Queen Mary

University of London

Arp Cache in large subnets

Christopher J. Walker
<C.J.Walker@qmul.ac.uk>

Overview

- The problem
- Arp
- Arp parameters
- Our solution

The problem

- Switch log: “Protocol control discards: arp-bcast or ipv6-nd packets are received at rate higher than 200pps,hence are discarded on queue 5!”
- TCPdump shows lots of arp packets on the network (~20/s)
- 700 devices on L2 subnet
  - 128 kept in cache

Arp

- Arp who has broadcast (IPv4)
  - IP address (L3) → ethernet MAC address (L2)
- Cached by Linux
  - “arp -a” to see cached entries
  - Linux replies for all IPs on the machine, not just those on this interface (by default)
- IPv6 does this differently

/proc/sys/net/ipv4/neigh/*/*

- https://linux.die.net/man/7/arp
- gc_interval (30s)
  - How frequently the garbage collector for neighbor entries should attempt to run.
- gc_stale_time (60s)
  - Determines how often to check for stale neighbor entries. When a neighbor entry is considered stale, it is resolved again before sending data to it.
- base_reachable_time_ms (30,000ms)
  - Once a neighbor has been found, the entry is considered to be valid for at least a random value between base_reachable_time/2 and 3*base_reachable_time/2. An entry's validity will be extended if it receives positive feedback from higher level protocols.

/proc/sys/net/ipv4/neigh/*/*

- gc_thresh1 (default 128)
  - The minimum number of entries to keep in the ARP cache. The garbage collector will not run if there are fewer than this number of entries in the cache.
- gc_thresh2 (default 512)
  - The soft maximum number of entries to keep in the ARP cache. The garbage collector will allow the number of entries to exceed this for 5 seconds before collection will be performed.
- gc_thresh3 (default 1024)
  - The hard maximum number of entries to keep in the ARP cache. The garbage collector will always run if there are more than this number of entries in the cache.

708 hosts on subnet

- gc_thresh(1,2,3)
  - We are garbage collecting down to 128 every 30s
    - Increase to cover size of cluster
- base_reachable_time_ms (default 30s)
  - Entries considered stale after 15-45s
    - Increase to 10 mins (5-15 mins)
      - 5 minute nagios checks
      - GPFS
      - Short enough to garbage collect stale entries after hardware replacement.

QMUL/CJW parameters

Parameter	Default	IBM	OCF	CJW	Rationale
gc_thresh1	128	808	4096	1024	> hosts
gc_thresh2	512	908	6144	2048	
gc_thresh3	1024	1008	8192	4096	
gc_interval	30	1,000,000,000		30	
gc_stale_time	60	2147483647	240	60	
base_reachable_time_ms	30,000	2147483647		600,000	10 mins (not 30s)

Conclusions

- Eliminated annoying log messages
- Increase performance
  - Reduced latency for new connections
  - Reduce broadcast traffic
  - Improvement difficult to measure