NA DORAZ

kam až sahají meze síťového subsystému Linuxového jádra

Matěj Grégr
mgregr@netx.as

Tomáš Podermański
tpoder@cesnet.cz

Cíle útočníka

- Vyčerpat (ucpat) kapacitu linky připojujícího serveru nebo sítě
  - Vysoký bit rate, relativně snesitelný packet rate

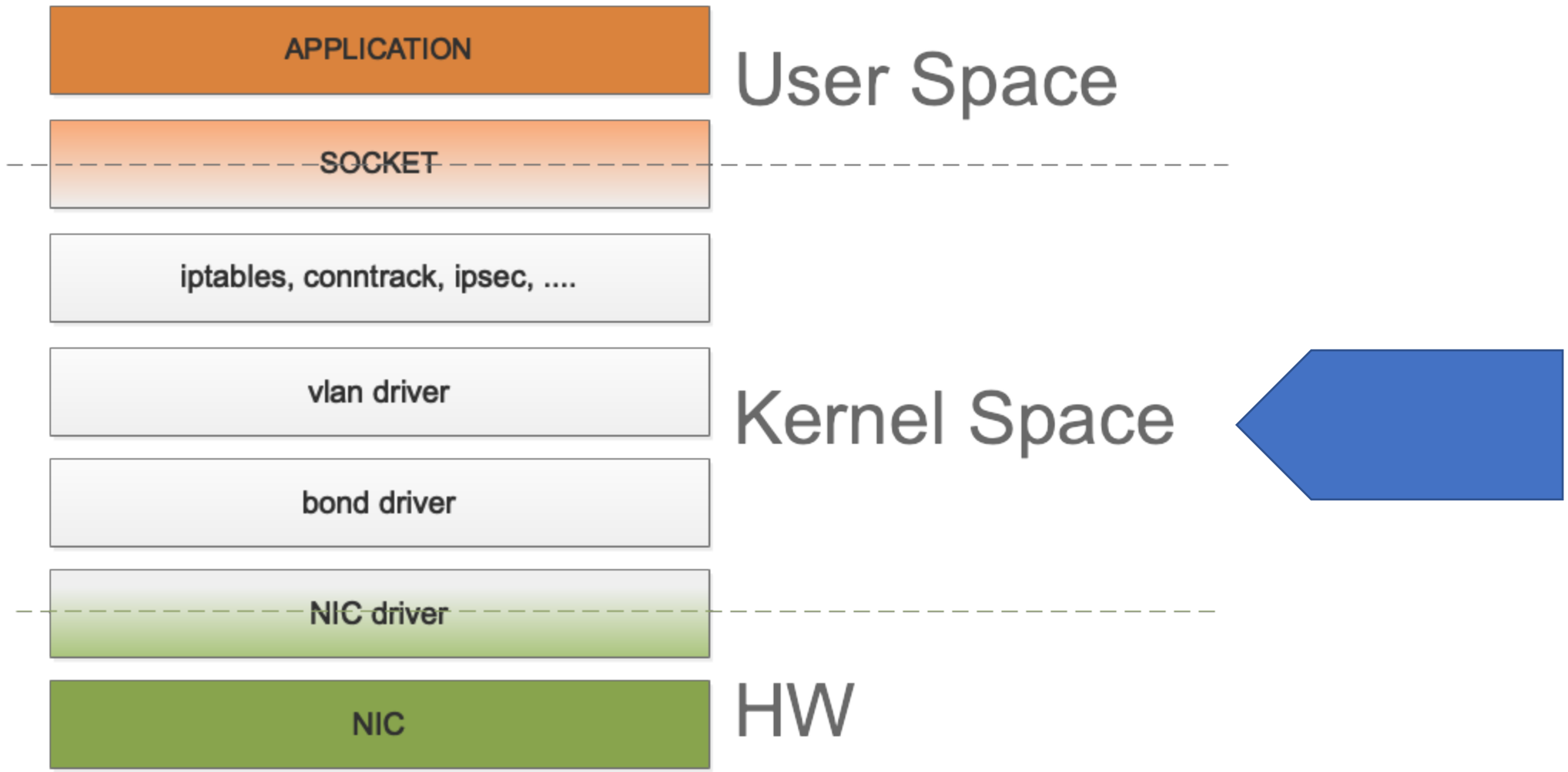
- Vyčerpat zdroje v cílovém systému a tím dosáhnout nefunkčnosti systému
  - “Vysoký“ packet rate, vysoký flow rate
  - Vyčerpání paměti
    - Donutím cílový systém aby alokoval paměť pro něco co nepotřebuje
  - Vyčerpání CPU
    - Donutím cílový systém aby se výpočetně zabýval něčím

Obecný (konvenční) postup

- “Ustát“ toho co nejvíce
  - Jenže …. limity, peníze, …

- Předřadit “něco“ co odolnost vůči útoku vyřeší za mně
  - Jenže …. limity, peníze, …

- Velké útoky je mnohdy nemožné zvládnout bez ztrát
  - Zpravidla hrajeme o to jak velké/malé ty ztráty budou


APPLICATION
User Space
SOCKET
iptables, conntrack, ipsec, ...
vlan driver
Kernel Space
bond driver
NIC driver
NIC
HW

Metodika meření

- Generátor provozu – Nx100Gb/s -> L2 switch -> meřené zařízení.
- Vypnuté šetřící režimy, snaha měřit vše v C0.
- Sledovaní zátěže CPU na měřeném zařízení, rozložení na jádra.
- Základní mez zátěže CPU 95% - horní mez stabilního systému.
- Mez zátěže CPU 100% - nestabilní systém.

- Měřeno nástrojem turbostat.

Sestava použitá v příkladech v této prezentaci

- 1x Intel(R) Xeon(R) CPU D-1587 @ 1.70GHz
- 16 (32) jader
- Systém RedHat/CentOS 7.8
- kernel: vanilla 5.4.53-1.2.el7.x86_64 #1 SMP x86_64 GNU/Linux
- driver: mlx5_core, version: 5.0-1.0.0.0
  firmware-version: 16.27.1016 (MT_0000000012)
- Hypethreading
- bond driver, vlan driver

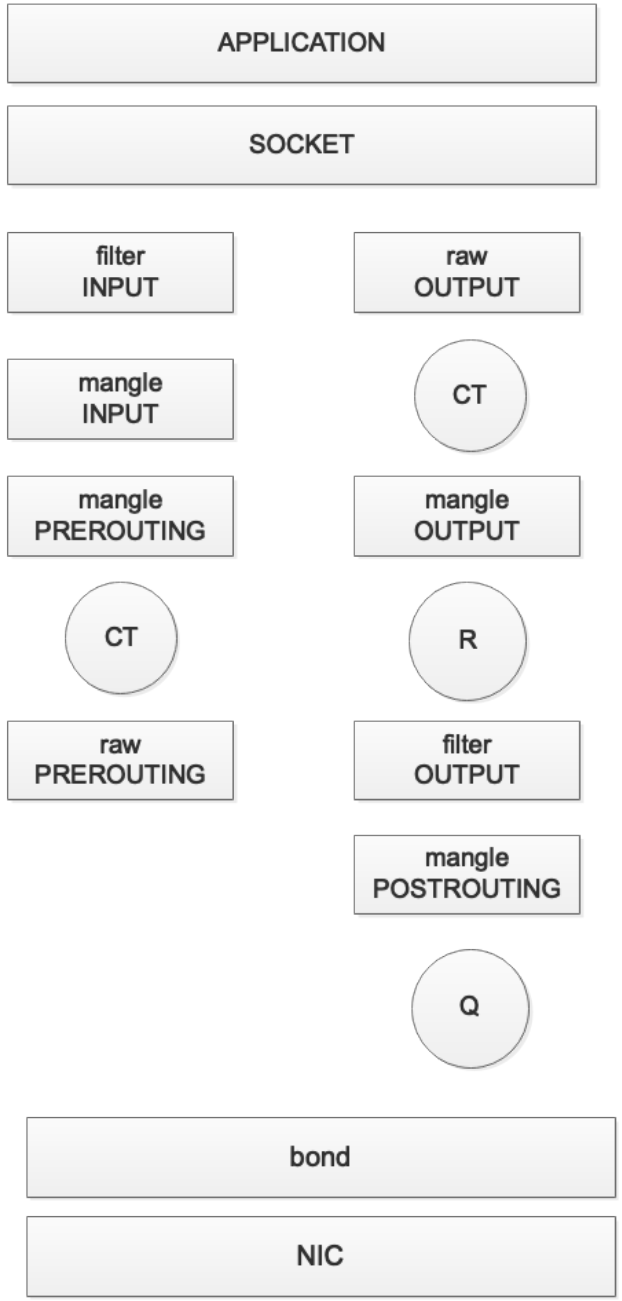
Jednotlivé typy příchozího provozu

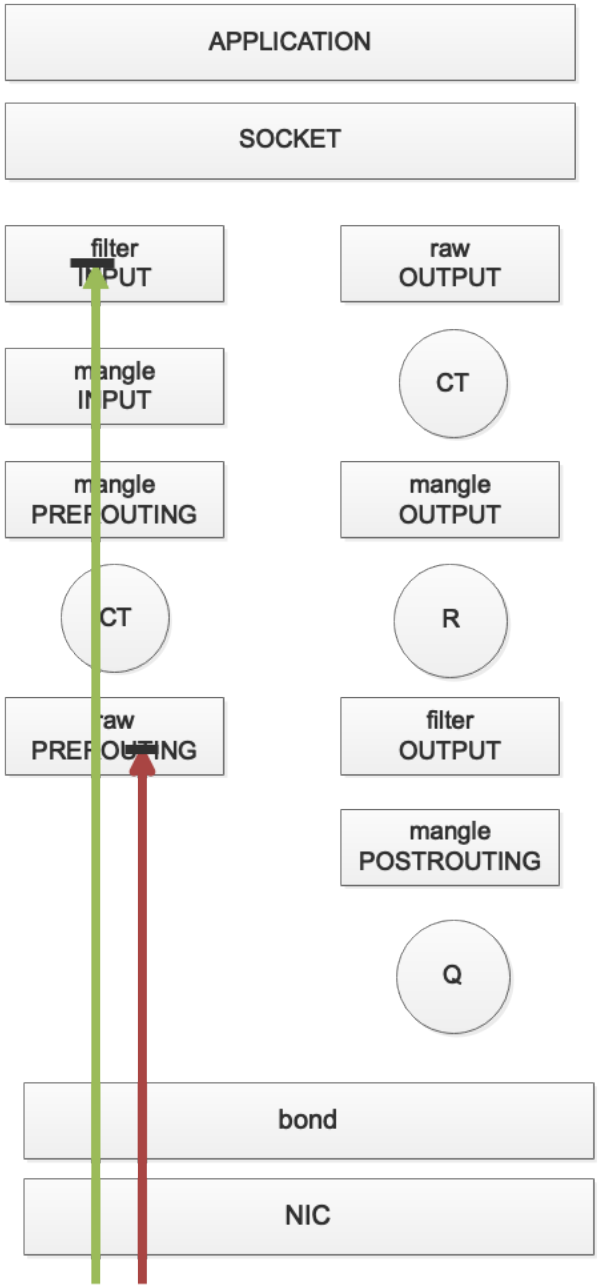
- Zbytný - cokoliv co nesouvisí s provozovanou službou
  - NTP, DNS, ICMP, … jakékoliv protokoly/porty (zpravidla kromě 80 a 443)
  - SYN-ACK směrem na adresu serveru
- Neidentifikovatelný zdrojem
  - Úvodní SYN, SYN-ACK komunikace
- Identifikovatelný zdrojem
  - Již po ustanoveném spojení TCP: ACK, FIN, RST, PSH, …

Zbytný provoz

- Filtrace přímo v infrastruktuře (ACL na směrovači, přepínači)
- Oddělení management provozu, připadně komunikace s backendem
- Specialita: SYN-ACK směrem na adresu serveru

- Když není možnost filtrace na úrovni HW serveru (v kartě) připadně raw table (u Linuxu).


APPLICATION
SOCKET
filter
INPUT
raw
OUTPUT
mangle
INPUT
CT
mangle
PREROUTING
mangle
OUTPUT
CT
R
raw
PREROUTING
filter
OUTPUT
mangle
POSTROUTING
Q
bond
NIC



Hodnoty v Mp/s	**CPU 95%**	**CPU 100%**
Prostý drop v FILTER table	5.6	5.9
Prostý drop v RAW table	27.6	28.7

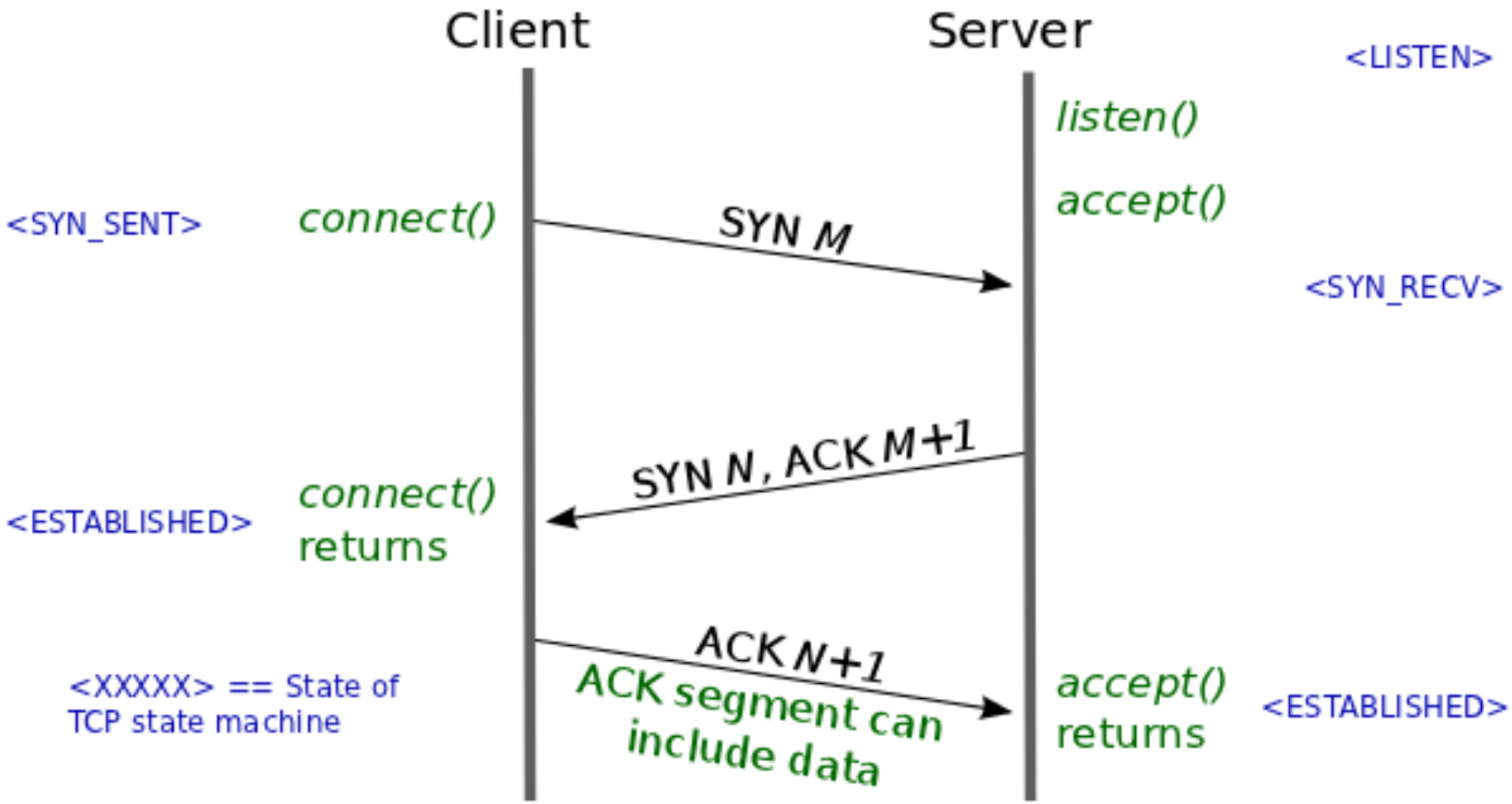
```
# iptables -t raw -I PREROUTING -i eth0 -p tcp \
       -m tcp --dport 80 -j ACCEPT
# iptables -t raw -I PREROUTING -i eth0 -j DROP
```

Provoz neidentifikovatelný zdrojem

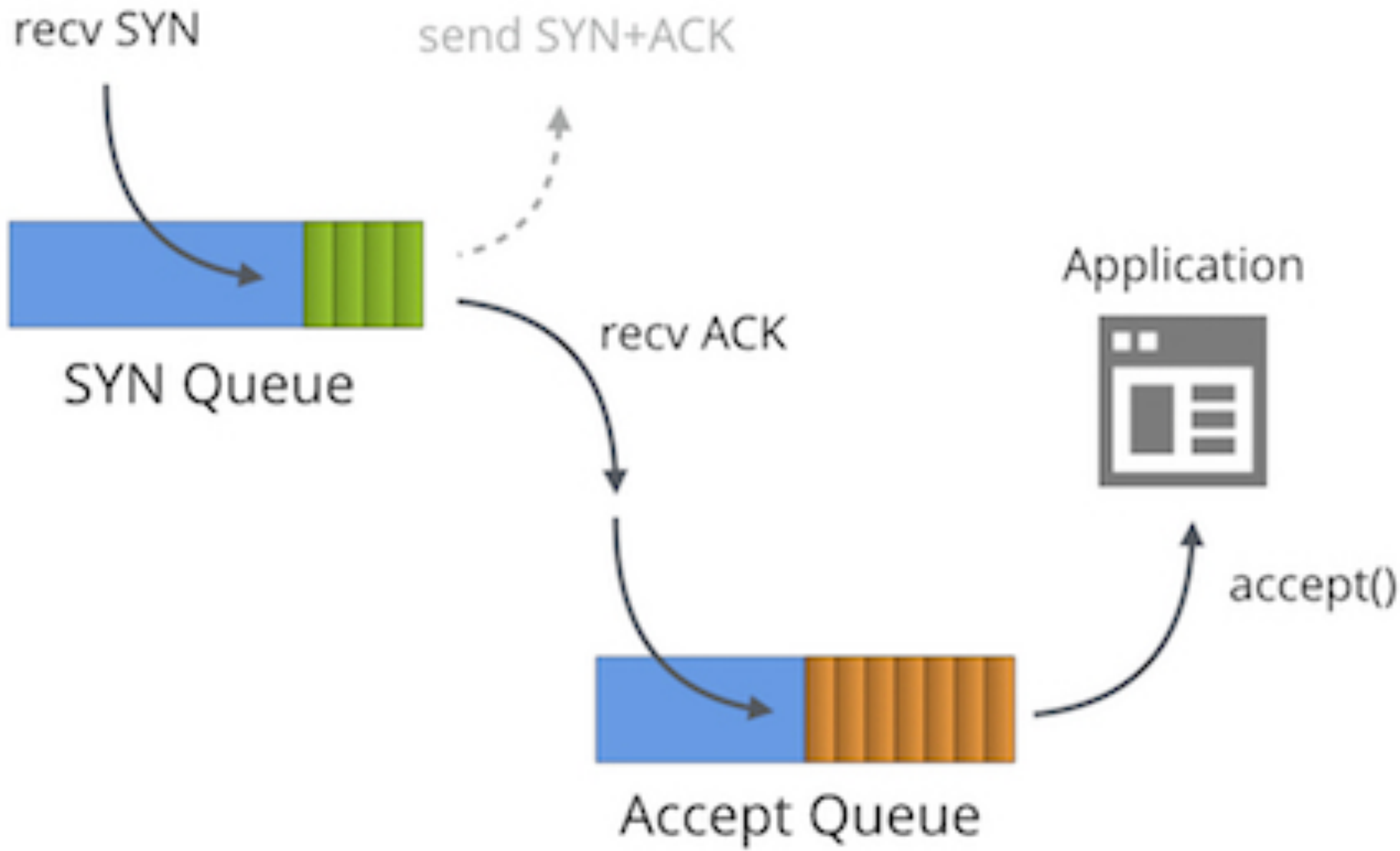
- Provoz u kterého je obtížné rozlišit zda zdrojová adresa odpovídá legitimnímu zdroji či nikoliv.

- Typický představitel: **úvodní SYN paket**
  - Nebezpečí vyčerpání zdrojů
  - Nelze na něj neodpovědět

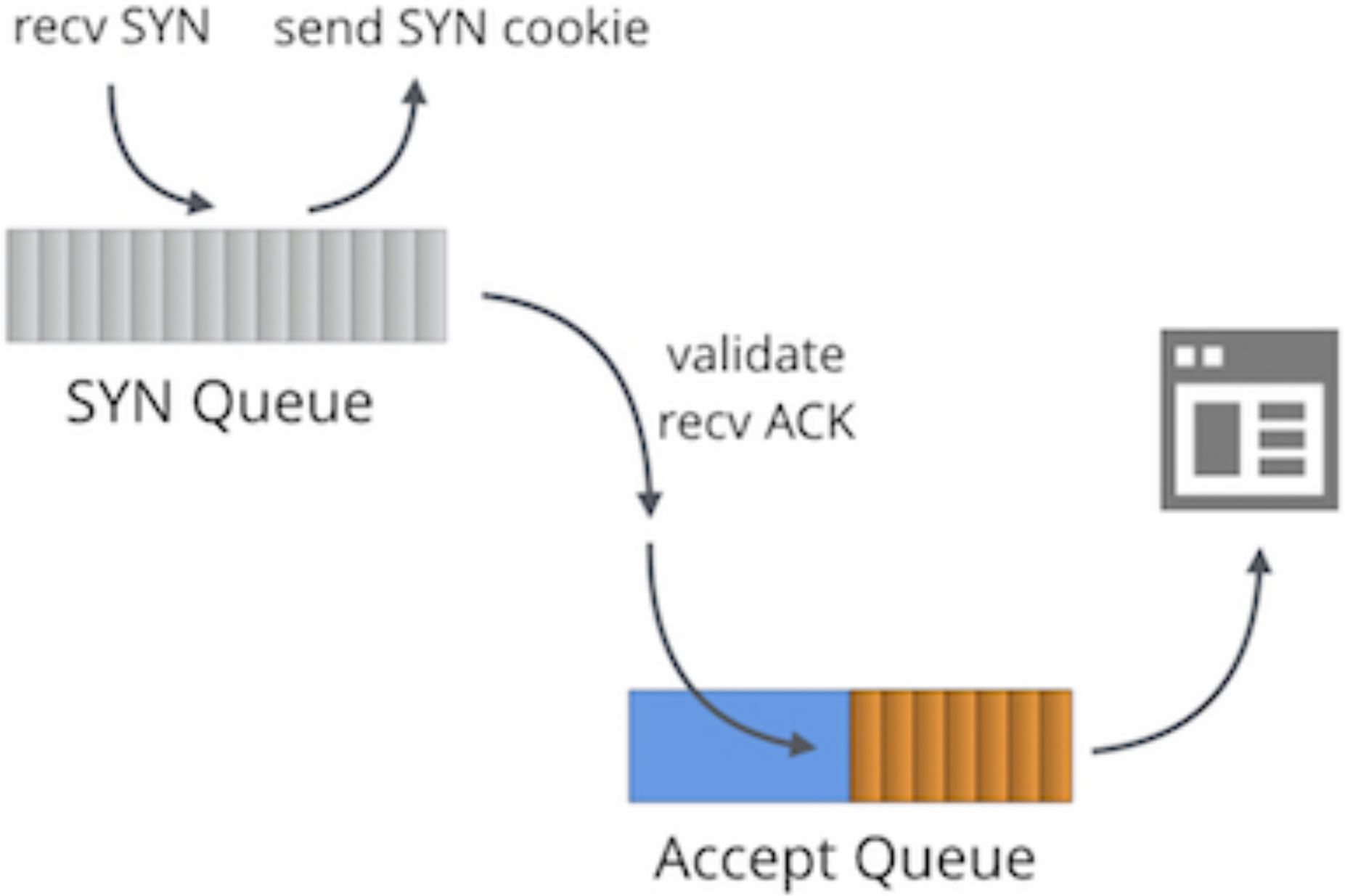
- Částečné řešení – syncookie
  - Přenesení stavové informace do hlavičky paketu -> server nemusí udržovat stav



Převzato z: https://lwn.net/Articles/508865/



Převzato z: https://blog.cloudflare.com/syn-packet-handling-in-the-wild/



Převzato z: https://blog.cloudflare.com/syn-packet-handling-in-the-wild/

Syncookie linux

```
aktivace při docházejících zdrojích
# sysctl net.ipv4.tcp_syncookies = 1

trvalá aktivace
# sysctl net.ipv4.tcp_syncookies = 2
```

- Obojí víceméně nepoužitelné
  - Zpracovává se až na vyšších vrstvách stacku
  - Nepodporuje IPv6

SYNPROXY

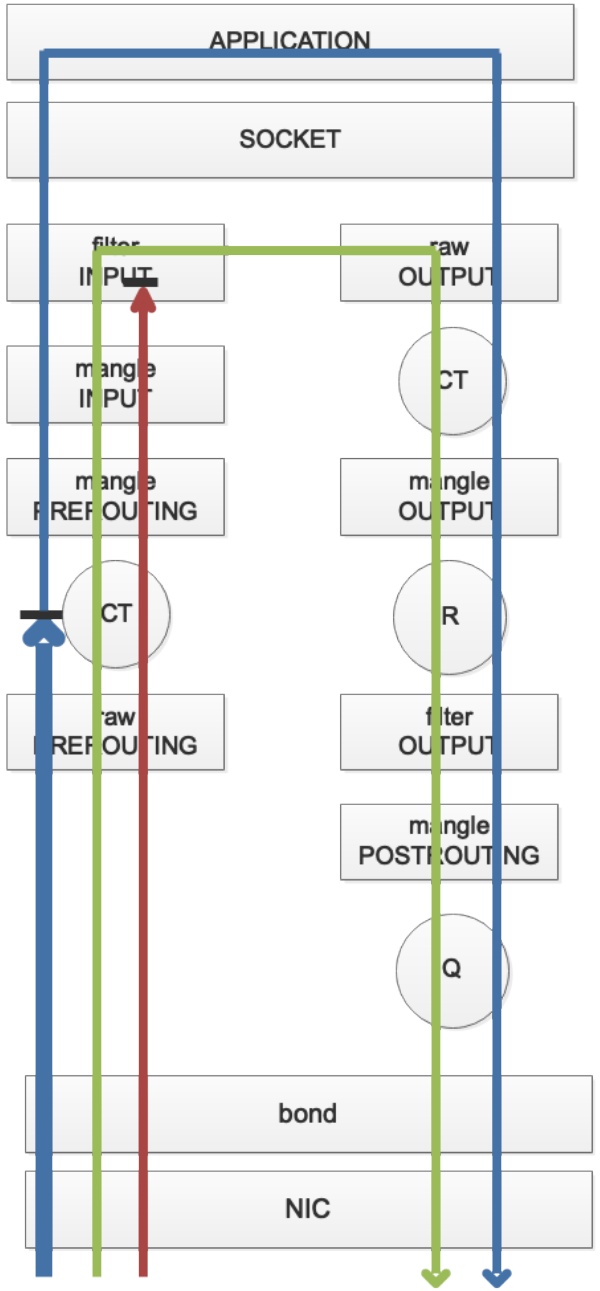
- Modul do iptables z dílny RH, nyní součást iptables

```
# iptables -t raw -I PREROUTING -i eth0 –
      -p tcp -m tcp --syn --dport 80 -j CT --notrack

# iptables -A INPUT -i eth0 -p tcp -m tcp /
      --dport 80 -m state --state INVALID,UNTRACKED /
      -j SYNPROXY --sack-perm --timestamp --wscale 7

# iptables -A INPUT -m state --state INVALID -j DROP
```

https://www.redhat.com/en/blog/mitigate-tcp-syn-flood-attacks-red-hat-enterprise-linux-7-beta



R:SYN S:SYN+ACK

R:SYN+ACK

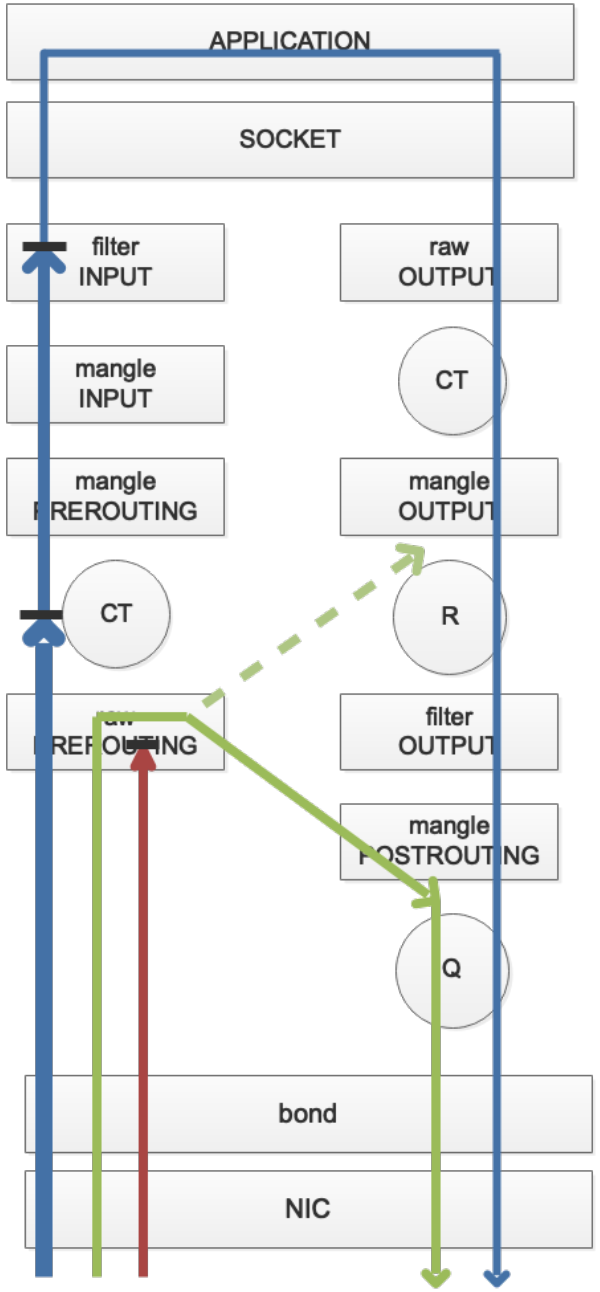
R:ACK, ACK+PUSH, FIN, (PUSH, RESET)

Hodnoty v Mp/s	**CPU 95%**	**CPU 100%**
Výchozí konfigurace	0.3	0.32
Živý soket bez iptables	3.6	3.9
Prostý drop v RAW table	27.6	28.7
Synproxy v doporučené konfiguraci	4.6	4.9

Optimalizace modulem rawcookie

- Modul posouvá reakci na SYN paket a výrobu SYNCOOKIE do nejnižší možné vrstvy kernelu (raw table v iptables).
- Ve variantě **DIRECT MODE** obchází routing system Linuxu (směrovací tabulku, arpy) a paket s odpovědí (SYN+ACK) zašle na MAC (nebo manuálně nakonfigurovanou) adresu směrovače odkud mu přisel úvodní SYN paket.
- Nárůst výkonu o 47% oproti SYNPROXY (z 4,6 Mp/s na 8,8 Mp/s)

https://github.com/netx-as/xt_RAWCOOKIE



R:SYN S:SYN+ACK

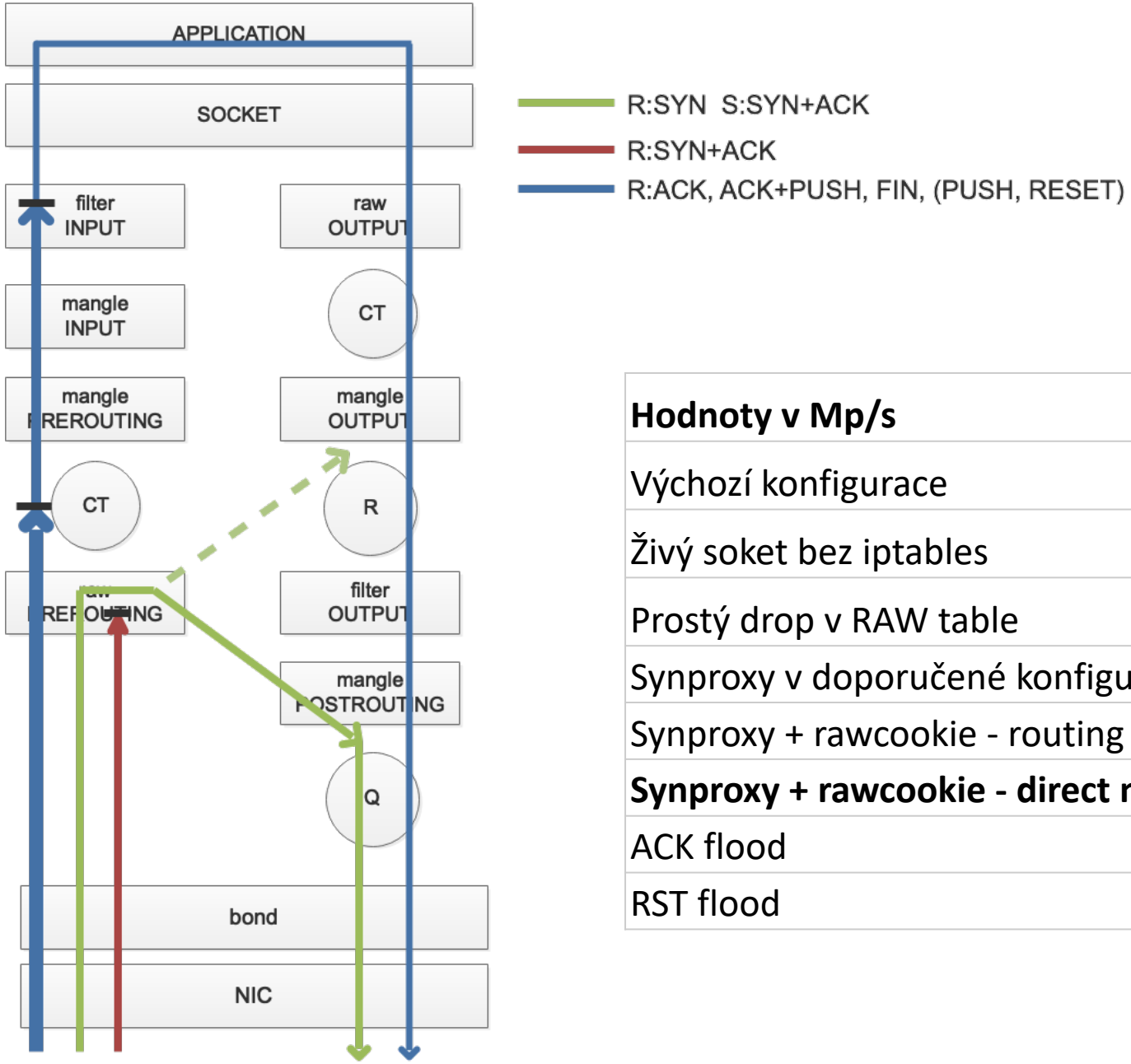
R:SYN+ACK

R:ACK, ACK+PUSH, FIN, (PUSH, RESET)

Hodnoty v Mp/s	**CPU 95%**	**CPU 100%**
Výchozí konfigurace	0.3	0.32
Živý soket bez iptables	3.6	3.9
Prostý drop v RAW table	27.6	28.7
Synproxy v doporučené konfiguraci	4.6	4.9
Synproxy + rawcookie - routing mode	5.2	5.5
Synproxy + rawcookie - direct mode	8.8	9.2

Jak dál

- Eliminace nového vytváření paketu
- Zachování zpracování SYN paketu na jednom CPU
- Přesun úvodní SYN,SYN+ACK akce do eBPF (ideálně do HW na kartu)
- Jenže… má to vůbec smysl… ?



Hodnoty v Mp/s	**CPU 95%**	**CPU 100%**
Výchozí konfigurace	0.3	0.32
Živý soket bez iptables	3.6	3.9
Prostý drop v RAW table	27.6	28.7
Synproxy v doporučené konfiguraci	4.6	4.9
Synproxy + rawcookie - routing mode	5.2	5.5
Synproxy + rawcookie - direct mode	8.8	9.2
ACK flood	7	7.9
RST flood	8.3	8.7

Provoz identifikovatelný zdrojem

- U tohoto provozu si můžeme byt jistí, že zdrojová IP adresa je autentická, je za ní skutečné komunikující zařízení.

- Mitigaci je nutné zvolit dle samotné aplikace
  - Vytváření blacklistů
  - Rate limiting per IP
  - …

Obecná doporučení

- Vhodnou konfiguraci serveru lze podstatně zvýšit odolnost serveru vůči útokům
- Reálné možnosti se pohybují v řádech vyšších jednotek Mp/s na běžně dostupném HW
- Pokud je to možné filtrovat zbytný provoz už v infrastruktuře
- Přednostně využívat blokaci v raw table, SYNPROXY modul případně doplnit o RAWCOOKIE