TCP and MTU in IPv6

Akira Kato

Keio Univ./WIDE Project
kato@wide.ad.jp

IPv6 Review

- **No fragmentation in the middle**
  - End-to-End fragmentation is supported
  - Should be avoided as much as possible
- **PMTUD to seek largest possible size**
  - May not always works well
- **Gives up PMTUD and use MTU 1280**
  - It is less efficient
  - But much better than loss of TCP connectivity
  - bind9 uses this option when possible

Larger Packet in IPv6

- ☆ **PMTU may not work well**
  - ICMP doesn’t return due to filtering
  - The TCP session results in break
- ☆ **TCP doesn’t handle IPv6 MTU issue**
  - At least in BSD variant OSs
  - TCP tries to send a larger segment (such as 1440)
  - IP6 divides the segment to multiple fragments
- ☆ **Fragments may be filtered out**
  - At a middle box or a router in the middle
  - Results in reassembling failure (ICMP is returned)
  - The TCP session results in break

Larger Packet in IPv6

☆ This issue is addressed in IETF 6man ML

- Jul 22, 2016 by Mark Andrews
- Jinmei and Dupont commented
- No explicit conclusion was given

☆ This is NOT specific to DNS

- Applicable to all cases
- Serious in Yeti Project
  - no fallback to IPv4 possible

Proposal

☆ IPV6_USE_MIN_MTU socket option (RFC3542)

- Specify to use 1280 MTU, no PMTUD performed

☆ TCP behavior if IPV6_USE_MINMTU=1

- Local MSS to be 1220 (1280-ip6/tcp header)
  - No local fragmentation is required
- Advertise MSS to be 1220
  - Remote site won’t fregment outgoing TCP segment
- MSS value from peer should clip to 1220
  - No local fragmentation is required

☆ IPV6_USE_MIN_MTU is the right knob?

- May be not, but no other knob is defined

Patch Files

- ☆ **Set of patches to NetBSD7 kernel developed**
  - netinet/tcp_input.c
  - netinet/tcp_output.c
  - netinet/tcp_subr.c
- ☆ **They may not be the best patches**
  - But anyway, it works
- ☆ **Reported to current-users list in Nov 7th**

Sample Patch to tcp_input

```
   sc->sc_ourmaxseg = tcp_mss_to_advertise(m->m_flags &
M_PKTHDR?m->m_pkthdr.rcvif : NULL, sc->sc_src.sa.sa_family);
+#ifdef INET6
+  if (tp && tp->t_in6pcb && tp->t_in6pcb->in6p_outputopts) {
+    if (tp->t_in6pcb->in6p_outputopts->ip6po_minmtu ==
+      IP6PO_MINMTU_ALL) {
+      sc->sc_ourmaxseg = min(sc->sc_ourmaxseg,
+        IPV6_MMTU - sizeof(struct ip6_hdr) - sizeof(struct tcphdr));
+      sc->sc_peermaxseg = min(sc->sc_peermaxseg,
+        IPV6_MMTU - sizeof(struct ip6_hdr) - sizeof(struct tcphdr));
+    }
+  }
+ #endif
   sc->sc_win = win;
```

Deployment in Yeti

☆ WIDE DM will be using the modified kernel

- Effective in earlier in the next week
- Report me if you see inconvenience