UDP buffers tuning

Ciprian Cosma
Senior Systems Development Engineer
Amazon Web Services

UDP buffers tuning

Buffers too small:

- dropped packets
- cannot cope with spikes in traffic

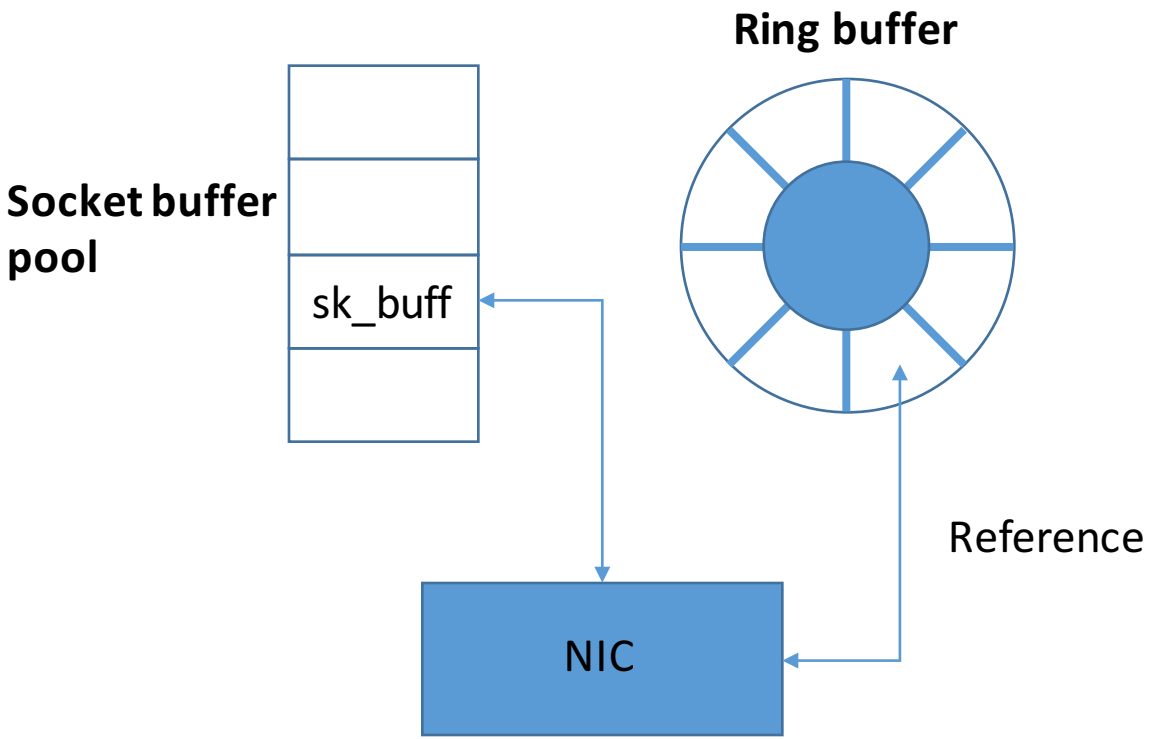
Buffers too large:

- increased delay
- slow to fail

Packets vs bytes:

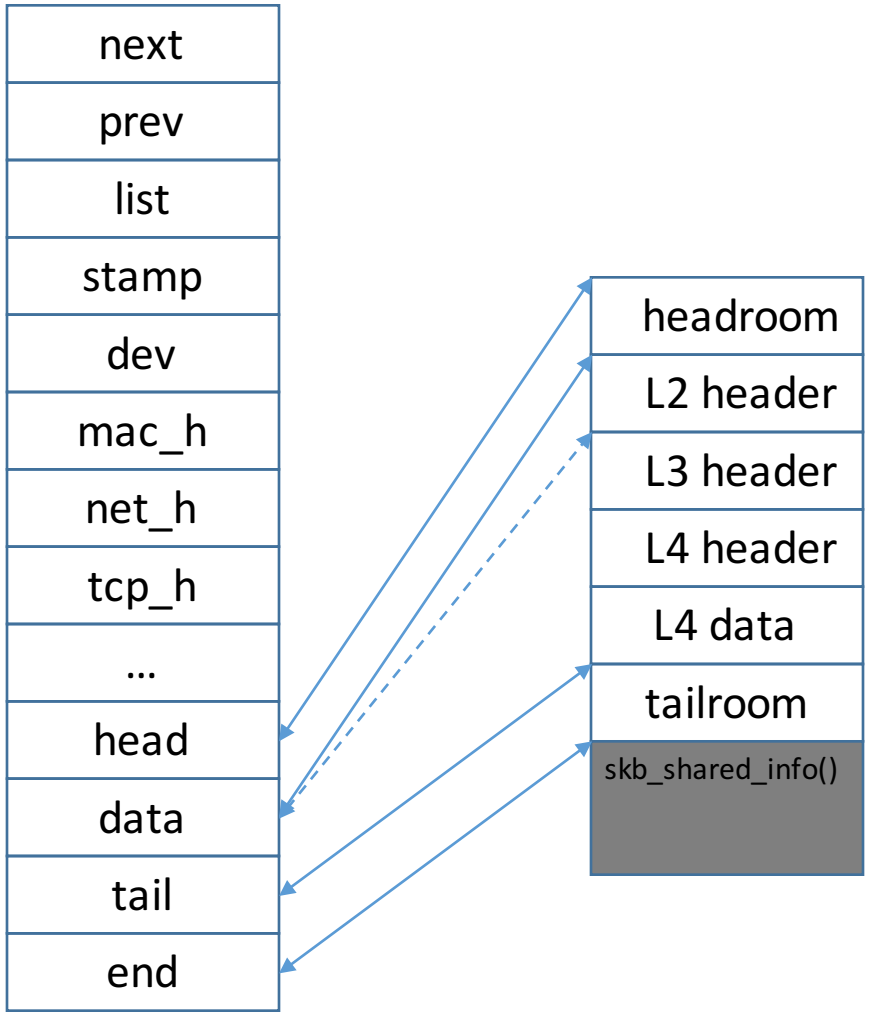
- QPS x mean packet size x scaling factor
- type and size of the request
- buffer allocation and accounting

Packet receive workflow



- NIC copies the packet using DMA
- socket buffers are slab allocated
- the kernel is notified via an interrupt

sk_buff structure



- most important networking structure
- contains a control structure and the actual packet
- as it moves up the stack the contents are not copied
- allocated space is larger than needed

Buffer accounting

- skb->truesize
  - sizeof(sb_buff)
  - sizeof(skb_shared_info)
  - packet data
- can be inspected in /proc/net/udp

```
sl local_address  rem_address  st tx_queue  rx_queue  tr tm->when  retrnsmt  uid timeout  inode ref pointer  drops
5709: 0100007F:0035  00000000:0000  07 00000000:00000300  00:00000000  00000000  0 0 14346 2 ffff8803f453b740  0
```

Buffer size

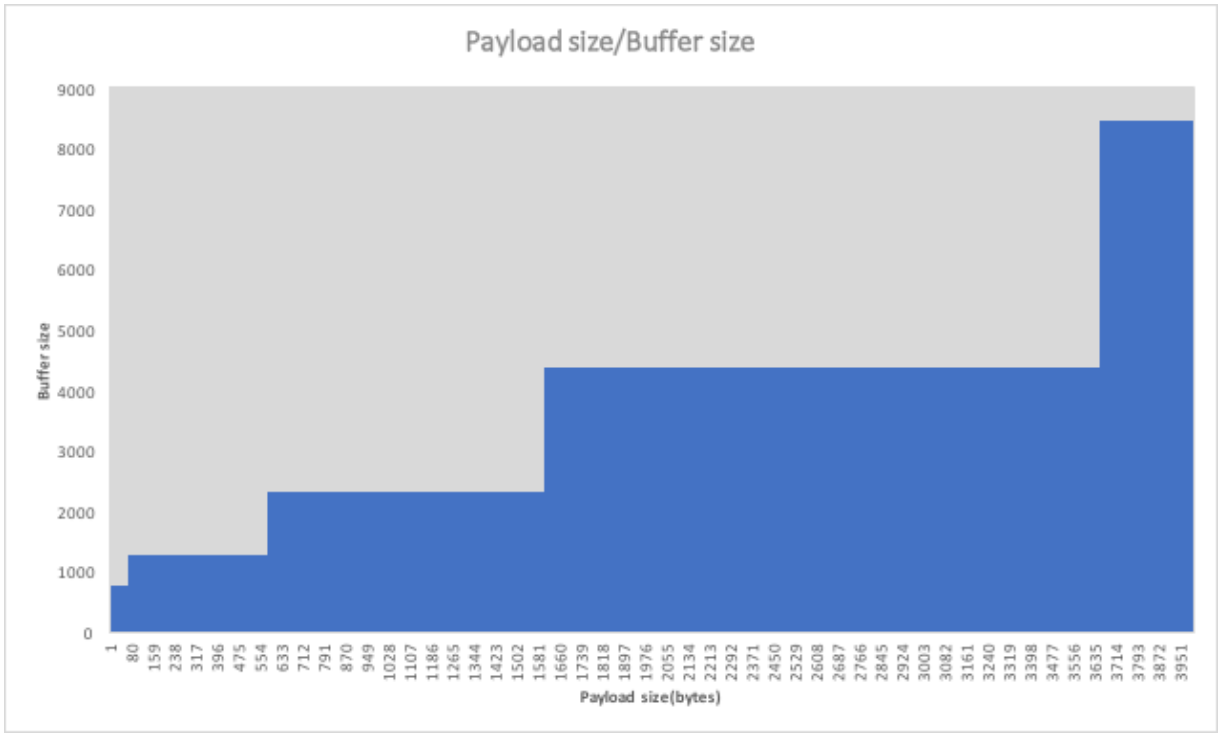
- Default values
  - `net.core.rmem_max`
  - `net.core.rmem_default`
  - `net.ipv4.udp_mem`
- Socket options
  - `setsockopt` - value will be doubled

Buffer size

```
output = subprocess.check_output(['/sbin/sysctl', 'net.core.rmem_default'])
print "default buffer size: {}".format(output)
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.bind(('localhost',5555))

buf = s.getsockopt(socket.SOL_SOCKET, socket.SO_RCVBUF)
print "current buffer (default config): {}".format(buf)

s.setsockopt(socket.SOL_SOCKET,socket.SO_RCVBUF, buf)
buf = s.getsockopt(socket.SOL_SOCKET, socket.SO_RCVBUF)
print "current buffer (setsockopt config): {}".format(buf)
```

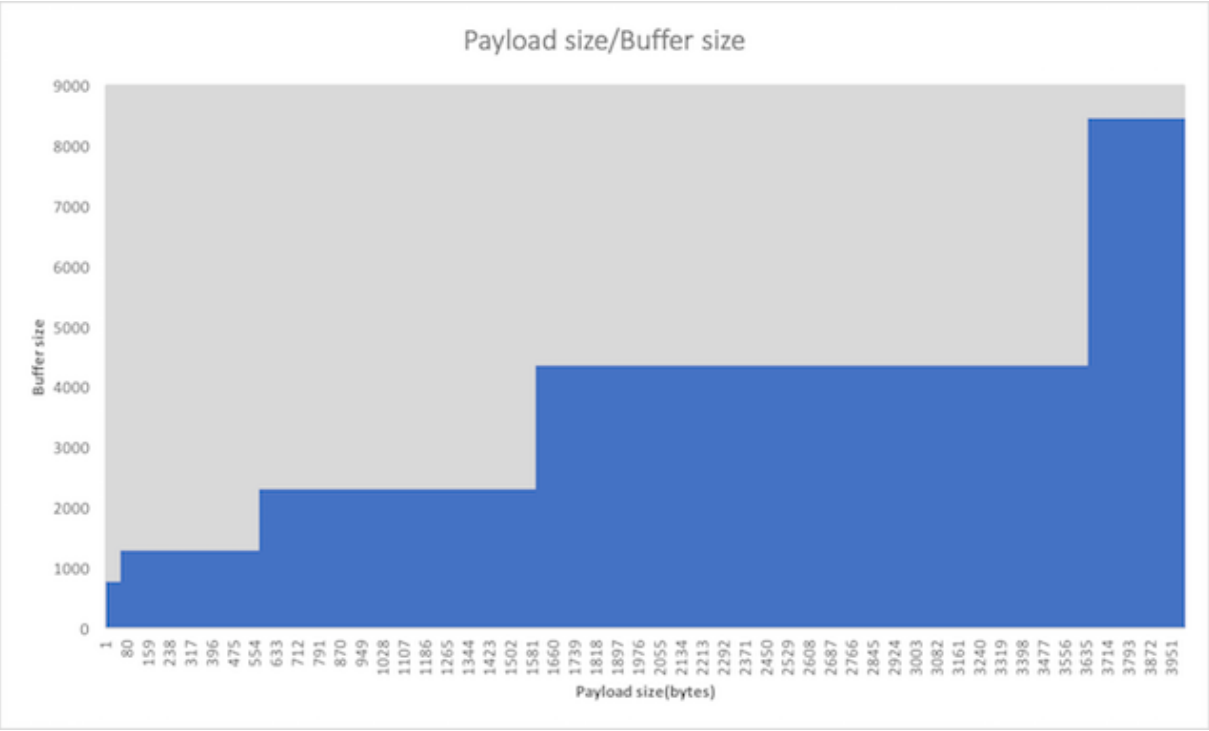
```
default buffer size: net.core.rmem_default = 1000000

current buffer (default config): 1000000
current buffer (setsockopt config): 2000000
```

Packet size vs. buffer size
IPv4



Payload size (bytes)	Buffer size (bytes)
1-69	768
70-581	1280
582-1605	2304
1606-3653	4352
3654-3999	8448

Packet size vs. buffer size
IPv6



Payload size (bytes)	Buffer size (bytes)
1-56	768
57-568	1280
569-1593	2304
1594-3640	4352
3641-3999	8448

Systemtap analysis

- probe on sock_queue_rcv_skb()
- inspect all packets based on socket address (taken from /proc/net/udp)
- size of sk_buff: 248
- size of skb_shared_info: 320
- data: 192

```
sizeof sk_buff: 248
skb_shared_info: 320
packet len: 52
linear or not (0-linear): 0
skb->truesize: 768
head: 0xffff88001194c600
data - head: 36
tail: 88
end: 192
```

Conclusions

- OS overhead is much larger than expected for typical query size values (20x vs 2x)
- a typical query (<50bytes payload) will require 768bytes of buffer space
- due to allocation strategies the buffers do not scale linearly with packet size

	1000 QPS	10k QPS	100k QPS	1M QPS
10 ms	7.5kB	75kB	750kB	7.32MB
100ms	75kB	750kB	7.32MB	73.24MB
1 s	750kB	7.32MB	73.24MB	732MB
10 s	7.32MB	73.24MB	732MB	7.15GB

Questions ?