RTT matters

Matsuzaki ‘maz’ Yoshinobu
<maz@iij.ad.jp>
Round Trip Time (RTT)
TCP 3 way handshake and RTT
TCP and RTT

TCP data
TCP data
TCP data
TCP data
TCP ACK
TCP data
TCP data
TCP data
TCP ACK

window size = TCP RX buffer

RTT

RTT

RTT

Internet Initiative Japan Inc. (IIJ/AS2497)
packet loss and RTT

RTO
retransmission timeout

RTO is computed based on RTT -- see rfc6298
RTT

• the lower, the better
  – faster TCP handshake
  – better TCP throughput
  – faster TCP recovery from a packet loss

• There are various efforts to improve TCP performance even in a high latency situation
Wi-Fi is getting popular
I faced a trouble

- RTT was not stable from a wireless client to the server 2. ☹️
let’s try to isolate the problem

1) the local wifi network looks pretty stable

2) RTT to server 1 looks also stable

64 bytes time=3.757 ms
64 bytes time=3.789 ms
64 bytes time=3.768 ms
64 bytes time=3.718 ms
64 bytes time=3.644 ms
64 bytes time=3.762 ms

64 bytes time=169.588 ms
64 bytes time=170.666 ms
64 bytes time=169.793 ms
64 bytes time=168.185 ms
64 bytes time=169.783 ms
64 bytes time=169.957 ms
3) from the wired host in the same network, RTT to the server 2 looks stable
SO...

• this strange behavior happens only for this combination 😞
RTT distribution to server 1

from wired host to server 1

from wifi host to server 1

Internet Initiative Japan Inc. (IIJ/AS2497)
RTT distribution to server 2

wired

from wired host to server 2

wifi

from wifi host to server 2
the wifi AP was buffering packets

• and this caused the unstable RTT 😞
My wifi adapter does sleep

• to reduce battery usage
• before sleeping, the client send a notification to the wifi AP, and the AP keeps packets until the client wake up

• so, my PC was asking the buffering!
wifi AP sends beacon

• beacon interval
  – time interval between beacon transmissions
  – usually 100msec, but it’s configurable

• TIM (Traffic Indication Map)
  – to tell any sleeping clients if the AP has any buffered frames present for it

• wifi adapter can sleep between beacons, and wake up to check a beacon (TIM can indicate if the adapter need to receive data or not)
the scenario

• My wifi adapter went to sleep after 200msec of no traffic
  – that’s why the unstable RTT happens only when I was communicating with server 2 (average RTT is 300msec)

• Based on the beacon interval information (which was 100msec in my case), it woke up and received a response
  – that’s why most RTT distribution is within 100msec
sleeping and buffering

200msec no traffic

goes to sleep until the next beacon

buffered
Summary

• Strange RTT behavior happens if your communication is between:
  – a host connected to a wifi network and
  – a far end host (RTT>200msec)

• Your wifi adapter goes to sleep
  – “200msec of no traffic” seems a common trigger

• The sleep duration is manageable by setting beacon interval of your wifi AP
  – 100msec would be reasonable
  – You might be able to reduce battery usage by setting it as 1000msec, but this could introduce more RTT penalty