

TCP-BIC

M11015104 戴天佑

1. BIC 背景：

在 TCP-Reno 中進入 congestion avoidance 或是 fast recovery 後每經過一個 RTT， $cwnd$ 才會+1，假設連線狀況良好但是 RTT 很長的話，就需要花很長的時間才能找到最好的 $cwnd$ ，因此有了 BIC(Binary Increase Congestion control)，其核心的思想就是通過二分搜索來找到當前最適合的 $cwnd$ 。

因此相較於 TCP-Reno，BIC 更適合在高 bandwidth 的傳輸上，不過在下面的原理中也有講到，如果傳輸量太低的話反而使用 TCP-Reno 較有效率。

2. BIC 原理：

S_{max} ：window 的最大增加量。

S_{min} ：window 的最小減少量。

β ：減少 window 的 multiplier factor。

default_max_win：default 最大 window。

max_win：maximum window(封包遺失時當下的 $cwnd$)。

min_win：minimum window(封包遺失時當下的 $cwnd \times \beta$)。

pre_win：上一個 maximum window(即上一個 max_win)。

target_win：max_win 和 min_win 的中間點(預測的 $cwnd$)。

Binary search increase：

loop

當封包遺失時：

max_win= $cwnd$

min_win= $cwnd \times \beta$

target_win=(max_win+min_win)/2

但是當 max_win 較大時，可能會造成傳輸的抖動，因此：

If (target_win - $cwnd$) > S_{max} ：

$cwnd$ 只有一個增加率 = S_{max}

若封包沒遺失：

(max_win 值不變)

min_win= $cwnd$

target_win=(max_win+min_win)/2

repeat until (target_win - $cwnd$) < S_{min}

Slow start：

If $cwnd > max_win$

每個 RTT $\text{target_win} = \text{cwnd} + 1, \text{cwnd} + 2, \text{cwnd} + 4, \dots$

Until $\text{cwnd} + S_{\max}$

接著進入 Binary search increase 執行封包沒遺失時的運算

Fast convergence :

為了確保不同的 TCP flows 有相同的收斂速度因此：

If 封包遺失且 cwnd 有下降趨勢：

$\text{target_win} = (\text{max_win} + \text{min_win}) / 2$

If 封包遺失且 cwnd 有上升趨勢：

$\text{target_win} = \text{max_win}$

Reno :

在 bandwidth 小的情況下使用 BIC 並沒有優勢，所以 BIC 規定如果 cwnd 小於 threshold 則使用 Reno

3. 基於 ns3.33/examples/tcp/tcp-variants-comparison.cc

加入以下的 code (NEW! 為新加的)

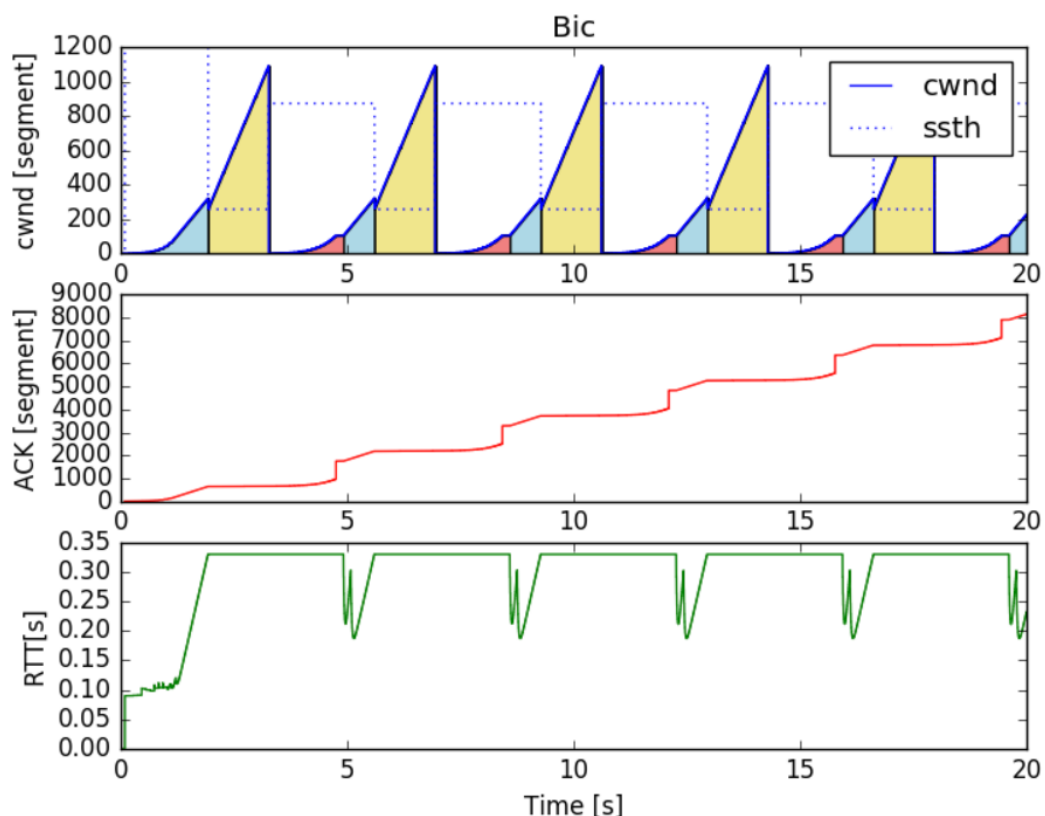
```
67 static Ptr<OutputStreamWrapper> nextRxStream;
68 static Ptr<OutputStreamWrapper> inFlightStream; // NEW!
69 static Ptr<OutputStreamWrapper> ackStream; // NEW!
70 static Ptr<OutputStreamWrapper> congStateStream;

206 //NEW!
207 static void
208 TraceAck (std::string &ack_file_name)
209 {
210     AsciiTraceHelper ascii;
211     ackStream = ascii.CreateFileStream (ack_file_name.c_str ());
212     Config::ConnectWithoutContext ("/NodeList/1/$ns3::TcpL4Protocol/SocketList/0/HighestRxAck", MakeCallback (&AckTracer));
213 }
214
215 //NEW!
216 static void
217 TraceCongState (std::string &cong_state_file_name)
218 {
219     AsciiTraceHelper ascii;
220     congStateStream = ascii.CreateFileStream (cong_state_file_name.c_str ());
221     Config::ConnectWithoutContext ("/NodeList/1/$ns3::TcpL4Protocol/SocketList/0/CongState", MakeCallback (&CongStateTracer));
222 }
223
224 ---
225 //NEW!
226 static void
227 TraceAck (std::string &ack_file_name)
228 {
229     AsciiTraceHelper ascii;
230     ackStream = ascii.CreateFileStream (ack_file_name.c_str ());
231     Config::ConnectWithoutContext ("/NodeList/1/$ns3::TcpL4Protocol/SocketList/0/HighestRxAck", MakeCallback (&AckTracer));
232 }
233
234 ---
235 //NEW!
236 static void
237 TraceCongState (std::string &cong_state_file_name)
238 {
239     AsciiTraceHelper ascii;
240     congStateStream = ascii.CreateFileStream (cong_state_file_name.c_str ());
241     Config::ConnectWithoutContext ("/NodeList/1/$ns3::TcpL4Protocol/SocketList/0/CongState", MakeCallback (&CongStateTracer));
242 }
243
244 ---
459 Simulator::Schedule (Seconds (0.1), &TraceNextTx, prefix_file_name + "-next-tx.data");
460 Simulator::Schedule (Seconds (0.00001), &TraceAck, prefix_file_name + "-ack.data"); // NEW!
461 Simulator::Schedule (Seconds (0.00001), &TraceCongState, prefix_file_name + "-cong-state.data"); // NEW!
462 }
```

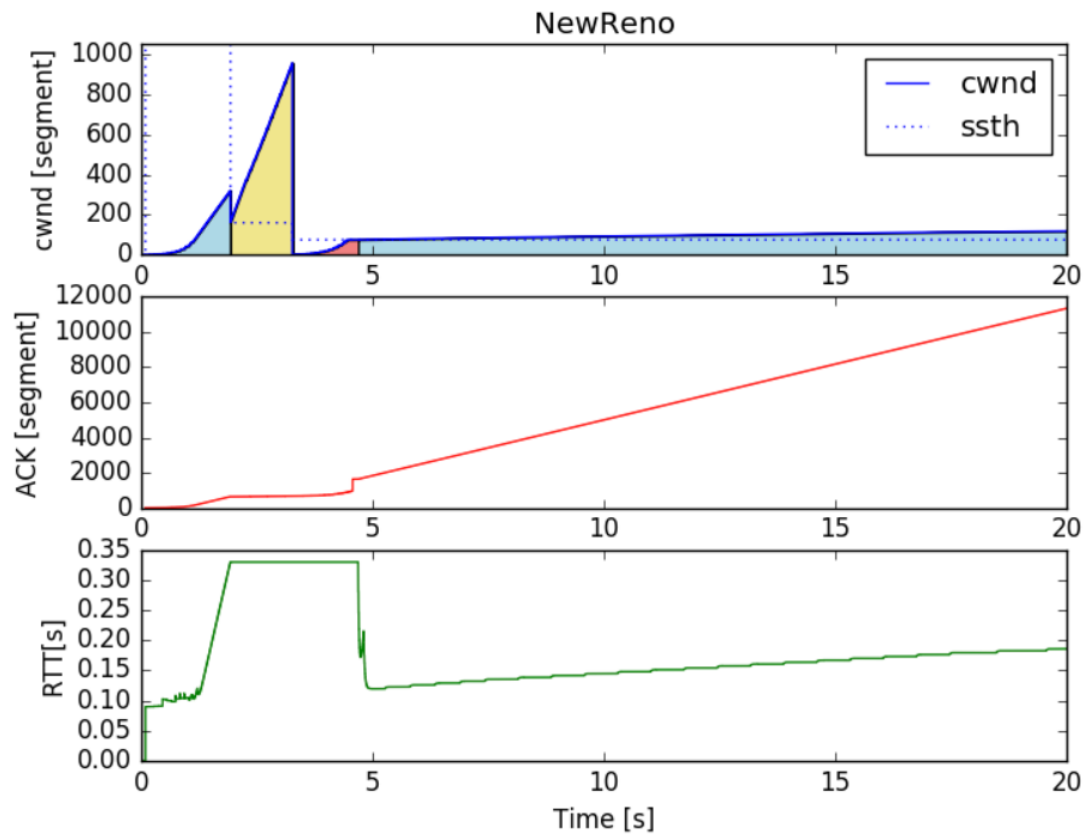
Run

```
ns-3.33 -- -zsh -- 80x24
[2026/2041] Creating pcfile: src/olsr/libns3.33-olsr-debug.pc
[2027/2041] Creating pcfile: src/point-to-point/libns3.33-point-to-point-debug.p
c
[2028/2041] Creating pcfile: src/point-to-point-layout/libns3.33-point-to-point-
layout-debug.pc
[2029/2041] Creating pcfile: src/propagation/libns3.33-propagation-debug.pc
[2030/2041] Creating pcfile: src/sixlowpan/libns3.33-sixlowpan-debug.pc
[2031/2041] Creating pcfile: src/spectrum/libns3.33-spectrum-debug.pc
[2032/2041] Creating pcfile: src/stats/libns3.33-stats-debug.pc
[2033/2041] Creating pcfile: src/tcp-client/libns3.33-tcp-client-debug.pc
[2034/2041] Creating pcfile: src/test/libns3.33-test-debug.pc
[2035/2041] Creating pcfile: src/topology-read/libns3.33-topology-read-debug.pc
[2036/2041] Creating pcfile: src/traffic-control/libns3.33-traffic-control-debug
.pc
[2037/2041] Creating pcfile: src/uan/libns3.33-uan-debug.pc
[2038/2041] Creating pcfile: src/virtual-net-device/libns3.33-virtual-net-device
-debug.pc
[2039/2041] Creating pcfile: src/wave/libns3.33-wave-debug.pc
[2040/2041] Creating pcfile: src/wifi/libns3.33-wifi-debug.pc
[2041/2041] Creating pcfile: src/wimax/libns3.33-wimax-debug.pc
Waf: Leaving directory `/Users/leoc99cc/ns-allinone-3.33/ns-3.33/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (3m9.598s)
[(base) leoc99cc@daitianyoudeMacBook-Pro ns-3.33 % mkdir data]
```

4. 使用 python 畫出圖表



因 tcp-variants-comparison.cc 中已有 TCP-NewReno 順便再輸出一個 NewReno 做比較



Reference

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6394438>