Metering and updating sets

```
nft add chain inet firewall incoming_dns

nft add set inet firewall dns_rate_meter '{ type ipv4_addr; size 64; flags timeout, dynamic; }'

nft add rule inet firewall incoming_dns udp dport 53 add @dns_rate_meter '{ ip saddr timeout 60s }' counter accept
nft add rule inet firewall incoming_dns counter log prefix 'dns-rate-meter-breach' group 4 counter drop

nft insert rule inet firewall input handle 5 udp dport 53 jump incoming_dns

sudo python3 -c 'from scapy.all import *; from ipaddress import IPv4Network; [sendp(Ether() / IP(src = str(addr), dst = "192.168.64.5") /
UDP(dport = 53) / DNS(rd = 1, qd = DNSQR(qname = "lame.ddos")), iface = "bridge100") for addr in IPv4Network("192.168.0.0/24").hosts()]'

nft list set inet firewall dns_rate_meter
table inet firewall {
   set dns_rate_meter {
      type ipv4_addr
      size 64
      flags dynamic,timeout
      elements = { 192.168.0.1 timeout 1m expires 57s576ms, 192.168.0.2 timeout 1m expires 57s580ms,
             192.168.0.3 timeout 1m expires 57s588ms, 192.168.0.4 timeout 1m expires 57s592ms,
             192.168.0.5 timeout 1m expires 57s600ms, 192.168.0.6 timeout 1m expires 57s604ms,
             192.168.0.7 timeout 1m expires 57s612ms, 192.168.0.8 timeout 1m expires 57s616ms,
             192.168.0.9 timeout 1m expires 57s624ms, 192.168.0.10 timeout 1m expires 57s628ms,
             …
             192.168.0.63 timeout 1m expires 57s936ms, 192.168.0.64 timeout 1m expires 57s944ms }
   }
}

nft list chain inet firewall incoming_dns
table inet firewall {
   chain incoming_dns {
      udp dport 53 add @dns_rate_meter { ip saddr timeout 1m } counter packets 192 bytes 10560 accept
      counter packets 570 bytes 31350 log prefix "dns-rate-meter-breach" group 4 counter packets 570 bytes 31350 drop
   }
}
```

Other nice features

- Glob matching interface names with iifname; (that may not exist when the rule set is first loaded) that are matched later (potentially slow, but useful for compatibility)

- iif uses the interface index ID and is resolved by interface name when the rules are loaded so the interface must already exist

- Can declare concatenated selectors as constraints for sets; useful for metering

- Queuing packets to user space (libnetfilter_queue)

```
nft insert rule inet firewall input iifname "wlp*" counter

nft insert rule inet firewall input iif enp0s1 counter

nft add set inet firewall concat_type '{ type ipv4_addr . inet_service; size 64; flags timeout, dynamic; }'

pip3 install --user NetfilterQueue
nft add rule inet firewall output tcp dport 80 queue num 2

python3 -c 'from netfilterqueue import NetfilterQueue; import sys; (lambda q: (q.bind(2, lambda pkt: (print(pkt), pkt.accept())), q.run(block = True)))( NetfilterQueue())'

TCP packet, 60 bytes
TCP packet, 52 bytes
TCP packet, 126 bytes
TCP packet, 52 bytes
TCP packet, 52 bytes
TCP packet, 52 bytes
```