
# فیلترینگ آنومالی

- ترافیک DDoS معمولاً نرمال نیستند.
- TTL و نوع پروتکل آنها اغلب غیر عادی هستند.
- ترافیک های غیر نرمال را فیلتر کنیم.
- این روش به طور کامل کارساز نیست ولی معمولاً جوابگو نیازها است.

# Pushback

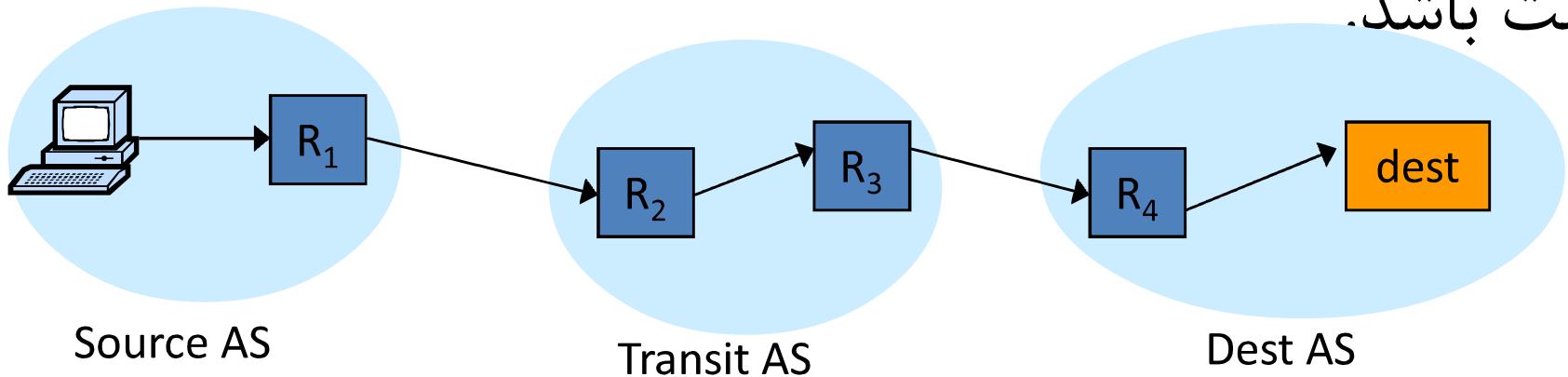
- هنگامی که لینک خروجی رو تر بلاک شد، لینک های ورودی که بسته می آید را بررسی می کند.
- به رو ترهای ورودی خبر می دهد تا نرخ ارسال بسته ها را کاهش دهند.
- این الگوریتم به صورت بازگشتی صورت می گیرد.

# Data Flow



# روش های دفاع با شناسائی منبع

- مشکلات:
  - باید در همه ISP ها پیاده سازی شود. باید به همه ISP ها اعتماد کرد. اگر ۱۰٪ از ISP ها پیاده سازی نکنند در اینصورت امکان دفاع نیست.
  - AS ها فقط بسته هایی را قبول می کنند که آدرس منبع درست باشد.



# IP Traceback

Traceback [Savage et al. '00] •

- بعد از دریافت بسته های حمله مسیر تا منبع تولید حمله مشخص شود.
- مسیریاب ها باید طوری تعییر یابند که اطلاعات مسیر در بسته ها ذخیره شود.
- فرض می شود:
  - مسیریاب ها تحت کنترل حمله کننده نیست.
  - تعداد بسته های حمله زیاد است.
  - مسیر حمله از مبدأ تا قربانی تقریباً ثابت است.

## روش ساده

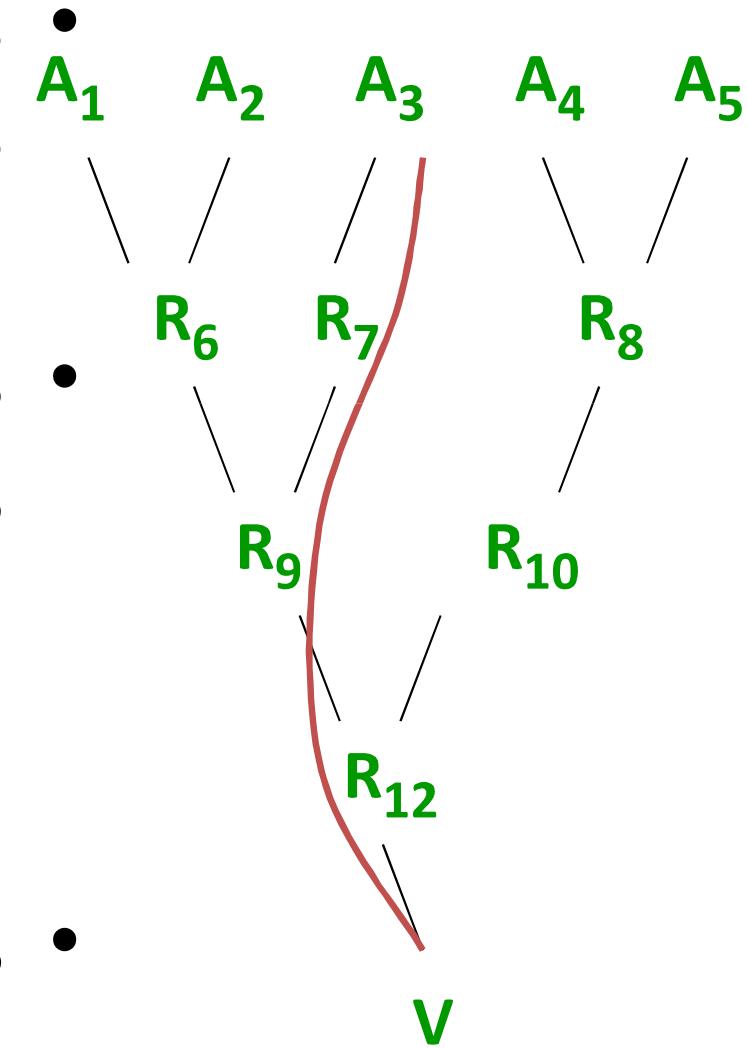
- هر مسیر یا ب آدرس IP خود را به بسته اضافه می کند.
  - قربانی مسیر را از بسته ها استخراج می کند.
- 
- مشکل
  - نیاز به فضا در داخل بسته IP
  - مسیر می تواند بزرگ باشد.
  - در فرمت بسته IP فعلی وجود ندارد و تغییر بسته IP انتظار زیادی است.

# روش بهتر

در DDoS تعداد بسیار زیاد بسته در یک مسیر ارسال می شود.

در یک بسته مشخصات یک مسیریاب ذخیره شود.  
- هر مسیر یاب با یک احتمالی آدرس خود را در بسته ذخیره می کند.

مستقل از طول مسیر نیاز به یک فضای ثابت است.



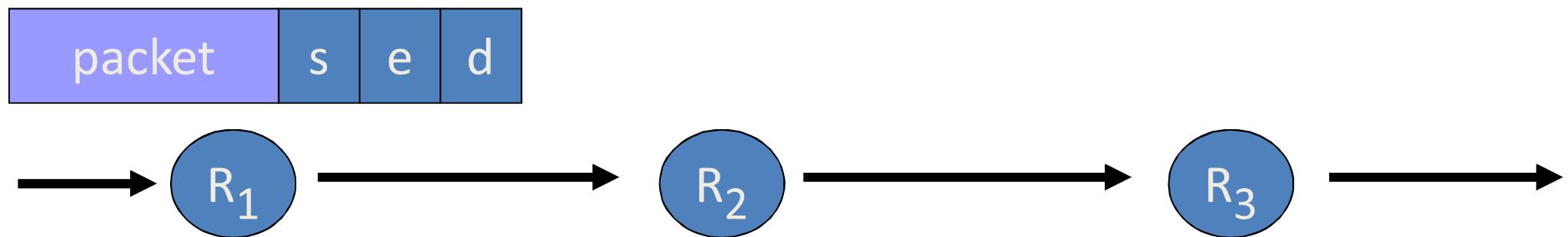
# نمونه برداری لبه

- دو آدرس IP "شروع(start)" و "انتهای(end)" و یک عدد صحیح(distance) که تعداد مسیر یا ب از آدرس شروع تا مقصد است، در بسته IP ذخیره می شود.
- الگوریتم تغییر این سه فیلد داده در مسیریاب  $R$

```
for each packet w
    let x be a random number from [0..1)
    if x < p then           //decide to mark packet
        write R into w.start and 0 into w.distance
    else                      // doesn't mark packet
        if w.distance = 0 then //already marked packet
            write R into w.end
        increment w.distance //always increment the distance
```

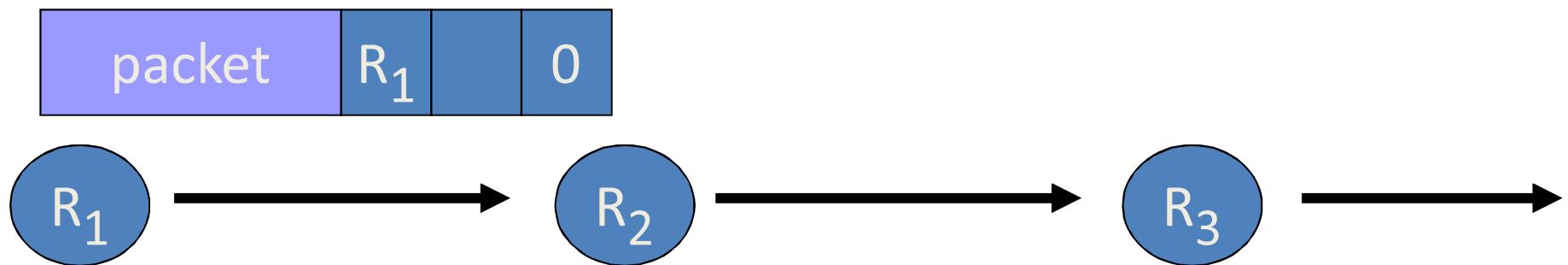
# Edge Sampling: picture

- بعد از دریافت بسته
  - $R_1$  بسته را از مبداء یا مسیریاب قبلی دریافت می کند.
  - بسته شامل سه فیلد start, end, distance است.



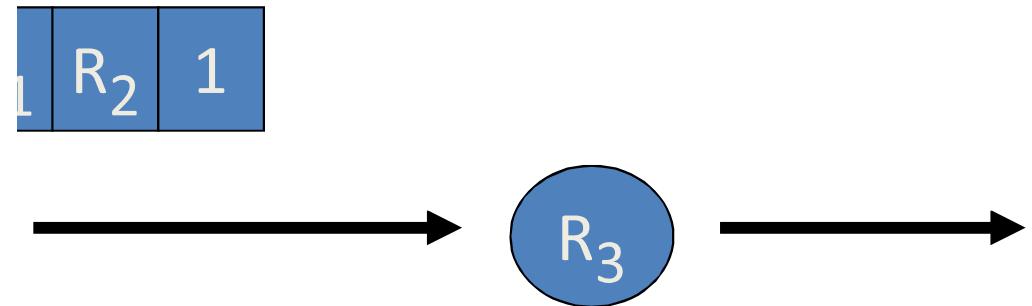
# Edge Sampling: picture

- شروع با نوشتن در لبه  
– تصمیم می گیرد مسیریاب را در edge بنویسد.  
– را برابر ۰ قرار می دهد.



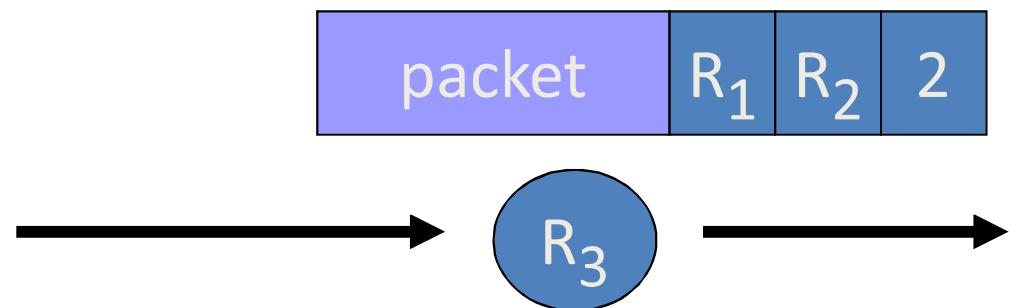
# Edge Sampling

- ❖ R2 نمی خواهد در edge بنویسد.(نمی خواهد mark کند)
- ❖ Distance برابر 0 است
- ❖ فیلد end نوشته شده و distance یکی اضافه می شود.



# Edge Sampling

R3 نمی خواهد در بسته بنویسد و  $distance > 0$  است: ◊  
یکی اضافه می شود. ◊ Distance



## **Path reconstruction procedure at victim v:**

*let G be a tree with root v*

*let edges in G be tuples (start,end,distance)*

*for each packet w from attacker*

*if w.distance = 0 then*

*insert edge (w.start,v,0) into G*

*else*

*insert edge (w.start,w.end,w.distance) into G*

*remove any edge (x,y,d) with d ≠ distance from x to v in G*

*extract path (R<sub>i</sub>..R<sub>j</sub>) by enumerating acyclic paths in G*

# Path reconstruction

- اطلاعات از بسته های حمله استخراج می شود.
- یک گراف که ریشه آن "قربانی" است درست می شود.
  - هر سه تائی (start,end,distance) یک لبه گراف است.
  - تعداد بسته لازم برای ساختن مسیر

$$E(X) < \frac{\ln(d)}{p(1-p)^{d-1}}$$

احتمال  $p$  mark کردن و  $d$  طول مسیر حمله است.

# محل ذخیره اطلاعات

fragmetation ۱۶ بیت فضای ID برای مورد استفاده قرار می گیرد.

- به ندرت اتفاق می افتد.

$\text{edge-id} = \text{start xor end}$

$\text{edge-id}$  در فیلد  $k$ ام بخش edge-chunk ذخیره می شود.

$k$  در offset ذخیره می شود

offset	distance	edge chunk
0	23	78 15

برابر آدرس دو مسیریاب پشت سر هم است

Version	Header Length
Type of Service	
Total Length	
Identification	
Flags	Fragment Offset
Time to Live	
Protocol	
Header Checksum	
Source Address of Originating Host	
Destination Address of Target Host	
Options	
Padding	
IP Data	

# دیگر روش های IP traceback

- روشن پیشرفت و همراه با هویت سنجی برای IP Traceback  
Song, Perrig. IEEE Infocomm '01 –  
– کاهش اطلاعات خراب و افزایش سرعت ساخت مسیر ها
- روشن جبری برای IP traceback  
Stubblefield, Dean, Franklin. NDSS '02 –
- Hash-Based IP Traceback  
Snoeren, Partridge, Sanchez, Jones, Tchakountio, –  
Kent, Strayer. SIGCOMM '01

# ICMP Traceback

- با یک احتمال خیلی کم ( $1/20000$ ) محتوی حمله در یک بسته ICMP کپی شده و به مقصد ارسال می کند.
- بسته خاص، اطلاعات مسیریاب های همسایه در مسیر به طرف مقصد را در خود ذخیره می کند.
- با یک حمله با تعداد کافی بسته، بسته های خاص ICMP به تعداد کافی موجود خواهند بود تا بتوان منبع حمله را پیدا کنید.
- مشخصه
  - دارای overhead کم
  - در حال استاندارد سازی توسط The Internet Engineering Task Force, (IETF)

# مراجع

[Practical network support for IP Traceback](#), S. Savage, et al.

[A DoS-Limiting Network Architecture](#), Yang, Wetherall, and Anderson