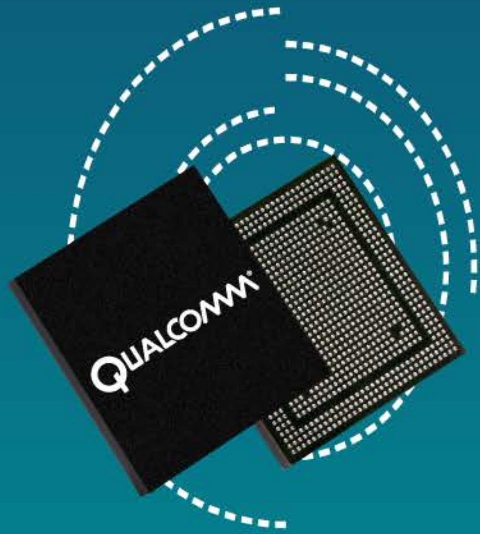
Qualcomm
TECHNOLOGIES, INC



Data Throughput Troubleshooting
数据吞吐量问题分析

80-ND934-2 B

Qualcomm Technologies, Inc. 5775 Morehouse Drive San Diego, CA 92121 U.S.A.
高通技术股份有限公司，美国加利福尼亚州圣地亚哥市莫豪斯路 **5775** 号，邮编 **92121**

Revision History

Revision	Date	Description
A	Dec 2013	Initial release
B	Mar 2014	Update the content of Chapter 2, 5, 7 and Appendix.

Note: There is no Rev. I, O, Q, S, X, or Z per Mil. standards.

Contents

1. Preparations and Basic Steps

Preparations

Prepare the following analysis tools:

- QXDM

  The QXDM is used to play back QXDM LOG to observe View of the physical layer, so as to acquire the data rate in the physical layer. For example, the observations of LTE:

  - LTE ML1 DL Throughput and BLER
  - LTE ML1 UL Throughput and BLER

- Wireshark

  The WIRESHARK LOG helps to acquire the data rate in the TCP/IP layer and to learn sending/receiving status of the data packet.

  - WIRESHARK LOG on the PC side (recommended)
  - WIRESHARK LOG on the UE side

- DU Meter

  The DU Meter installed on the PC helps to observe the data rate of PC Network Adapter.

Troubleshooting Steps

1. Define the target value of throughput, compare with the value acquired by the commercial reference device (REF) under the same test environment, so as to narrow down the scope of problematic modules.

2. View the application layer (HTTP/TCP/IP) first. Use Wireshark to check whether the TCP/IP layer throughput meets requirements, whether the TCP message loses packets, and whether retransmission occurs frequently.

3. Check whether flow control is started frequently or whether packet loss occurs in the Data Service layer if packet loss or retransmission often occurs in the TCP layer. If not, view the lower layer (RLC/MAC/PH) to check the channel quality, bit error rate (BER), and network scheduling data.

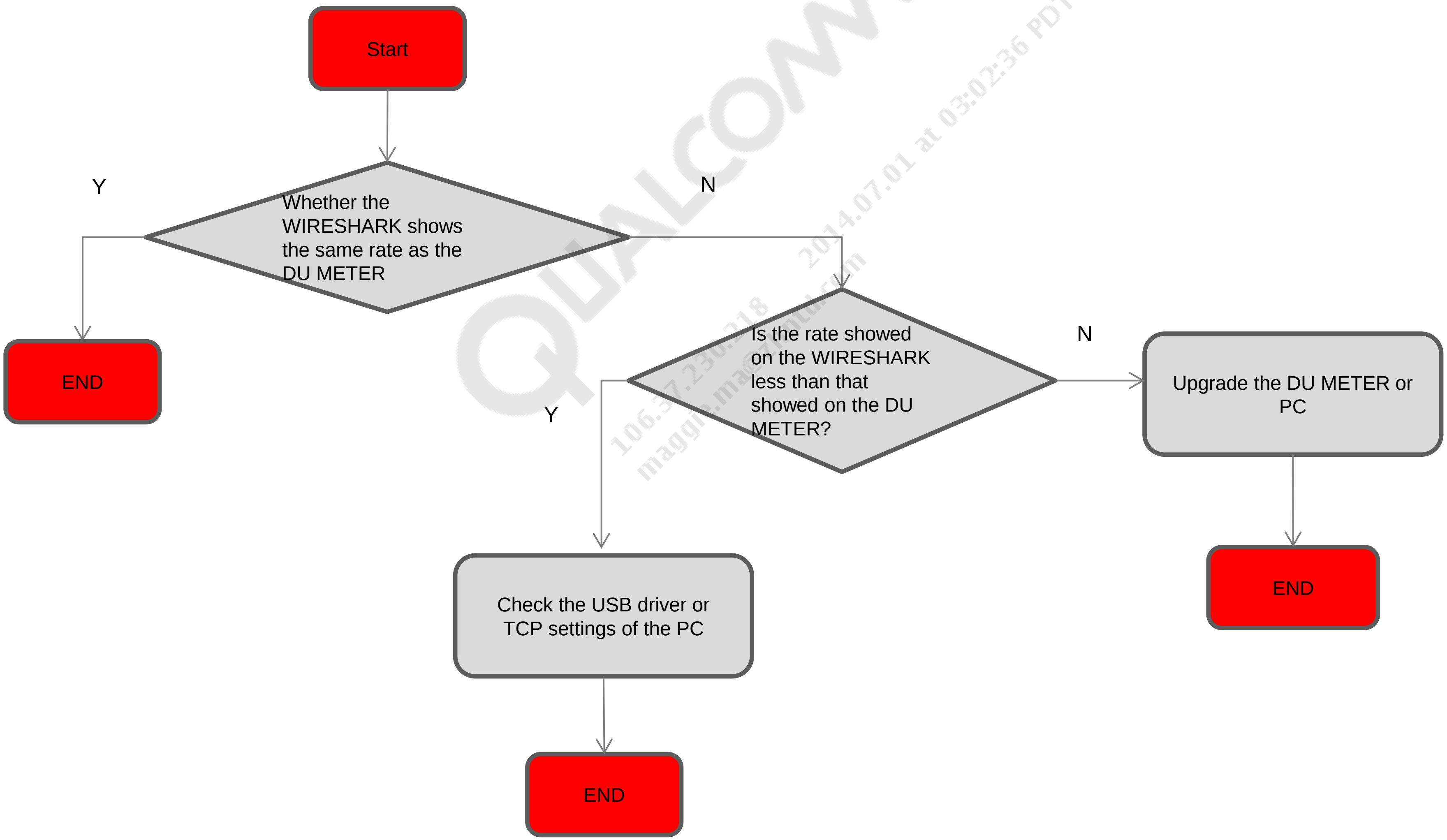
4. Check the RTT delay over ping packet (use the DUT and the REF respectively) if necessary, to figure out whether the DUT has a larger delay than the REF and whether the ping packet timeout occurs. Checking the ping packet (ICMP) problem is relatively easy, compared with TCP.

5. Use the iperf tool to perform the UDP test in case of insufficient bandwidth of the lower layer to locate the problem.

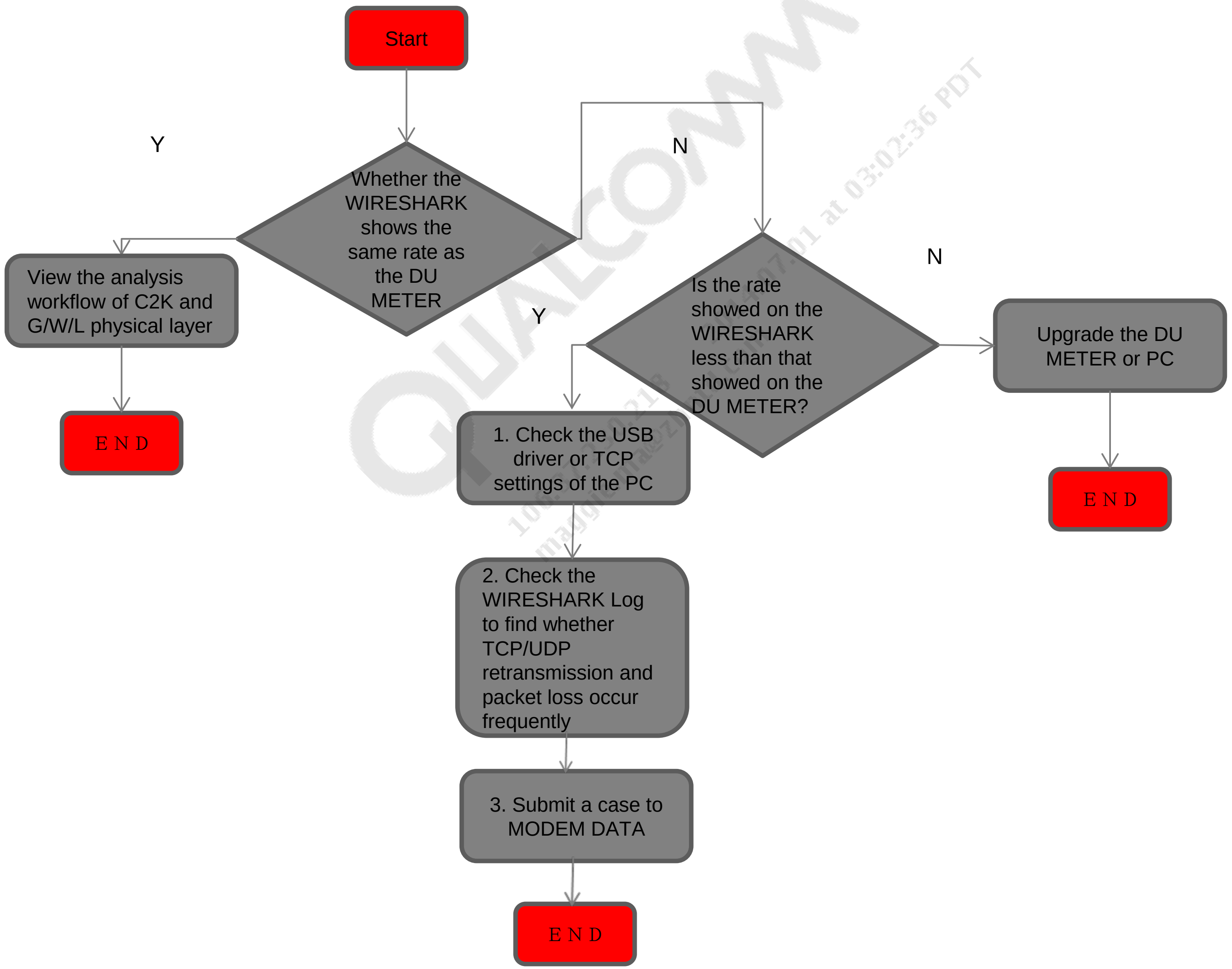
2. Data Service Layer Analysis

2.1 Wireshark Analysis

Workflow of Checking Log on the PC/Android Side with Wireshark

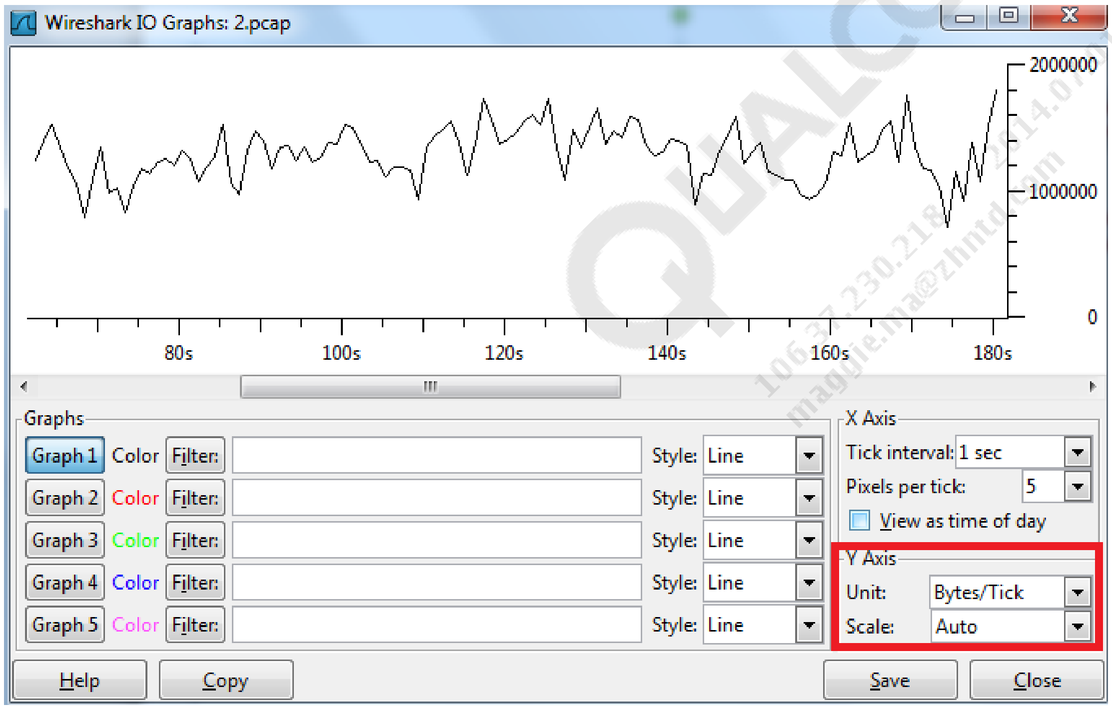


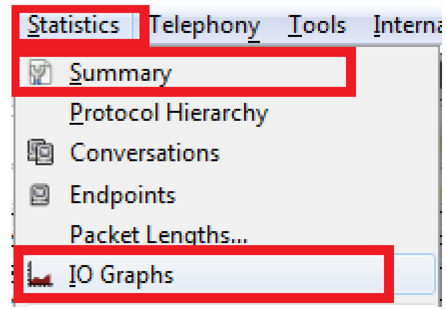
Workflow of Checking Log on the UE Side with Wireshark



Analyzing Message with Wireshark

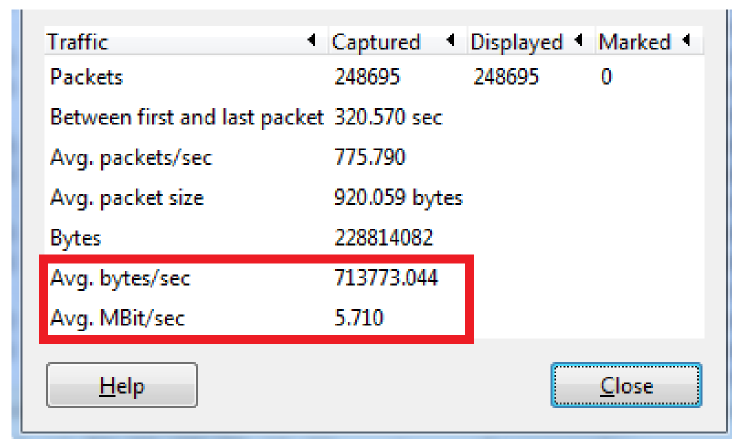
- Open the pcap log with Wireshark.
- Select Statistics -> IO Graphs (or Statistics -> Summary).
- Acquire data throughput status such as rate and fluctuation.





IO Graphs

Summary



Applications of IO Graphs

- Users can apply several filter criteria to draw different curves for comparison. For example, the following conclusions can be drawn from a comparison of the figures below:
  - Black curve: The throughput fluctuation ratio in the figure above (WiFi) is greater than that in the figure below (USB)
  - Red curve: The number of duplicated ACKs in the figure above (WiFi) is more than that in the figure below (USB)

Case Study of TCP Packet Log

- Check TCP data packet loss

2840	21:30:34.211	90.130.66.198	192.168.0.153	TCP	103	ftp > 63066 [PSH, ACK] Seq=293 Ack=83 Win=32120 Len=49
2841	21:30:34.212	192.168.0.153	90.130.66.198	TCP	60	63066 > ftp [PSH, ACK] Seq=83 Ack=342 Win=64979 Len=6
2879	21:30:34.531	90.130.66.198	192.168.0.153	TCP	54	[TCP Dup ACK 2840#1] ftp > 63066 [ACK] Seq=342 Ack=83 Win=32120 Len=0
2884	21:30:34.621	90.130.66.198	192.168.0.153	TCP	54	ftp > 63066 [ACK] Seq=342 Ack=89 Win=32114 Len=0
2892	21:30:35.472	90.130.66.198	192.168.0.153	TCP	78	[TCP Previous segment lost] ftp > 63066 [PSH, ACK] Seq=381 Ack=89 Win=32120 Len=24
2893	21:30:35.472	192.168.0.153	90.130.66.198	TCP	54	[TCP Dup ACK 2841#1] 63066 > ftp [ACK] Seq=89 Ack=342 Win=64979 Len=0
2925	21:30:36.217	90.130.66.198	192.168.0.153	TCP	93	[TCP Retransmission] ftp > 63066 [PSH, ACK] Seq=342 Ack=89 Win=32120 Len=39
2926	21:30:36.217	192.168.0.153	90.130.66.198	TCP	54	63066 > ftp [ACK] Seq=89 Ack=405 Win=64916 Len=0

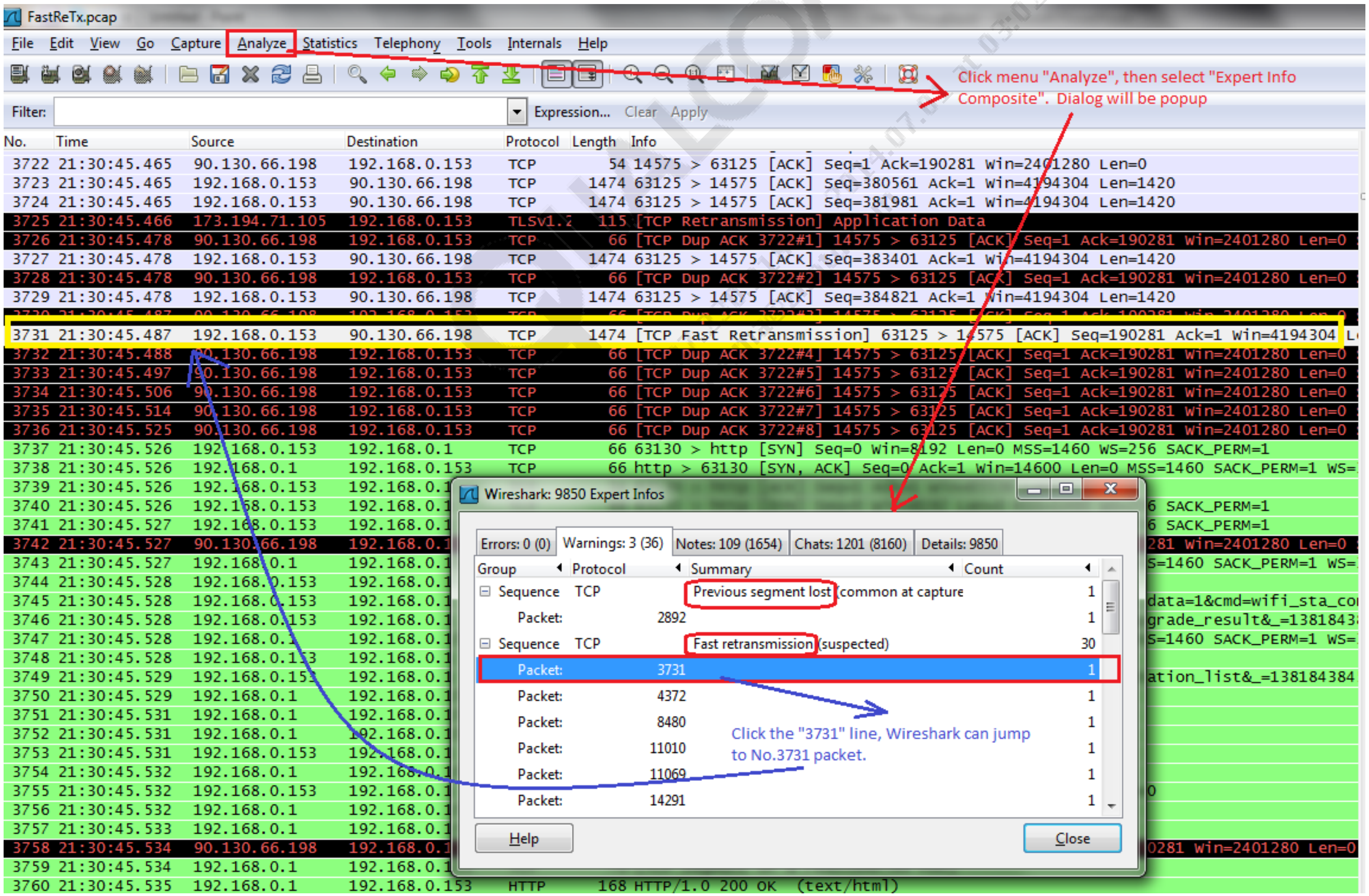
  - In line 2884, the sequence number of the last packet is #342.
  - In line 2892, the sequence number of this packet is #381, and the sequence number of the lost TCP packet is [342-380].
  - In line 2925, the packet with the sequence number of #342 is retransmitted.
  - In line 2926, ACK(381+24 = 405) is received. Succeed!
- DupACKs and fast retransmission

No.	Time	Source	Destination	Protocol	Length	Info
3523	21:30:45.028	192.168.0.153	90.130.66.198	TCP	1474	63125 > 14575 [ACK] Seq=190281 Ack=1 Win=4194304 Len=1420
3722	21:30:45.465	90.130.66.198	192.168.0.153	TCP	54	14575 > 63125 [ACK] Seq=1 Ack=190281 Win=2401280 Len=0
3726	21:30:45.478	90.130.66.198	192.168.0.153	TCP	66	[TCP Dup ACK 3722#1] 14575 > 63125 [ACK] Seq=1 Ack=190281 Win=2401280 Len=0 SLE=195961
3728	21:30:45.478	90.130.66.198	192.168.0.153	TCP	66	[TCP Dup ACK 3722#2] 14575 > 63125 [ACK] Seq=1 Ack=190281 Win=2401280 Len=0 SLE=195961
3730	21:30:45.487	90.130.66.198	192.168.0.153	TCP	66	[TCP Dup ACK 3722#3] 14575 > 63125 [ACK] Seq=1 Ack=190281 Win=2401280 Len=0 SLE=195961
3731	21:30:45.487	192.168.0.153	90.130.66.198	TCP	1474	[TCP Fast Retransmission] 63125 > 14575 [ACK] Seq=190281 Ack=1 Win=4194304 Len=1420

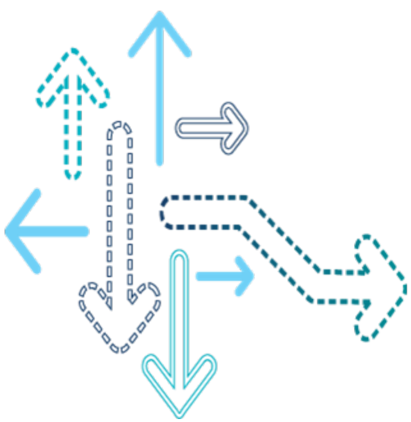
  - In line 3523, the packet with the TCP sequence number of #190281 is sent.
  - In line 3726/3728/2730, three duplicated ACKs are received.
  - In line 3731, the packet with the TCP sequence number of #190281 is fast transmitted (about 460 ms).
  - This shows TCP stack doesn't need to wait for RTO timeout, because fast retransmission can fast recover the TCP flow and keep a high level of throughput. For more information, refer to Appendix C.

Finding TCP Packet Loss/Retransmission with Wireshark

- The figure below shows how to find TCP packet loss and retransmission quickly with Wireshark.

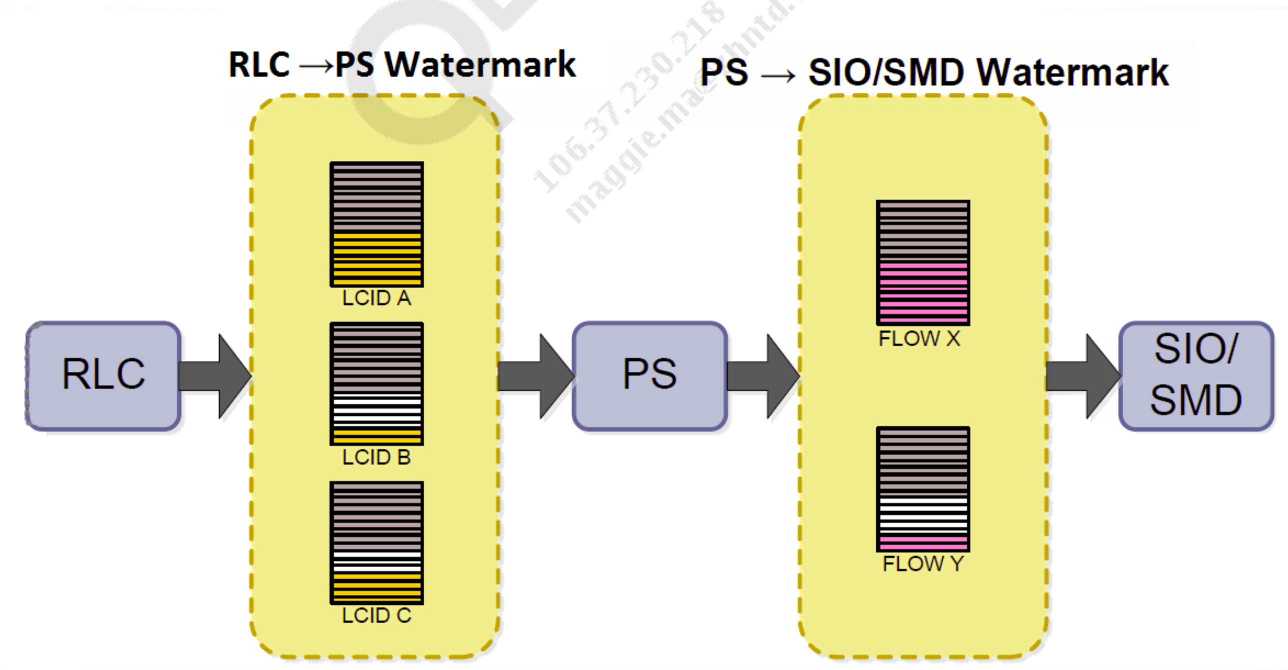


2.2. Watermark Flow Control Analysis



Watermark-Based Flow Control (WM Flow Control)

- Um Watermark
  - Buffer the data between RLC and PS (RLC <-> PS)
- Rm Watermark
  - Buffer the data between PS and SIO/SMD (PS <-> SIO/SMD)
- The data throughput will be affected if the WM flow control is frequently triggered. Thus finding the cause of flow control is necessary.



Case Study of Watermark Flow Control

- Check the QXDM log to find whether flow control is frequently triggered:
  - The key word is "disabling flow|enabling flow". See the figure below:

//The mask 0x800 module starts flow control and closes the data flow (PS cannot send data to Um Watermark), which lasts for 1.4 s
Data Services/Medium 08:59:54.933 ps_phys_link.c 00864 client 0x800 disabling flow on phys link 0x02BDC4F8 -> mask 0x800

Data Services/Medium 08:59:56.316 ps_phys_link.c 00793 client 0x800 enabling flow on phys link 0x02BDC4F8 -> mask 0x0

//The mask = 0x800 helps to confirm that the Low Layer (LL) started the flow control, and this issue needs to be checked in the RLC layer
#define DS_FLOW_LL_MASK 0x00000800 /**< Mask for all LL events. */

- Check whether the QXDM log includes the Watermark full message:
  - The key word is "WM full,freeing packet ". See the figure below:

Data Service Memory Buffers/Error 08:59:55.133 dsm_queue.c 00487 WM full,freeing packet 0x2261d6c:Watermark 0x2bdc720:Tried 1460

Data Service Memory Buffers/Error 08:59:55.133 dsm_queue.c 00487 WM full,freeing packet 0x2261d6c:Watermark 0x2bdc720:Tried 1460
... //Here are several WM full, freeing packet messages

Data Service Memory Buffers/Error 08:59:55.994 dsm_queue.c 00487 WM full,freeing packet 0x226200c:Watermark 0x2bdc720:Tried 1460

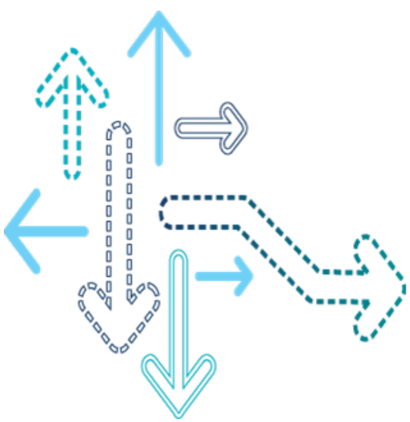
3. CDMA1x/EVDO Layer Analysis

EV-DO Data Throughput

- EV-DO data throughput troubleshooting list:
  - 80-N2363-1：DOrA_Data_Tput_Chklst_Troubleshoot
  - Perform self-check first according to the list in case of EV-DO throughput problems, and then submit a case if the problem remains.

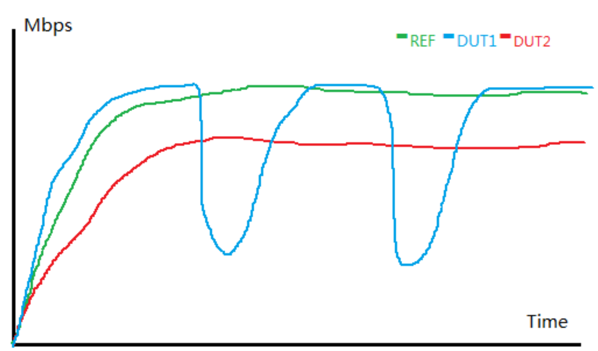
- Follow the settings below during the CT test
  - 1. 64 K Send buffer size settings of the Spirent APEX test device
    - Client Buffer Size Configurable True
    - Client Send Buffer Size(*1460) 44
    - Client Receive Buffer Size(*1460) 44
  - 2. TCP ACK prioritization
    - NV#67208 to 1 (Enabled)
  - 3. BP Removal ON
    - NV#69739 to 1, 1000
  - 4. Network model call
    - AT+CRM=2

4. WCDMA/HSPA Layer Analysis



Analysis of WCDMA/HSPA Data Throughput Problems

- As the WCDMA/HSPA data throughput test depends on the network environment, the target rate needs to be determined before the test. The rate of the same DUT varies in different test environments. Thus the DUT should be compared with the REF during problem analysis.
- The common rate issues are as follows:
  - The DUT1 underperforms the REF in the whole test process.
  - The DUT2 matches the performance of REF most of the time, but sometimes underperforms the REF.
- To solve the DUT2 issue, find when the rate slows and check from TCP to RLC to physical (TCP -> RLC -> Physical).
- To learn how to analyze the throughput problems from TCP to the physical layer, view this document that focuses on analyzing the DUT1 issues.



4.1 RLC Layer Analysis

Checking the RLC Layer (1)

- Analyze the WCDMA RLC layer if the DUT underperforms the REF in the UDP test.
  1. Check whether the RB rate meets the requirement.

     2013 Jul 18 12:22:00.429 [1C] 0x413A WCDMA Radio Bearer Rates

     Number of User Plane Radio Bearers = 2

Radio Bearer ID	Downlink RB Rate	Uplink RB Rate
5	42000000	5742000

     2013 Jul 18 12:24:42.650 [78] 0x413A WCDMA Radio Bearer Rates

     Number of User Plane Radio Bearers = 2

Radio Bearer ID	Downlink RB Rate	Uplink RB Rate
5	16000	16000

  2. Check RRC status. The RRC rate will be very low or zero in CELL_PCH/CELL_FACH status.

  3. Check whether error retransmission occurs in the RLC layer.

     1980 Jan 6 00:35:28.642 [F2] 0x414A WCDMA RLC DL AM Statistics

     Data Logical Channel ID = 19
     Control Logical Channel ID = 19
     Logical Channel Type = 5 = DTCH
     Total Number of PDU Bytes Received = 78745983
     Total Number of SDU Bytes to Upper Layer = 78303506
     Total Number of Error PDUs Received = 503
     Total Number of Data PDUs Received = 27869
     Total Number of Control PDUs Received = 1587
     Total Number of PDUs NAK'd by PE = 0
     Sequence Number of Last PDU in Seq. PDU = 289
     Highest Sequence Number PDU Received = 289

Checking the RLC Layer (2)

4. Search the keyword "RLC_ERR: RESET|Post DL reset msg|RESET: max_DAT|RLC RESET".

//The UE sends Reset to the network side

01:25:39.696 rlculam.c 3119 H RESET: max_DAT 0x8e (vt_dat, ignore MSB)for retx_sn 1 on LC 19

01:25:39.768 rlcul.c 5283 H Post RLC RESET to RRC, LC 19

//The UE receives reset sent from the network side

MSG WCDMA RLC/High 17:52:07.850 rlcdlam.c 11165 RLC_DL: Rcvd RESET PDU, reset_sn -- 0, LC Id 28
MSG WCDMA RLC/High 17:52:07.850 rlcdlam.c 11202 RLC_DL: Rcvd RESET with same reset_sn 0 on LC Id 28
MSG WCDMA RLC/High 17:52:08.310 rlcdlam.c 11165 RLC_DL: Rcvd RESET PDU, reset_sn -- 0, LC Id 28
MSG WCDMA RLC/High 17:52:08.310 rlcdlam.c 11202 RLC_DL: Rcvd RESET with same reset_sn 0 on LC Id 28

5. Check whether the RLC layer has flow control triggered. Currently, three types of flow control exist, which are based on memory, CPU loading and temperature respectively. Search the keyword "**rlcfc|RLC_FC**".

1) Memory-based flow control:

//Window open

MSG [00000/02] Legacy/High 00:00:57.586 pdcp.c 00487 Sending PDCP open Rx window request

MSG [03004/03] WCDMA RLC/Error 00:00:57.587 rlcul.c 00543 RLC_FC: NA: LC 19, Rcvd 3 (3:OPEN_RX_WIN,1:CLOSE_RX_WIN) cmd frm PDCP

MSG [03004/02] WCDMA RLC/High 00:00:57.587 rlcul.c 06886 Submit New WinSufi, ws 2047, LC 19

//Window close

MSG [00000/02] Legacy/High 00:00:58.099 pdcp.c 00432 Sending PDCP close Rx window request

MSG [03004/03] WCDMA RLC/Error 00:00:58.099 rlcul.c 00543 RLC_FC: NA: LC 19, Rcvd 1 (3:OPEN_RX_WIN,1:CLOSE_RX_WIN) cmd frm PDCP

MSG [03004/02] WCDMA RLC/High 00:00:58.099 rlcul.c 06886 Submit New WinSufi, ws 1, LC 19

Checking the RLC Layer (3)

2) **Temperature-based flow control:**

//Thermal flow control registered

MSG [03004/02] WCDMA RLC/High 15:02:33.087 rlcfc.c 01936 RLC_MULTI_FC::TEMPERATURE DL FC registerd for LC 19

//RLC window changed to 51

MSG [03004/02] WCDMA RLC/High 15:03:22.623 rlcfc.c 02446 TEMP_FC::Timer Callbk 0
MSG [03004/02] WCDMA RLC/High 15:03:22.623 rlcfc.c 02618 TEMP_FC::new window level 1
MSG [03004/02] WCDMA RLC/High 15:03:22.625 rlcfc.c 01772 MULTI_FC::prev output flow 2047 output flow 51
MSG [03004/02] WCDMA RLC/High 15:03:22.625 rlcfc.c 02351 RLC_MULTI_FC:: Action = TxWin(51)

//RLC window size changed to 20

MSG [03004/02] WCDMA RLC/High 15:05:22.622 rlcfc.c 02618 TEMP_FC::new window level 0
MSG [03004/02] WCDMA RLC/High 15:05:22.622 rlcfc.c 01772 MULTI_FC::prev output flow 51 output flow 20
MSG [03004/02] WCDMA RLC/High 15:05:22.622 rlcfc.c 02351 RLC_MULTI_FC:: Action = TxWin(20)

//Thermal flow control de-registered

MSG [03004/02] WCDMA RLC/High 15:06:15.147 rlcfc.c 01097 RLC DL De-registered with FC

3) **CPU loading flow control:**

//CPU loading is 100%

MSG [00043/02] Flow Controller/High 00:02:54.191 fc.c 00499 sleep 0, total 8196 samples -- loading 100

//Down the UL RLC window size

MSG [03004/02] WCDMA RLC/High 00:02:54.312 rlcfc.c 00637 Rxd DN CMD with cmd-id: 0 for 0 direction
MSG [03004/02] WCDMA RLC/High 00:02:54.312 rlcfc.c 00268 Step_dir 1, fc_ws_inc 0, fc_ws_stp 8
MSG [03004/02] WCDMA RLC/High 00:02:54.312 rlcfc.c 00722 FC WS: 20, STEP: 0, Step_WS_dir, 1

//Down the DL RLC window size

MSG [03004/02] WCDMA RLC/High 00:02:54.312 rlcfc.c 00637 Rxd DN CMD with cmd-id: 0 for 1 direction
MSG [03004/02] WCDMA RLC/High 00:02:54.312 rlcfc.c 00268 Step_dir 1, fc_ws_inc 0, fc_ws_stp 8
MSG [03004/02] WCDMA RLC/High 00:02:54.312 rlcfc.c 00722 FC WS: 20, STEP: 0, Step_WS_dir, 1
00:02:54.350 19 01 01 40 04 <-CONTROL PDU:: Type: STATUS SUFI[0]: WINDOW SIZE => 20 SUFI[1]: n/a

4.2 HSDPA Physical Downlink Analysis

Analyzing Physical Downlink (HSDPA) (1 of 4)

- If the RLC layer is normal, analyze the physical layer.
- Check HSDPA decoding status in downlink by selecting QXDM -> WCDMA -> HSDPA Decoding Statistics.
  - **Average physical layer requested rate –** It is acquired from the CQI measured in the physical layer, and reflects the channel quality of the current radio environment
  - **Average physical layer scheduled rate –** It is acquired from the size of scheduling block that the network side sends to the UE
  - **Average physical layer served rate –** It refers to the scheduling rate in a certain time that the network serves the UE
  - **Average MAC layer rate –** It refers to the rate that the UE successfully demodulates the block of the physical layer
  - **SBLER –** It refers to the block error rate of each sub-frame (retransmission is not included)
  - **Res BLER –** It refers to the block error rate after retransmission in the physical layer
- Check packet 0x4222 (WCDMA HS Decode Status With Data V3) in the QXDM log.

Analyzing Physical Downlink (HSDPA) (2 of 4)

WCDMA HSDPA Decoding Statistics

	Carrier 0	Carrier 1	Total
HS-SCCH DECODING STATISTICS			
Number of Subframes	8066	0	8066
HS-SCCH Attempts	8065	0	8065
HS-SCCH Successes	83	0	83
HS-SCCH Success Rate	1.03 %	0.00 %	1.03 %
ACK->NACK/DTX (Duplicate SB +) ...	1.32 %	0.00 %	1.32 %
TBS Changes During Retransmission	0	0	0
HS RATE STATISTICS			
Avg Physical Layer Rate (Requested)	1659.50 Kbps	-	1659.50 Kbps
Avg Physical Layer Rate P|S (Sched...	402.20 Kbps	-	402.20 Kbps
Avg Physical Layer Rate P|S (Served)	4.14 Kbps	-	4.14 Kbps
Avg MAC Layer Rate	3.54 Kbps	-	3.54 Kbps
MIMO STATISTICS			
Percentage Single Stream	0.00 %	0.00 %	0.00 %

MIMO	-
64QAM	-
UE Category	-
% Dual Carrier	0.00 %
HS MAC Resets	0
% 1 TB Requested	0.00 %
% 2 TBs Requested	0.00 %

Reset All

HS-DSCH Decoding Statistics Carrier 0

Carrier	TBS (bits)	QPSK P|S	16QAM ...	64QA...	SB- P|S	SB+ P|S	Dup. SB+ P|S	SBLER[1st] P|S (%)	Block-	Bloc...	Res. BLER (%)	1	2	3	4	5	>= 6
C0	Totals	81	2	0	7	75	1	8.54 [8.54]	7	75	8.54	75	0	0	0	0	0
C0	5782	0	1	0	0	1	0	0.00 [0.00]	0	1	0.00	1	0	0	0	0	0
C0	4748	0	1	0	0	1	0	0.00 [0.00]	0	1	0.00	1	0	0	0	0	0
C0	2404	5	0	0	2	3	0	40.00 [40.00]	2	3	40.00	3	0	0	0	0	0
C0	2046	5	0	0	1	4	0	20.00 [20.00]	1	4	20.00	4	0	0	0	0	0
C0	1711	1	0	0	0	1	0	0.00 [0.00]	0	1	0.00	1	0	0	0	0	0
C0	1380	3	0	0	1	2	0	33.33 [33.33]	1	2	33.33	2	0	0	0	0	0
C0	1036	2	0	0	0	2	0	0.00 [0.00]	0	2	0.00	2	0	0	0	0	0
C0	699	7	0	0	0	7	0	0.00 [0.00]	0	7	0.00	7	0	0	0	0	0
C0	365	58	0	0	3	54	1	5.26 [5.26]	3	54	5.26	54	0	0	0	0	0

Analyzing Physical Downlink (HSDPA) (3 of 4)

```
1980 Jan  6  00:33:50.647  [AA]  0x4222  HS Decode Status Log Packet with Data Edition 3
Version =  9
Number of Samples =  25
Start SFN = 850
Mac-ehs enabled = Yes
MIMO CQI reporting = FALSE
Transport Block Size Table = octet aligned
64QAM configured = TRUE
Max number of Harq Processes = 6
Dual Carrier Enabled = false
|--|-----|----|-----|---|---|----|----|-----|----|-----------|---|---|---|---|---|
| #| SCCH|DSCH|HS TB|XRV|New|Num |Code|     |HARQ|SCCH       |   |   |   |Sec|Sec|
|  |A/V/T|Stat| size|   | Tx|Code|Off | Mod |  Id|Type|Ord|Id|DTX|DRX|HSL| HS| UL|
|--|-----|----|-----|---|---|----|----|-----|----|----|---|--|---|---|---|---|---|
| 0|1 1 1|PASS|32264|  6|  1|  14|   1|64QAM|   0|   1|  0| 1|   |   |   |   |   |
| 1|1 1 1|PASS|24232|  6|  1|  15|   1|16QAM|   1|   1|  0| 1|   |   |   |   |   |
| 2|1 1 1|PASS|16352|  6|  1|  13|   1|16QAM|   5|   1|  0| 1|   |   |   |   |   |
| 3|1 1 1|PASS|31128|  6|  1|  15|   1|64QAM|   2|   1|  0| 1|   |   |   |   |   |
| 4|1 1 1|PASS|31128|  6|  1|  15|   1|64QAM|   3|   1|  0| 1|   |   |   |   |   |
| 5|1 1 1|PASS|28976|  6|  1|  15|   1|64QAM|   4|   1|  0| 1|   |   |   |   |   |
| 6|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   0|   1|  0| 1|   |   |   |   |   |
| 7|1 1 1|PASS|31128|  6|  1|  15|   1|64QAM|   1|   1|  0| 1|   |   |   |   |   |
| 8|1 1 1|PASS|26976|  6|  1|  15|   1|64QAM|   5|   1|  0| 1|   |   |   |   |   |
| 9|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   2|   1|  0| 1|   |   |   |   |   |
|10|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   3|   1|  0| 1|   |   |   |   |   |
|11|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   4|   1|  0| 1|   |   |   |   |   |
|12|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   0|   1|  0| 1|   |   |   |   |   |
|13|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   1|   1|  0| 1|   |   |   |   |   |
|14|1 1 1|PASS|31128|  6|  1|  15|   1|64QAM|   5|   1|  0| 1|   |   |   |   |   |
|15|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   2|   1|  0| 1|   |   |   |   |   |
|16|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   3|   1|  0| 1|   |   |   |   |   |
|17|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   4|   1|  0| 1|   |   |   |   |   |
|18|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   0|   1|  0| 1|   |   |   |   |   |
|19|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   1|   1|  0| 1|   |   |   |   |   |
|20|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   5|   1|  0| 1|   |   |   |   |   |
|21|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   2|   1|  0| 1|   |   |   |   |   |
|22|1 1 1|PASS|31128|  6|  1|  15|   1|64QAM|   3|   1|  0| 1|   |   |   |   |   |
|23|1 1 1|PASS|31128|  6|  1|  15|   1|64QAM|   4|   1|  0| 1|   |   |   |   |   |
|24|1 1 1|PASS|31128|  6|  1|  15|   1|64QAM|   0|   1|  0| 1|   |   |   |   |   |
```

Analyzing Physical Downlink (HSDPA) (4 of 4)

- Common causes of a low HSDPA rate:
  - RF has not been calibrated.
  - Equalizer, SCH-IC and QICE are not enabled. Modifying the NV 3825 might cause this problem. This NV should not be activated, and the corresponding default value is set in the software code.
  - The RF diversity is not enabled, the diversity has not been calibrated, or the test cable is not connected to the diversity.
  - A large and fixed transmission block is set on the device side. Thus all the initial transmissions are incorrectly demodulated, and the rate is affected.
  - The HSDPA category of the UE does not match the network.

4.3 HSUPA Physical Uplink Analysis

Analyzing Physical Uplink (HSUPA)

- To check HSUPA of the uplink, click QXDM-→WCDMA→EUL Link Statistics.

WCDMA EUL Link Statistics

Throughput	Last Second	Total
Avg Raw TP	4437.16 kbps	4067.85 kbps
Avg Total TP	4369.15 kbps	3991.06 kbps
Avg Scheduled TP	4369.15 kbps	3991.06 kbps
Avg TP Supported by SG	4754.74 kbps	4184.90 kbps
Avg TP Supported by Avail Power	5738.81 kbps	5738.56 kbps
Avg TP Supported by Sched Data	8225.59 kbps	12081.56 kbps
% Limited by SG	69.67 %	58.40 %
% Limited by Power	0.00 %	0.01 %
% Limited by Sched Data	0.00 %	48.64 %

Grants	Last Second	Total
Number of AGs	12	1346
Mean AG	21.92	21.33
% Serving RG Down	0.00 %	0.00 %
% Serving RG Hold	100.00 %	100.00 %
% Serving RG Up	0.00 %	0.00 %
% Non-Serving RG Down	0.00 %	0.00 %
% Non-Serving RG Hold	0.00 %	0.00 %
Avg Serving Grant	28.89 dB	25.15 dB

Buffer Status	Last Second	Total
Avg Scheduled Buffer Status	13864.36 byte(s)	13152.19 byte(s)
Avg Non-Scheduled Buffer Status	0.00 byte(s)	0.00 byte(s)
% Scheduled Buffers Empty	0.00 %	23.77 %
% Non-Scheduled Buffers Empty	92.33 %	92.60 %
% Happy	0.00 %	27.25 %
Number of SIs	205	23783
% of TTIs with SI	68.33 %	55.57 %

UL Transmission	Last Second	Total
% New Transmissions	74.00 %	62.50 %
% Retransmissions	7.67 %	7.47 %
% DTX	18.33 %	30.02 %
BLER	10.20 %	10.85 %
Residual BLER	0.90 %	0.19 %
MAC-e Resets	0	4

UL Power	Last Second	Total
DPCCH	-54.27 dBm	-54.81 dBm
DPCCH %	0.0000 %	0.0000 %
DPDCH T/P	-	-
DPDCH %	0.0000 %	0.0000 %
HS-DPCCH T/P	-0.0070 dB	-0.0070 dB
HS-DPCCH %	0.0000 %	0.0000 %
E-DPCCH T/P	-0.8913 dB	-1.5587 dB
E-DPCCH %	0.0000 %	0.0000 %
E-DPDCH T/P	19.2094 dB	18.1457 dB
E-DPDCH %	0.0001 %	0.0001 %
Power Remaining	58.91dB	60.49dB
Power Remaining %	99.9999 %	99.9999 %

Analyzing Physical Uplink (HSUPA)

- Or you can analyze log packet 0x4309.

1980 Jan 6 00:34:01.909 [10] 0x4309 EUL Combined L1/MAC

Number of Samples = 40

TTI = 2ms

ETFCI Table = 1

Start CFN = 84

Number of Cells = 1

Serv Cell = 0

EDCHcell	RG_ID	TPC_ID	PSC
0 (serv)	0	0	111

SuFN	HQ	Serv RGCH	NS RGCH	Cell_RGCH 111 0			AGCH V	S	I	MAC_D	Reas	SG	LUPR	ReTx CTR	SI	HP	ETFCI	TBS	CM	Comb HICH	Cell_HICH 111 0			
109	5	HLD		HLD						00000010	BO	14	13	0		1	3	354		ACK	ACK			
110	6	HLD		HLD						00000010	BO	14	0	0		1	3	354		ACK	ACK			
111	7	HLD		HLD							BO	14	0	DTX						NAK	DTX			
112	0	HLD		HLD						00000010	BO	14	0	0		1	3	354		ACK	ACK			
113	1	HLD		HLD						00000010	SG	14	0	0		1	3	354		ACK	ACK			
114	2	HLD		HLD								14	13	1		1	3	354		ACK	ACK			
115	3	HLD		HLD						00000010	SG	14	13	0		1	3	354		ACK	ACK			
116	4	HLD		HLD						00000010	SG	14	13	0		1	3	354		ACK	ACK			
117	5	HLD		HLD						00000010	SG	14	13	0		1	3	354		ACK	ACK			
118	6	HLD		HLD						00000010	SG	14	13	0		1	3	354		ACK	ACK			
119	7	HLD		HLD						00000010	SG	14	0	0		1	3	354		ACK	ACK			
120	0	HLD		HLD						00000010	SG	14	13	0		1	3	354		ACK	ACK			
121	1	HLD		HLD						00000010	SG	14	13	0		1	3	354		ACK	ACK			
122	2	HLD		HLD						00000010	SG	14	13	0		1	3	354		ACK	ACK			
123	3	HLD		HLD						00000010	SG	14	13	0		1	3	354		NAK	DTX			

Analyzing Physical Uplink (HSUPA)

- Common causes for low speed of HSUPA
  - Low SG allocated by the network

SuFN	HQ	Serv RGCH	NS RGCH	Cell_RGCH 111 0				AGCH V	S	I	MAC_D	Reas	SG
198	6	HLD		HLD							00000010	SG	27
199	7	HLD		HLD							00000010	SG	27
200	0	HLD		HLD							00000010	SG	27
201	1	HLD		HLD							00000010	SG	27

- Power used for HSUPA transmission is limited. You need to check the transmit calibration and the maximum transmit power of the mobile phone.

SuFN	HQ	Serv RGCH	NS RGCH	Cell_RGCH 309 0				AGCH V	S	I	MAC_D	Reas
1019	3	HLD		HLD							00000010	MP
1020	4	HLD		HLD							00000010	MP
1021	5	HLD		HLD							00000010	MP
1022	6	HLD		HLD							00000010	MP

Analyzing Physical Uplink (HSUPA)

- The data sent to the physical layer from the upper layer is too little. You need to check whether there is flow control for the DATA layer or whether there is packet loss.

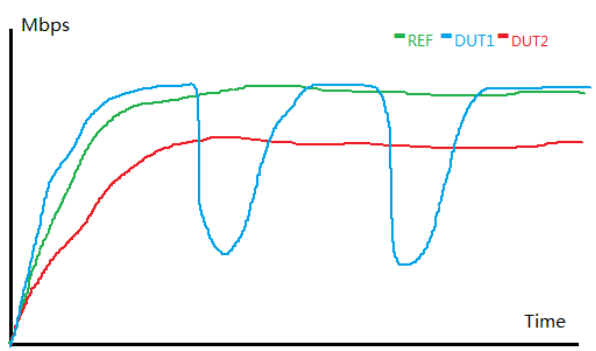
SuFN	HQ	Serv RGCH	NS RGCH	Cell_RGCH 111 0				AGCH V	S	I	MAC_D	Reas
148	4	HLD		HLD								BO
149	5	HLD		HLD								BO
150	6	HLD		HLD								BO
151	7	HLD		HLD								BO

- The transmission bit error rate is too high. You need to check whether the RF transmission is correct and do the comparison with the reference device.

5. TD-SCDMA/HSPA Layer Analysis

Analysis of TD-SCDMA/HSPA Data Throughput Problems

- As the TD-SCDMA/HSPA data throughput test depends on the network environment, the target rate needs to be determined before the test. The rate of the same DUT varies in different test environments. Thus the DUT should be compared with the REF during problem analysis.
- The common rate issues are as follows:
  - The DUT1 underperforms the REF in the whole test process.
  - The DUT2 matches the performance of REF most of the time, but sometimes underperforms the REF.
- To solve the DUT2 issue, find when the rate slows and check from TCP to RLC to physical (TCP -> RLC -> Physical).
- To learn how to analyze the throughput problems from TCP to the physical layer, view this document that focuses on analyzing the DUT1 issues.



5.1 RLC Layer Analysis

Checking the RLC Layer (1)

- Analyze the TD-SCDMA RLC layer if the DUT underperforms the REF in the UDP test.

1. Check whether the RB rate meets the requirement.

1980 Jan 8 09:09:11.389 [00] 0xD0E4 TDSCDMA RRC RB Rate Info

```
Version = 1
Version 1 {
  num_rbs = 1
  Rb Rate {
    rb_id = 5
    dl_rb_rate = 2800000//2.8Mbps means DL configure HSDPA.
    ul_rb_rate = 16000//UL is 16kbps
  }
}
```

2. Check RRC status. The RRC rate will be very low or zero in CELL_PCH/CELL_FACH status.

3. Check whether error retransmission occurs in the RLC layer.

```
Rlc Dl Am Stat[3] {
    log_data_dl_rlc_id = 27
    log_ctl_dl_rlc_id = 27
    log_chan_type = UE_LOGCHAN_DTCH
    tot_num_pdu_byte_rxd = 1148902
    tot_num_sdu_byte_rxd = 1095428
    tot_num_error_pdu_rxd = 35
    tot_num_data_pdu_rxd = 13744
    tot_num_ctl_pdu_rxd = 232
    tot_num_pdu_nak = 0
    sn_last_inseq_pdu_rxd = 1456
    highest_sn_pdu_rxd = 1456
  }
```


Checking the RLC Layer (2)

4. Search the keyword "RLC_ERR: RESET|Post DL reset msg|RESET: max_DAT|RLC RESET".

//The UE sends Reset to the network side

MSG [10254/02] TDSCDMA RLC/High 02:01:22.970 tdsrlcul.c 04360 Post RLC RESET to RRC, LC 24

MSG [10255/02] TDSCDMA RRC/High 02:01:22.971 tdsrrccu.c 22719 RLC RESET type:0,LC_id:24,dir:1

//The UE receives reset sent from the network side

MSG [10254/02] TDSCDMA RLC/High 02:17:11.350 tdsrlcdlam.c 05831 RLC_DL: Rcvd RESET PDU, reset_sn -- 0, LC Id 25

MSG [10254/02] TDSCDMA RLC/High 02:17:11.350 tdsrlcdlam.c 01725 Post UL RESET Msg, LC 25

5.2 HSDPA Physical Downlink Analysis

Analyzing Physical Downlink (HSDPA) (1 of 5)

- If the RLC layer is normal, analyze the physical layer.
- First check the HSPDA Summary in APEX

CQI Summary		
	Average RTBS	46.64
	% of 16QAM	98
	% of QPSK	1
	No. of times seEff = 0 / Total Sample, (%)	0 / 3727, (0)
	No. of times RTBS = 0 / Total Sample, (%)	0 / 3727, (0)
	Average Requested code rate	0.86
Average Target SIR	2.60	
Average HS-SCCH SIR	17.80	
Average SIR	9.46	
Average RSCP	-87.98	
Average Tx Power		
	Slot	Tx Power
Percentage of Scheduling: (HS-SCCH Successes / Attempts, (Success Rate in %))	4612 / 14336, (32.1708)	
Percentage of allocated modulation scheme		
	QPSK	21.39
	16 QAM	78.61
Average TBS index	42.9612	
HS-SCCH BLER HCSN out of sequence	1	
Percentage of Ts Hs Pdsch		
	Ts Hs Pdsch	Percentage
4:HS-PDSCH TS is 4	4	0.24
6:HS-PDSCH TS is 4,5	6	0.38
8:HS-PDSCH TS is 3	8	22.15
12:HS-PDSCH TS is 3,4	12	11.92
14:HS-PDSCH TS is 3,4,5	14	65.31

Analyzing Physical Downlink (HSDPA) (2 of 5)

- Check TDS HSSCCH Sched by selecting APEX->TDSCDMA->L1->0xD032 TDSCDMA L1 HSDPA HSSCCH Statistics

1980 Jan 8 09:09:18.025 [00] 0xD032 TDSCDMA L1 HSDPA HSSCCH Statistics

```
Version = 1
Version 1 {
   cell_id = 0
   start_sys_frame_no = 2902
   reserved = 0
   num_sub_frames = 512
   num_scch_valid = 512//means the number of SCCH every 512 sub frames
}
```

- If num_scch_valid is very small, means NW Scheduling HSSCCH is not sufficient.

Analyzing Physical Downlink (HSDPA) (3 of 5)

- Check HCSN by selecting APEX->TDSCDMA->L1->0xD031 TDSCDMA L1 HSDPA Decode Status

1980 Jan 8 09:09:11.575 [00] 0xD031 TDSCDMA L1 HSDPA Decode Status

Version = 1
Version 1 {
  cell_id = 0
  reserved = 0
  num_samples = 10
  info = 976
  Decode Status[0] {
    phyChType = 0
    Scch {
      arriving_sys_frame_no = 2248
      arriving_sub_frame_no = 4497
      crc_pass = 1
      hcsn = 0

Time	hcsn
09:09:12.285	0
09:09:12.285	1
09:09:12.285	2
09:09:12.285	3
09:09:12.490	4
09:09:12.490	5
09:09:12.490	6
09:09:12.490	7
09:09:12.490	0
09:09:12.725	1
09:09:12.725	2
09:09:12.725	3
09:09:12.725	4
09:09:12.725	5
09:09:12.905	6
09:09:12.905	7
09:09:12.905	0
09:09:12.905	1
09:09:12.905	2
09:09:13.210	3
09:09:13.210	4
09:09:13.210	5
09:09:13.210	0

miss 6 and 7, SCCH not continuous

- HCSN should be a cycle from 0 to 7, if not, means SCCH is not continuous.

Analyzing Physical Downlink (HSDPA) (4 of 5)

- Check BLER by selecting APEX->TDSCDMA->L1->0xD033 TDSCDMA L1 HSDPA HARQ Statistics

1980 Jan 8 09:09:25.705 [00] 0xD033 TDSCDMA L1 HSDPA HARQ Statistics

Version = 3
Version 3 {
  cell_id = 0
  reserved = 0
  num_sub_frames = 512
  start_sys_frame_no = 3670
  num_harq_proc = 5
  bler_overall_newtx = 6
  bitrate_avg_kbps = 578
  bler_2secs_newtx = 0
  bitrate_2secs_kbps = 149

Time	bler_2secs_newtx
09:08:05.975	0
09:09:12.885	8
09:09:15.465	1
09:09:18.025	8
09:09:20.585	8
09:09:23.145	11
09:09:25.705	0
09:09:28.265	0
09:09:30.825	0
09:09:33.385	10
09:09:35.945	1
09:09:38.505	1
09:09:41.065	1
09:09:43.625	1
09:09:46.185	3
09:09:48.745	4
09:09:51.305	3
09:09:53.865	1
09:09:56.425	0

- If bler_2secs_newtx is very high, may be related to RF receive.

Analyzing Physical Downlink (HSDPA) (5 of 5)

- Common causes of a low HSDPA rate:
  - RF has not been calibrated.
  - If there is power or signal issue, maybe RF antenna switch timing has been changed.
  - A large and fixed transmission block is set on the device side. Thus all the initial transmissions are incorrectly demodulated, and the rate is affected.
  - The HSDPA category of the UE does not match the network.

5.3 HSUPA Physical Uplink Analysis

Analyzing Physical Uplink (HSUPA)

- To check HSUPA ETFC, click APEX-→TDSCDMA→0xD044 TDSCDMA L1 UPA ETFC Restriction Info.

1980 Jan 6 02:42:10.075 [0/0x00] 0xD044 TDSCDMA L1 UPA ETFC Restriction Info

```
Version = 2
Version2 {
   num_samples = 40
   first_sample_sub_fn = 1169
   max_allowed_tx_pwr = 24
   Etfc Restrict Info
```

#	subfn_offset	sg_type	snpl	max_epuch_pwr_avail	other_ch_type	prri	uph	pebase	dtx_flag	max_sup_etfci	max_sg_etfci	etfci	harq_id
0	0	0	31	26	0	20	30	-126	0	59	59	59	0
1	1	0	31	26	0	20	30	-126	0	59	59	59	2
2	2	0	31	26	0	20	30	-125	0	59	59	59	1
3	3	0	31	26	0	20	30	-125	0	59	59	59	3
4	4	0	31	26	0	20	30	-123	0	59	59	59	0
5	5	0	31	26	0	20	30	-123	0	59	59	59	2
6	6	0	31	26	0	20	30	-121	0	59	59	59	1
7	7	0	31	26	0	20	30	-121	0	59	59	59	3
8	8	0	31	26	0	20	30	-121	0	59	59	59	0
9	9	0	31	26	0	20	30	-121	0	59	59	59	2
10	10	0	31	26	0	20	30	-121	0	59	59	59	1
11	11	0	31	26	0	20	30	-121	0	59	59	59	3
12	12	0	31	26	0	20	30	-121	0	59	59	59	0
13	13	0	31	26	0	20	30	-121	0	59	59	59	2
14	14	0	31	26	0	20	30	-121	0	59	59	59	1

- If etfci is very small, need further check others, such as the total data from up layer, the Absolute Grant allocated by the network and the TX power of UE.

Analyzing Physical Uplink (HSUPA)

- Check AG in 0xD046 TDSCDMA L1 UPA EDL Dec And SG Info.

1980 Jan 6 02:42:10.075 [0/0x00] 0xD046 TDSCDMA L1 UPA EDL Dec And SG Info

```
Version = 1
Version 1 {
   num_samples = 40
   first_sample_sub_fn = 1168
   Sample Info
```

#	sg_type	ag_flag	prri	crri	trri	eni	rdi	hich_rvd_flag	hich_ack_nack	hich_harq_id	select_harq_id	new_tx	etfci	curr_tx_nb	max_tx_nb	rtx_tmr	rtx_tmr_exp	erucch_flag
0	1	1	20	0	8	1	0	1	1	0	0	1	59	0	5	0	72	0
1	1	1	20	0	8	1	0	1	1	2	2	1	59	0	5	0	72	0
2	1	1	20	0	8	1	0	1	1	1	1	1	59	0	5	0	72	0
3	1	1	20	0	8	1	0	1	1	3	3	1	59	0	5	0	72	0
4	1	1	20	0	8	1	0	1	0	0	0	0	59	1	5	0	72	0
5	1	1	20	0	8	1	0	1	0	2	2	0	59	1	5	0	72	0
6	1	1	20	0	8	1	0	1	1	1	1	1	59	0	5	0	72	0
7	1	1	20	0	8	1	0	1	0	3	3	0	59	1	5	0	72	0

- If PRRI is very small, means the Absolute Grant allocated by the network is low.

Analyzing Physical Uplink (HSUPA)

- Check BLER in Log packet 0xD048 TDSCDMA L1 UPA Statistics Info.

0xD048 TDSCDMA L1 UPA Statistics Info

```
Version = 3
Version 3 {
   curr_pathloss = 107
   snpl = 31
   schld_stats_included = 1
   non_schld_stats_included = 0
   Schld Stats Info {
      start_sub_fn = 1008
      num_sub_frames = 200
      num_sg_avail_sub_frames = 196
      num_new_tx = 184
      num_re_tx = 12
      num_ack = 184
      num_nack = 12
      num_nack_new_tx = 12
      num_failed_harq = 0
      total_bler = 6.52 %
      new_tx_bler = 6.52 %
      residual_bler = 0.00 %
      upa_l1_tput = 415840
      sum_raw_bits = 442960
      sum_pwr_bits = 442960
      sum_sg_bits = 442960
      sum_s_buf = 0
      sum_etfci = 10856
      num_agch = 196
      num_spacing_tti = 0
      sum_prri = 1568
      pebase = -123
      num_tpc_up = 22
      num_tpc_down = 174
      num_dtx = 0
      sum_num_vu = 3136
      num_rtx_timeout = 0
      num_tx_num_exceed = 0
      num_first_tx_acked = 172
      num_pwr_limited = 0
      num_sg_limited = 184
      num_si_epuch = 0
      sum_epuch_pwr = 510
      num_success_erucch = 0
      num_failed_erucch = 0
      num_abort_erucch = 0
      num_ehich_lost = 0
      num_eagch_lost = 0
      sum_num_ts = 196
      num_sg_maller_than_uph = 0
      num_harq_fail_per_id = { 0, 0, 0, 0 }
```

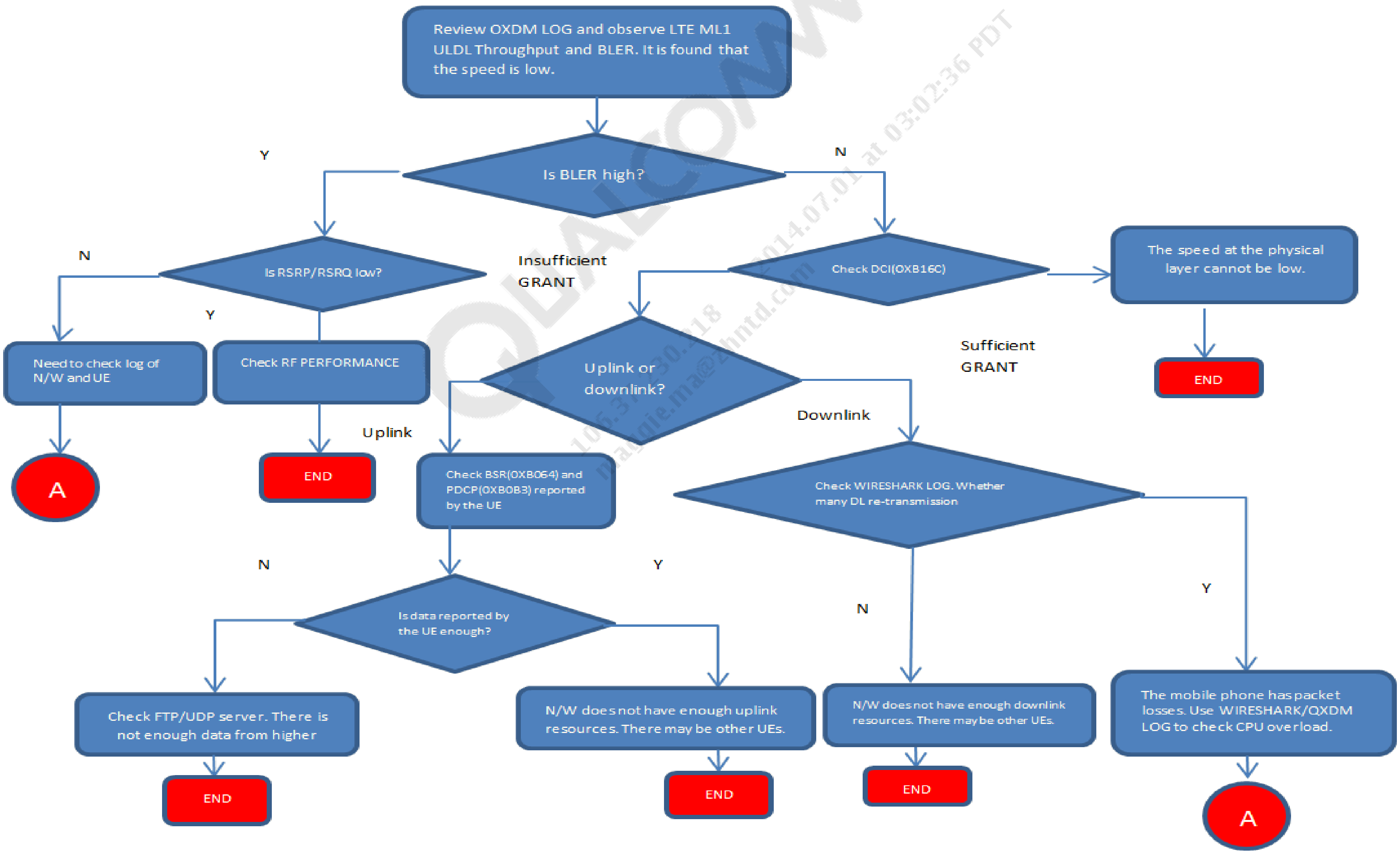
If total_bler is very high, may be related to RF transmission.

Analyzing Physical Uplink (HSUPA)

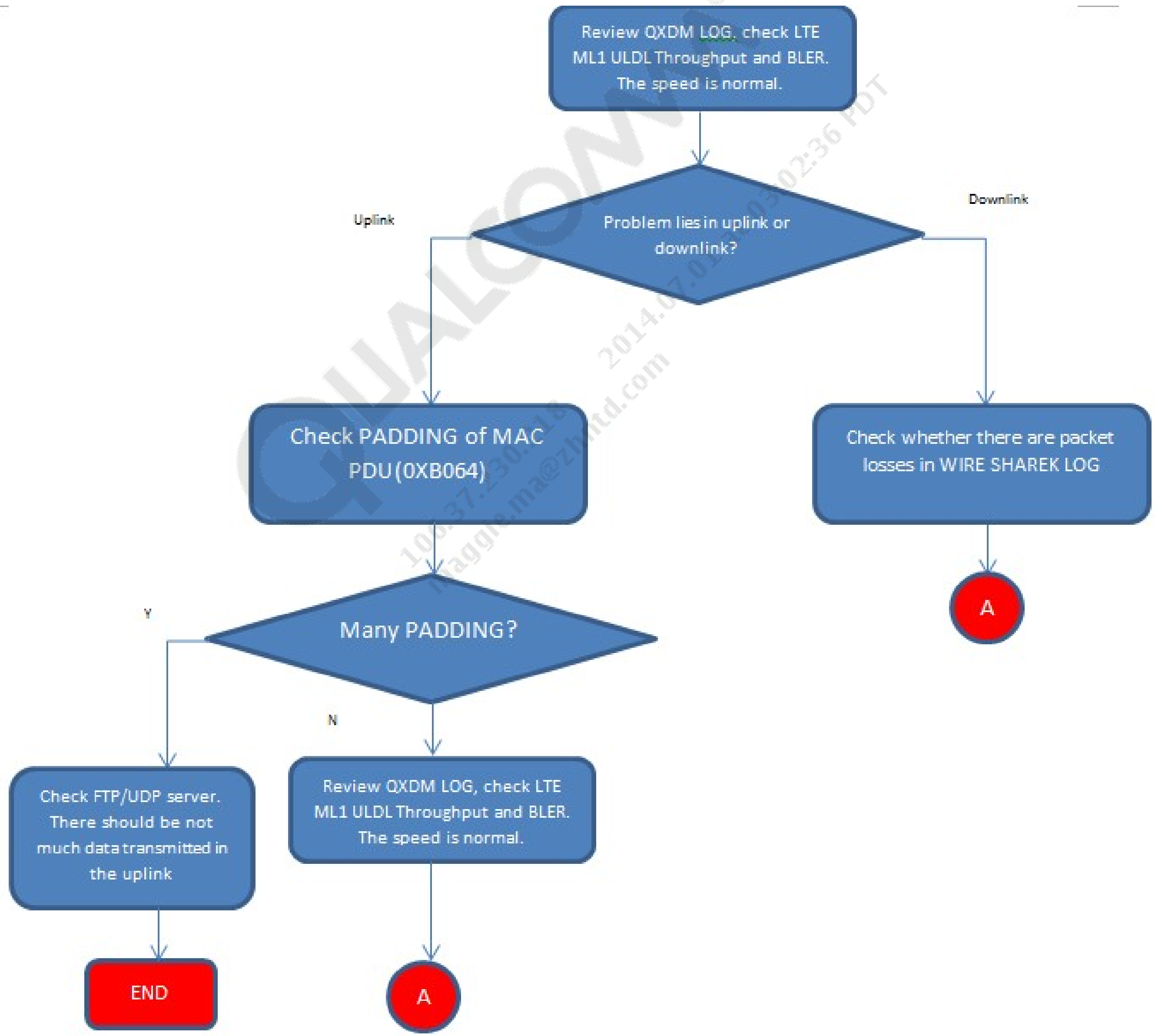
- Common causes for low speed of HSUPA
  - Low AG allocated by the network
- Power used for HSUPA transmission is limited. Check the transmit calibration and the maximum transmit power of the mobile phone.
- The data sent to the physical layer from the upper layer is too little. Check whether there is flow control for the DATA layer or whether there is packet loss.
- The transmission bit error rate is too high. Check whether the RF transmission is correct and do the comparison with the reference device.

6. LTE Layer Analysis

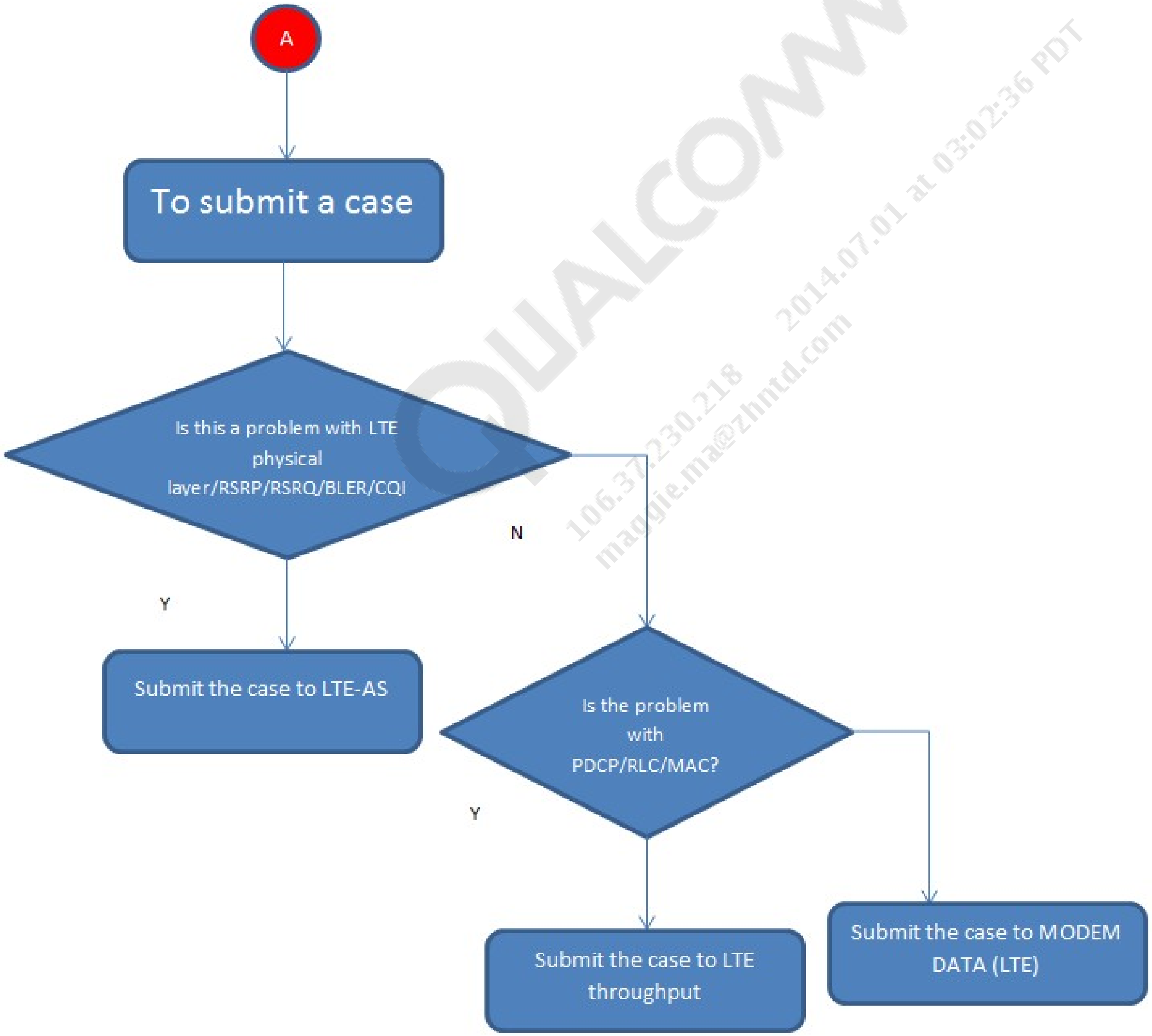
Check Procedures for LTE Physical Layer with Low Speed



Check Procedures for LTE Physical Layer with Normal Speed



Submit Cases



7. RFSW Analysis

RF Analysis contents

- Scope of application
- RF timing
- Mistake list
- Our rule

Scope of Application

- All QCT platform which use GPIO for antenna switch ,PA_ON, TX_ON, and RX_ON includes 8960,8930, 9x15, Fusion3, 8974, 8974AB, 9x25, 8x26,8926,8x10,8x12..

Antenna timing-RX

Use 9x15 code for example: QCT default Rx antenna timing is (-5,0), RX_ON0 timing is (-150,0), the unit of timing is us, for example -5 means open ASM Rx path advance 5 us of rx frame.

```
rfc_signal_gsm_cfg_type rf_card_wtr1605_chn3_rx0_tdscdma_b34_sig_cfg =
{
  {
    WTR1605, /* rf_asic_name */
    0, /* rf_asic_id */
    ( RFDEVICE_PA_LUT_MAPPING_INVALID ), /* rf_asic_band_pa_lut_map */
    (int)WTR1605_TDSCDMA_PRX_BAND34_PMB1, /* rf_asic_sw_port_map */
    FALSE, /* txlut, config */
  },
  0x01230111,   /* Revision: V1.35.273 */
  {
    {  { (int)RFC_WTR1605_CHN3_RX_ON0_DEFAULT, RFC_HIGH}, {-150,0}  },
    {  { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_10_DEFAULT, RFC_HIGH}, {-5,0}  },
    {  { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_11_DEFAULT, RFC_LOW}, {-5,0}  },
    {  { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_12_DEFAULT, RFC_HIGH}, {-5,0}  },
    {    { (int)RFC_SIG_LIST_END, RFC_LOW  },   {0, 0}} ,
  },
};
```

Antenna timing-TX

- Use 9x15 code for example: QCT default Tx antenna timing is (-5,-6), PA ON timing is (-5,-4), TX ON timing is (-25,12)

```
rfc_signal_gsm_cfg_type rf_card_wtr1605_chn3_tx0_tdscdma_b34_sig_cfg =
{
  {
    WTR1605, /* rf_asic_name */
    0, /* rf_asic_id */
    ( RFDEVICE_PA_LUT_MAPPING_INVALID ), /* rf_asic_band_pa_lut_map */
    (int)WTR1605_TDSCDMA_TXWSAW_BAND34_TMB1, /* rf_asic_sw_port_map */
    TRUE, /* txlut, config */
  },
  0x01230111,    /* Revision: V1.35.273 */
  {
    {   { (int)RFC_WTR1605_CHN3_PA_ON_2_TX0_DEFAULT, RFC_HIGH}, {-5,-4}  },
    {   { (int)RFC_WTR1605_CHN3_TX_ON0_DEFAULT, RFC_CONFIG_ONLY}, {-25,12}  },
    {   { (int)RFC_WTR1605_CHN3_PA0_R0_DEFAULT, RFC_CONFIG_ONLY}, {-5,-4}  },
    {   { (int)RFC_WTR1605_CHN3_PA0_R1_DEFAULT, RFC_CONFIG_ONLY}, {-5,-4}  },
    {   { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_10_DEFAULT, RFC_HIGH}, {-5,-6}  },
    {   { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_11_DEFAULT, RFC_HIGH}, {-5,-6}  },
    {   { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_12_DEFAULT, RFC_HIGH}, {-5,-6}  },
    {   { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_13_DEFAULT, RFC_LOW}, {-5,-6}  },
    {   { (int)RFC_WTR1605_CHN3_PA0_SMPS_PDM_DEFAULT, RFC_CONFIG_ONLY}, {-54,-4}  },
    {   { (int)RFC_WTR1605_CHN3_GNSS_BLANK_TDS, RFC_HIGH}, {-6,-3}  },
    {   { (int)RFC_WTR1605_CHN3_INTERNAL_03_DEFAULT, RFC_CONFIG_ONLY}, {-11,-3}  },
    {     { (int)RFC_SIG_LIST_END, RFC_LOW  },  {0, 0}} ,
  },
```

Mistake List (1)

Below example change antenna timing to pass some test, such as RF HW TX spurious emission, so change code from (-5,-6) to (-30,0)

this will cause critical low TD-HSDPA throughput issue

```
rfc_signal_gsm_cfg_type rf_card_wtr1605_chn3_tx0_tdscdma_b34_sig_cfg =
{
  {
    WTR1605, /* rf_asic_name */
    0, /* rf_asic_id */
    ( RFDEVICE_PA_LUT_MAPPING_INVALID ), /* rf_asic_band_pa_lut_map */
    (int)WTR1605_TDSCDMA_TXWSAW_BAND34_TMB1, /* rf_asic_sw_port_map */
    TRUE, /* txlut, config */
  },
  0x01230111,   /* Revision: V1.35.273 */
  {
    {  { (int)RFC_WTR1605_CHN3_PA_ON_2_TX0_DEFAULT, RFC_HIGH}, {-5,-4}  },
    {  { (int)RFC_WTR1605_CHN3_TX_ON0_DEFAULT, RFC_CONFIG_ONLY}, {-25,12}  },
    {  { (int)RFC_WTR1605_CHN3_PA0_R0_DEFAULT, RFC_CONFIG_ONLY}, {-5,-4}  },
    {  { (int)RFC_WTR1605_CHN3_PA0_R1_DEFAULT, RFC_CONFIG_ONLY}, {-5,-4}  },
    {  { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_10_DEFAULT, RFC_HIGH}, {-30,0}  },
    {  { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_11_DEFAULT, RFC_HIGH}, {-30,0}  },
    {  { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_12_DEFAULT, RFC_HIGH}, {-30,0}  },
    {  { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_13_DEFAULT, RFC_LOW}, {-30,0}  },
    {  { (int)RFC_WTR1605_CHN3_PA0_SMPS_PDM_DEFAULT, RFC_CONFIG_ONLY}, {-54,-4}  },
    {  { (int)RFC_WTR1605_CHN3_GNSS_BLANK_TDS, RFC_HIGH}, {-6,-3}  },
    {  { (int)RFC_WTR1605_CHN3_INTERNAL_03_DEFAULT, RFC_CONFIG_ONLY}, {-11,-3}  },
    {    { (int)RFC_SIG_LIST_END, RFC_LOW  },  {0, 0}} ,
  },
};
```


Mistake List (2)

- On DIME project, for TDSCDMA QCT default use MIPI ASM not GRFC ASM. So if you use GRFC ASM, you can not use QCT default timing as below

For RX:

Error code

```
rfc_sig_tdd_info_type rf_card_wtr1605_sglte_cmcc_4mode_rx0_tdscdma_b34_sig_cfg =
{
 0x02220204,  /* Revision: v2.34.516 */
 {
  //{ (int)RFC_WTR1605_SGLTE_CMCC_4MODE_TIMING_ASM_CTL,  { RFC_CONFIG_ONLY /*Warning: Not specified*/ , -5 }, {RFC_LOW, -6 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RX_ON0,  { RFC_HIGH, -150 }, {RFC_LOW, 0 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_20,  { RFC_LOW, -25 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_01,  { RFC_HIGH, -25 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_02,  { RFC_LOW, -25 }, {RFC_LOW, -4 } },
  { (int)RFC_SIG_LIST_END,  { RFC_LOW, 0 }, {RFC_LOW, 0 } }
 },
};
```

Right code

```
rfc_sig_tdd_info_type rf_card_wtr1605_sglte_cmcc_4mode_rx0_tdscdma_b34_sig_cfg =
{
 0x02220204,  /* Revision: v2.34.516 */
 {
  //{ (int)RFC_WTR1605_SGLTE_CMCC_4MODE_TIMING_ASM_CTL,  { RFC_CONFIG_ONLY /*Warning: Not specified*/ , -5 }, {RFC_LOW, -6 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RX_ON0,  { RFC_HIGH, -150 }, {RFC_LOW, 0 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_20,  { RFC_LOW, -5 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_01,  { RFC_HIGH, -5 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_02,  { RFC_LOW, -5 }, {RFC_LOW, -4 } },
  { (int)RFC_SIG_LIST_END,  { RFC_LOW, 0 }, {RFC_LOW, 0 } }
 },
};
```


Mistake List (3)

- Tx is same

For TX :

Wrong code:

```
rfc_sig_tdd_info_type rf_card_wtr1605_sglte_cmcc_4mode_tx0_tdscdma_b34_sig_cfg =
{
 0x02220204,  /* Revision: v2.34.516 */
 {...
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_PA0_R0,  { RFC_CONFIG_ONLY, -7 }, {RFC_LOW, -4 } },//PA range
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_PA_ON_16,  { RFC_HIGH, 0 }, {RFC_LOW, 0 } },//PA enable
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_07,  { RFC_HIGH, -25 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_11,  { RFC_LOW, -25 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_20,  { RFC_HIGH, -25 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_01,  { RFC_LOW, -25 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_02,  { RFC_HIGH, -25 }, {RFC_LOW, -4 } },
  //{ (int)RFC_WTR1605_SGLTE_CMCC_4MODE_INTERNAL_GNSS_BLANK_CONCURRENCY,  { RFC_HIGH, -5 }, {RFC_LOW, -6 } },
  //{ (int)RFC_WTR1605_SGLTE_CMCC_4MODE_TX_GTR_TH,  { RFC_CONFIG_ONLY, -5 }, {RFC_LOW, -6 } },
  //{ (int)RFC_WTR1605_SGLTE_CMCC_4MODE_PA_IND,  { RFC_HIGH, -5 }, {RFC_LOW, -6 } },
  { (int)RFC_SIG_LIST_END,  { RFC_LOW, 0 }, {RFC_LOW, 0 } }
 },
};
```

Right code:

```
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_07,  { RFC_HIGH, -5 }, {RFC_LOW, -4 } },
 { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_11,  { RFC_LOW, -5 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_20,  { RFC_HIGH, -5 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_01,  { RFC_LOW, -5 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_02,  { RFC_HIGH, -5 }, {RFC_LOW, -4 } },
```

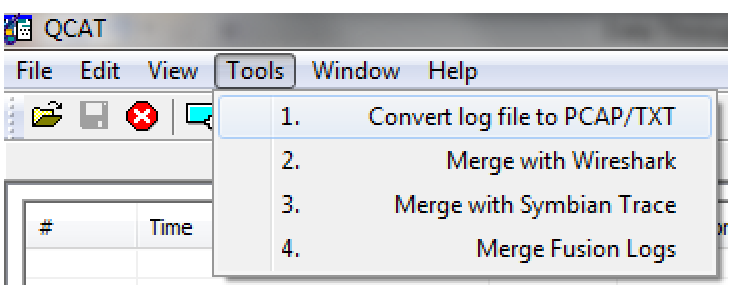
Our Rule

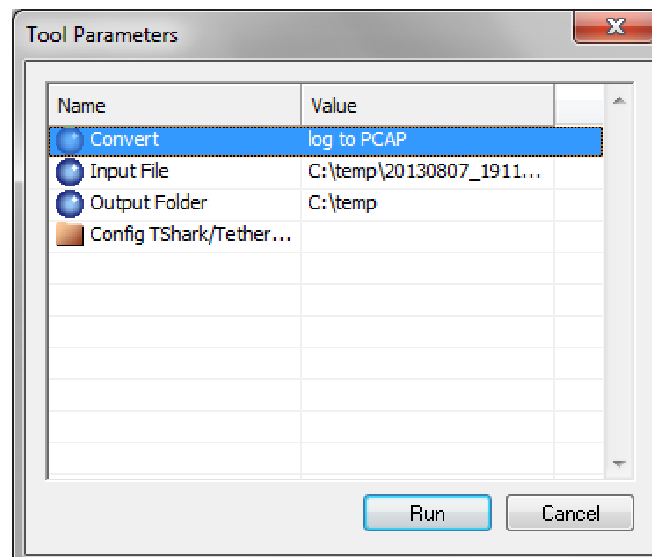
- In principle, these parameter don’t need change, because MTP with default parameter can pass all RF HW test case.
- if customer must change it due to customer’s components are different with QCT, It is better change it slightly(1 to 2 unit) and must do below roll-back test to verify no side-effect.
- Roll-back test1: TD-HSDPA throughput(Tx2Rx3,4,5) test on 8820C to check if have no side effect.
- Roll-back test2: Tx2 EVM, and Rx3,4,5 sensitivity.
  Tx configure at slot2, Rx configure at slot3,4,5

Appendix A: How to Capture Logs

Capturing TCP/IP/UDP/ICMP Packets

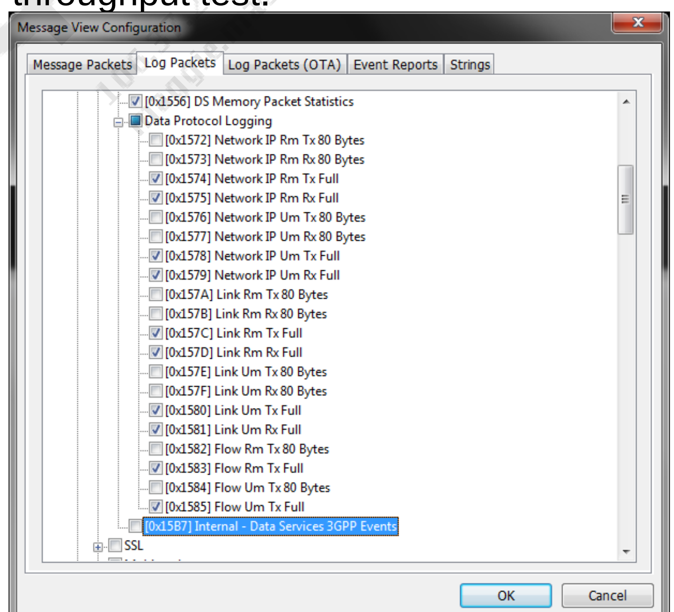
- Pcap log on the PC side is obtained by using Wireshark. Please enable Wireshark on the PC and monitor the traffic on the USB interface (Virtual Ethernet Adapter/PPP).
- If the test is on an Android device, run tcpdump on adb shell to capture pcap of rmnet/wlan device.
  - Command: adb shell tcpdump -i any -s 0 -w /data/tcpdump.pcap
- Pcap log on the UE side is extracted from the QXDM Log. But you must enable LOG PACEKT of Data Protocol Logging (DPL).
  - Converting tool: QCAT/Tools/Convert log file…
  - As shown in the following:





How to Capture DPL log

- DPL Logging
  - QXDM Log mask path: `Message View/Known Log Items/Common/Data Services/Data Protocol Logging`
  - DPL full packet logging: To capture all the bytes of the data packet (e.g.1500 bytes)
    - This is used when you want to check the whole packet. For example, when you need to track the complete HTTP data flow.
    - If the traffic volume is huge, the throughput will be affected since the log is big.
  - DPL packet headers logging: To capture the header of the data packet (e.g. 80 bytes)
    - Since the size of a TCP/IP header is usually 40 bytes, capturing the first 80 bytes is enough. This method has less impact on high throughput test.



Appendix B: ping Packet Test

Ping Packet Test

- Ping packet test is easy and useful.
  - It includes Ping Request/Ping Reply.
  - The sender keeps transmitting no matter whether a reply is received.
  - It can test RTT from the test client to the server.
  - This test can be implemented on both PC and ADB shell.

- If Data Stall occurs, the ping packet test can check whether the lower data link is normal. Typical cases:
  - **Phenomenon**: Some websites cannot be opened by using the Android Browser. For example: http://m.baidu.com.
  - **Test**: Perform the ping packet test when a website cannot be opened. If all the ping packets receive replies, the lower data link should be normal. Then, use Wireshark to analyze data flow of the opened website. It is found that the browser does not transmit data packets after the URL is re-directed to another URL. The website can be opened by using a third-party browser.
  - **Conclusion**: There is a problem of URL redirection in the Android browser. The ping packet test efficiently ruled out the lower data link problem.

- When the data throughput does not meet the requirement, keep the ping packet test while doing the download.
  - When data amount is big, the analysis of TCP data flow is time-consuming but ping packet test is relatively easy.
  - You can check whether the ping packet delay is long and whether there is packet loss.
  - You can compare ping packets with packets of PDCP/RLC/MAC layer. In this way, you can easily check delay or packet loss in each layer. It is difficult to check TCP when data throughput is high.

Appendix C: TCP Protocol Setting/Key Algorithm Analysis

Checking TCP Parameter Setting

- TCP window size must be larger than Bandwidth Delay Product (BDP) of the DUT.
  - TCP_WinSIZE>=Bandwidth X RTT(BDP)
  - WinXP – HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters->TcpWindowSize
  - Linux – TCP window size can be set in /etc/sysctl.conf by adding:
    - net.ipv4.tcp_rmem = <min> <default> <tcp max>
    - net.ipv4.tcp_wmem = <min> <default> <tcp max>

- Set IP Maximum Transmission Unit (MTU) to 1500: MTU ≈ TCP MSS + TCP Hdr + IP Hdr

- Check the value of TcpAckFrequency. (The default value is 2.)

Linux TCP Setting

- Get the default TCP Settings
  - cat /proc/sys/net/core/rmem_max
  - cat /proc/sys/net/core/wmem_max
  - cat /proc/sys/net/core/rmem_default
  - cat /proc/sys/net/core/wmem_default
  - cat /proc/sys/net/ipv4/tcp_rmem
  - cat /proc/sys/net/ipv4/tcp_wmem
  - cat /proc/sycat /proc/sys/net/ipv4/tcp_window_scalings

- Tune the TCP parameters for peak data throughput test
  - echo 4194304 > /proc/sys/net/core/rmem_max
  - echo 4194304 > /proc/sys/net/core/wmem_max
  - echo 2097152 > /proc/sys/net/core/rmem_default
  - echo 2097152 > /proc/sys/net/core/wmem_default
  - echo 524288 2097152 4194304 > /proc/sys/net/ipv4/tcp_rmem
  - echo 524288 2097152 4194304 > /proc/sys/net/ipv4/tcp_wmem
  - echo 1 > /proc/sys/net/ipv4/tcp_window_scaling

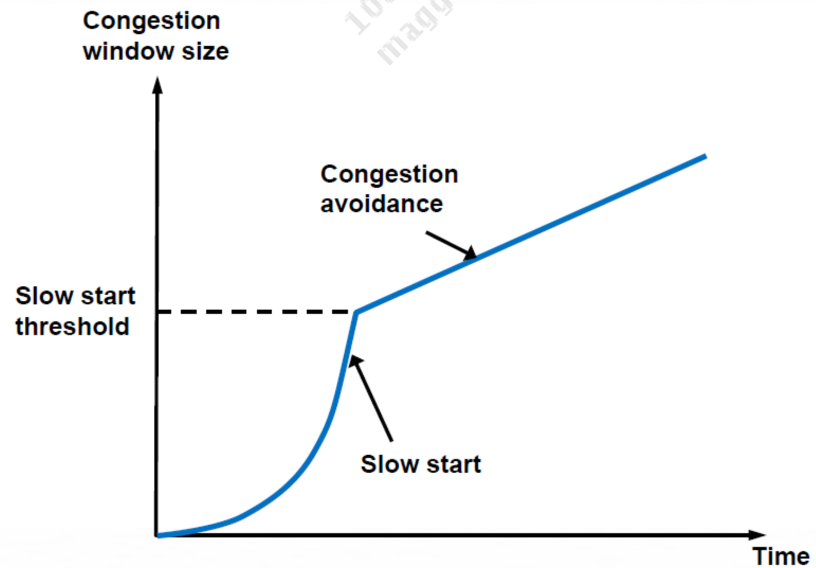
  - Note: above values are only for debug, the value may be changed as per needed.

TCP Characteristics

- TCP is a reliable transmission protocol and controls the sending/receiving by using sliding windows.
- TCP is based on the confirmation mechanism of ACK. If any data segment is lost, the receiver will send duplicated ACK (DupACK).
- Key TCP window concepts:
  - rx_win
    - “Receiving Window” of the receiver: Included in each TCP ACK data packet. Its size is dynamically adjusted based on the receiver load and capability (flow control).
  - cwnd
    - Congestion Window: Dynamically adjusted based on the data flow.
  - tx_win
    - “Sending Window” of the sender: A local variable of TCP sender. The receiver cannot obtain its value.
    - tx_win = MIN(rx_win, cwnd)

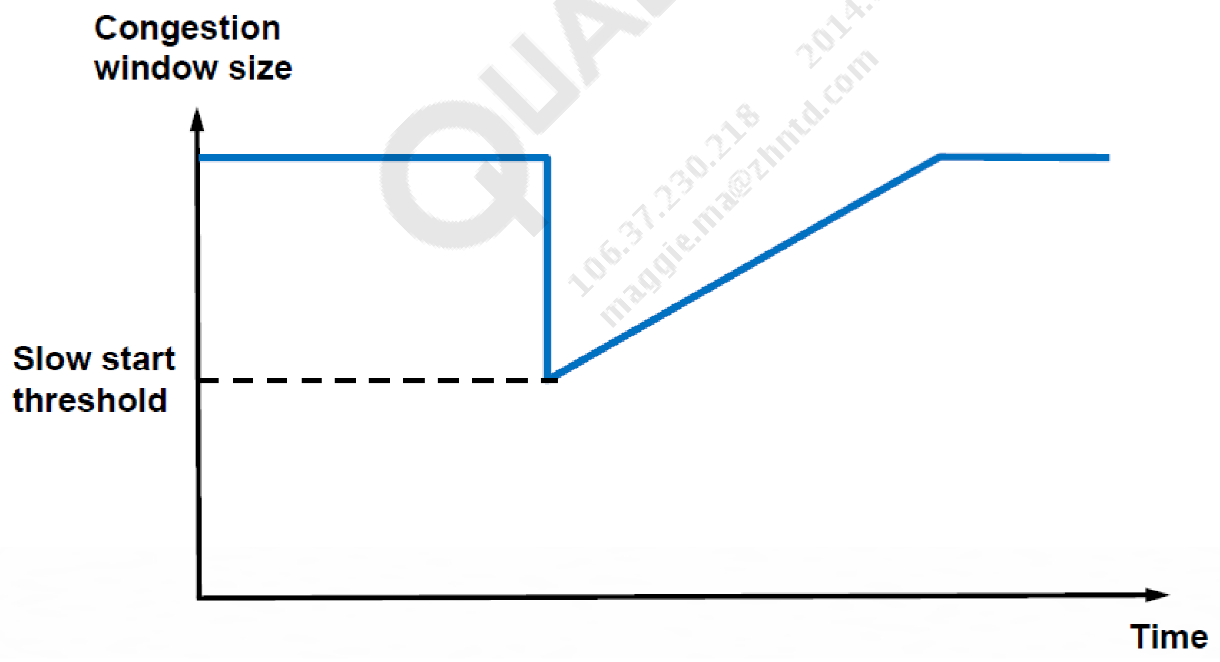
TCP Congestion Control

- The TCP Congestion Window Size (cwnd) is dynamically adjusted based on the current channel conditions.
  - Slow Start stage: cwnd increases exponentially when it is lower than Slow Start Threshold (ssthresh).
  - Congestion Avoidance stage: cwnd increases in a linear way when it is higher than Slow Start Threshold (ssthresh).
  - Once congestion occurs because of channel change, set the value of tx_win to half of the current value.
- The following figure shows the algorithm of the above two stages.



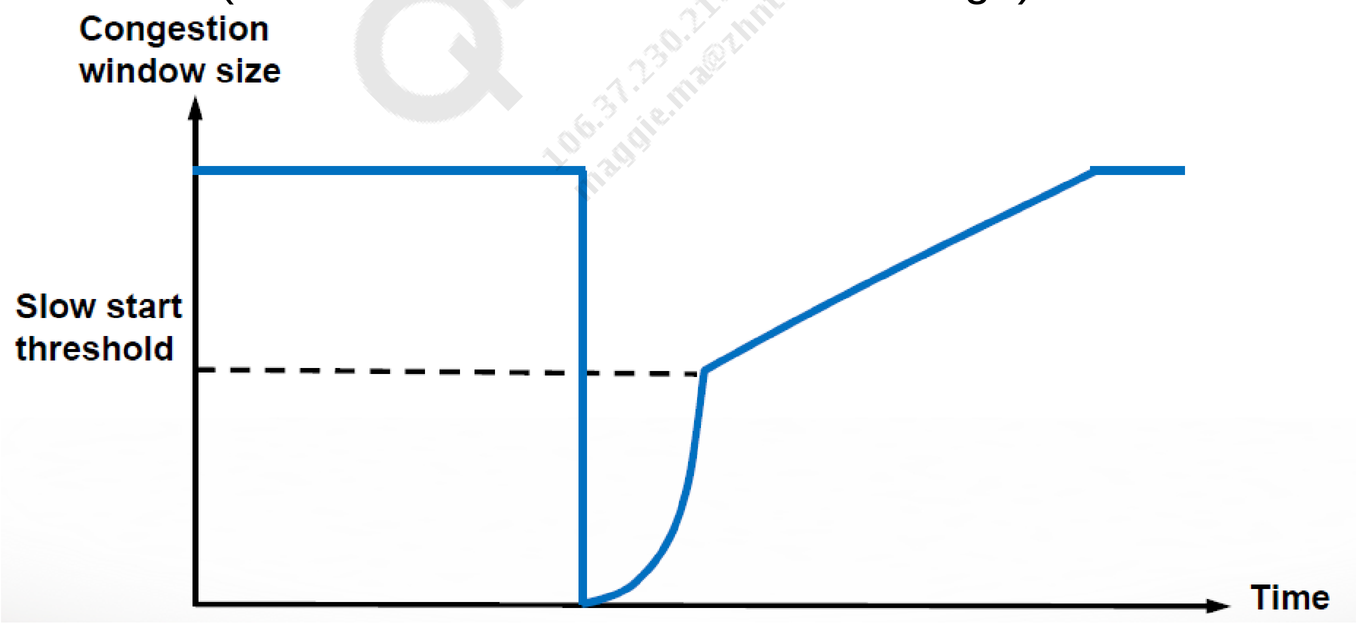
TCP Congestion Control – Fast Retransmission

- When the sender continuously receives three ACK(DupACK) of the same sequence number, the sender should immediately re-transmit the TCP segment and set the following parameters:
  - Reduce *ssthresh* to half of *tx_win*;
  - Set *cwnd* to *ssthresh* + 3 segment_size



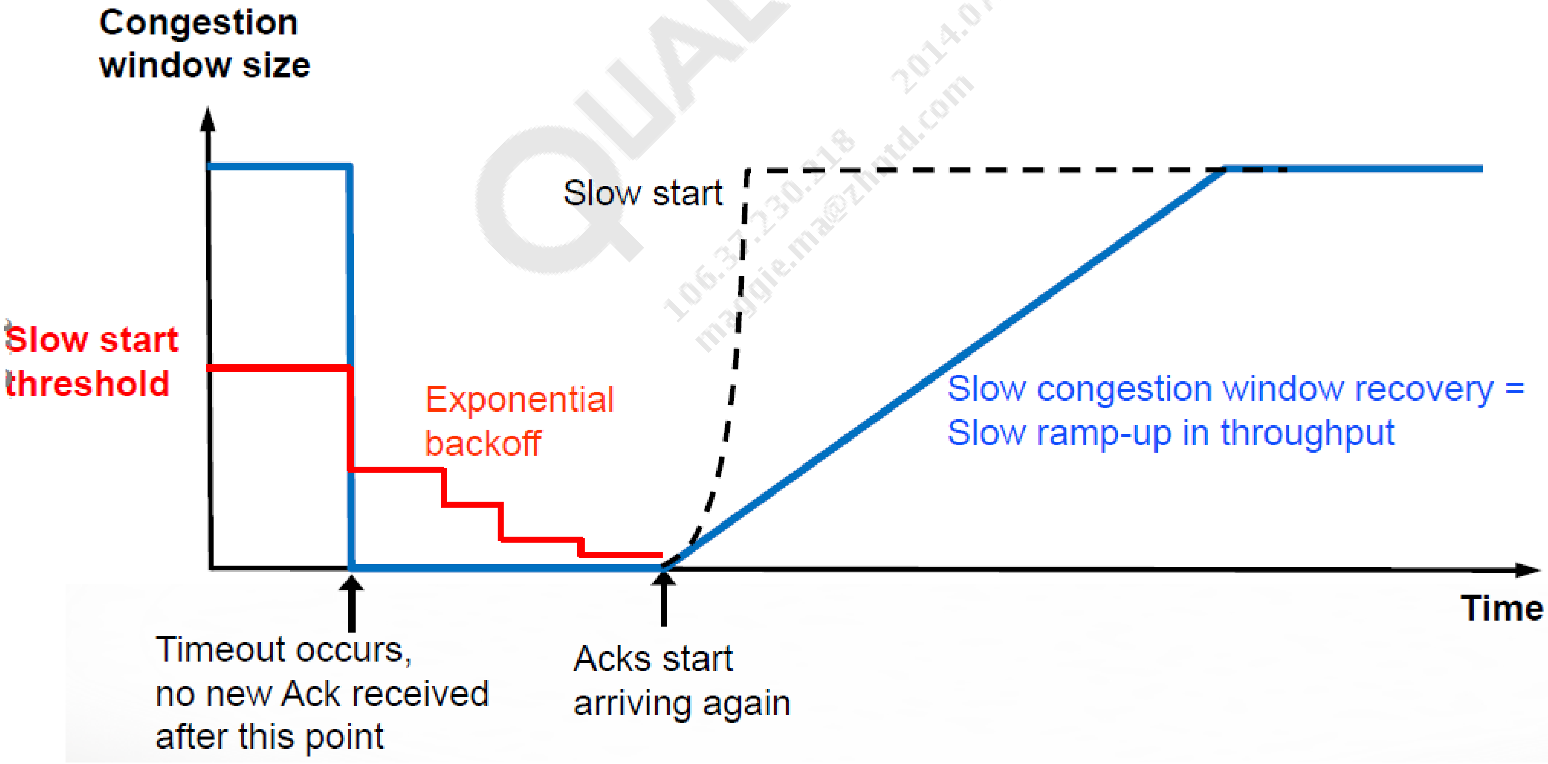
TCP Congestion Control – RTO

- If the sender does not receive the ACKs of continuous sequence numbers within ReTransmission Timeout (RTO), it will re-transmit the lost TCP segment. The RTO is calculated as follows:
  - RTO = A + 4D (A – Average RTT, D – Mean deviation)
- When timeout occurs, set the TCP parameters:
  - Set **ssthresh** to half of **tx_win**.
  - Set **cwnd** to 1 (TCP will re-enter the slow start stage)



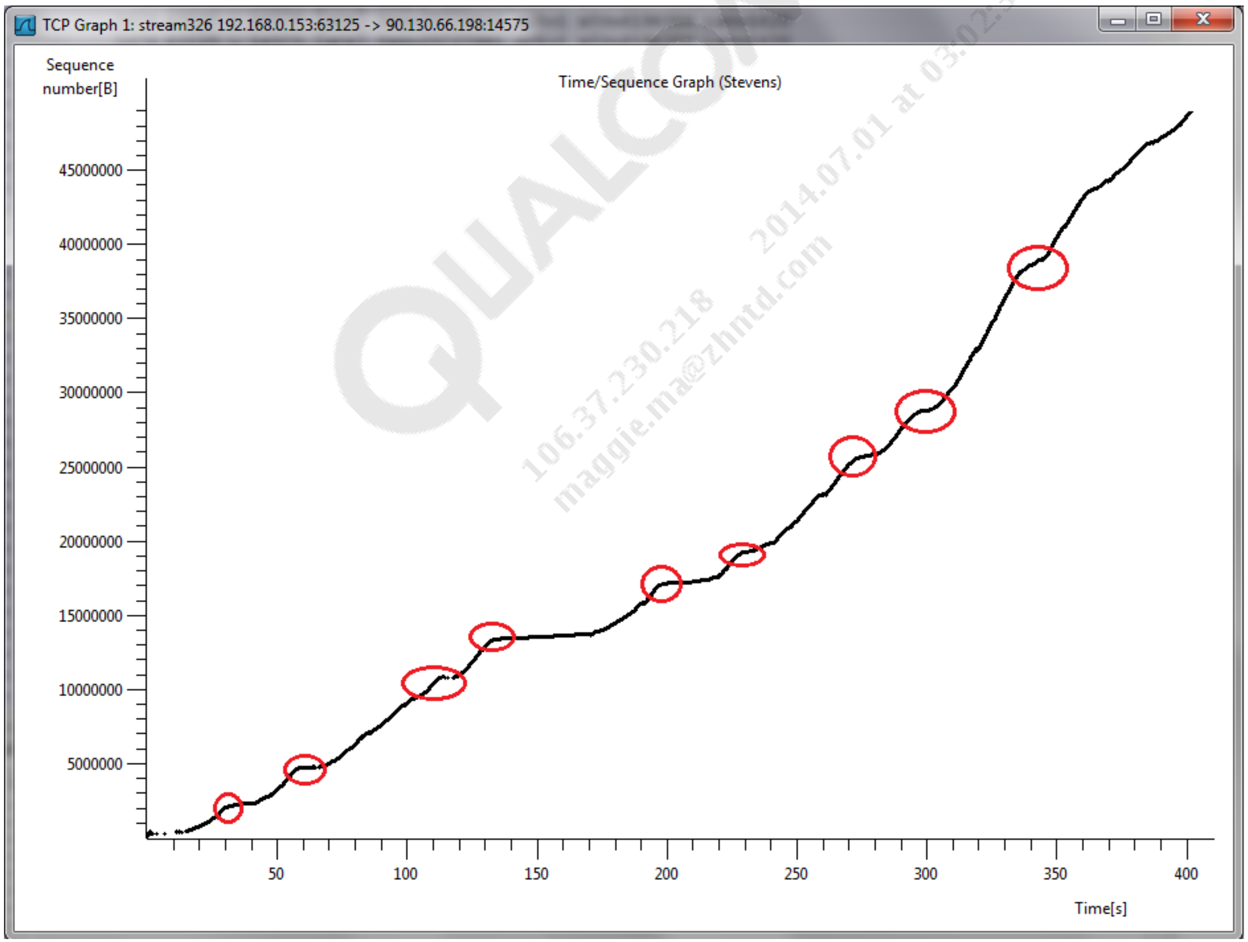
TCP Congestion Control–Several Continuous TCP Packet Loss

- If several continuous TCP packets are lost, the value of **ssthresh** will quickly decrease to a low value.
  - At each RTO, the value of ssthresh will decrease by half (exponential backoff).
  - At the same time, TCP flow quickly decreases to a low level and it takes a long time for this to recover.



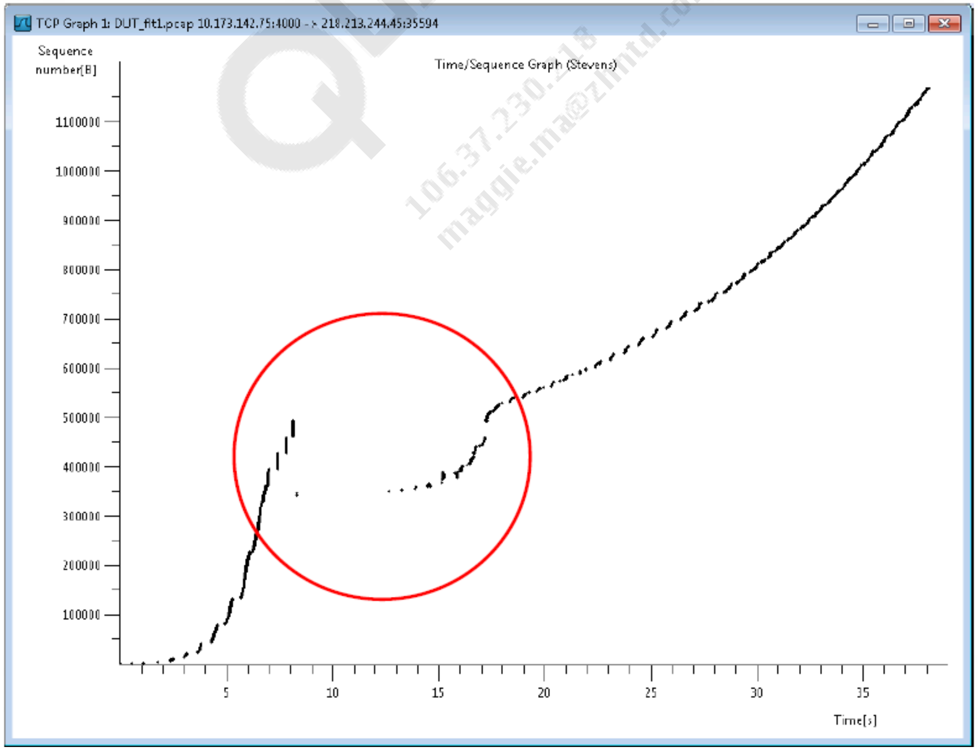
TCP Congestion Avoidance Shown in Wireshark – Fast Retransmission

- TCP Fast Retransmission can quickly recover the TCP congestion and prevent the TCP flow from fast dropping to a low level.



TCP Congestion Avoidance Shown in Wireshark – RTO

- RTO will quickly reduce the TCP flow to a low level.
- If TCP packet loss occurs frequently, it indicates that the network is congested or channel quality is poor. Then, TCP should enter exponential backoff , and TCP flow will decrease to a very low level and it takes a long time for this to recover.
- As a result, the frequency of TCP packet loss is an important indicator of the lower channel quality.



TCP Bidirectional TCP Problems

- Problem 1: During bidirectional (upload+download) data transmission, ACK packets and Data packets are transmitted together. If ACK packets and Data packets have the same priority, then the delay of ACK packets during bidirectional data transmission is longer than that during un-directional data transmission, especially when the asymmetrical uplink and downlink bandwidth.
- Problem 2: Uplink data flow control may cause ACK Compression. The causes are as follows:
  - When uplink data transmission is too quick to exceed the maximum level of the Um interface Watermark, data flow control is triggered.
  - At this time, ACK packets will accumulate at Watermark of the Rm interface.
  - Data flow control stops after the data at the Um interface is transmitted and the low level of Watermark is reached.
  - Then, the large amount of ACK packets buffering at the Rm interface are quickly transmitted to the network through the Um interface.
  - When the network receives multiple ACK packets at the same time, the TCP server will send a large amount of downlink TCP packets. With this flooding on the downlink network, some TCP packets may be lost and throughput will be affected.

Bidirectional TCP Solution – Prioritize TCP Acks

- To solve Problem 1, the mechanism to prioritize uplink TCP ACK packets is used on the UE side.
  - The feature is defined as FEATURE_DATA_PS_TCP_ACK_PRIO.
  - In the Watermark queue, the priority of TCP ACK packets is higher than that of TCP Data packets.
  - To enable this feature, set NV 67208 to 1 (Default) .

Bidirectional TCP Solution – Back Pressure Removal

- To solve Problem 2, Back pressure removal is used on the UE side.

- Disable UL data flow control.
  - It is periodically checked whether there are uplink TCP ACK packets. This is to confirm whether there is downlink TCP data flow.
  - If there are downlink data packets, uplink data flow control is disabled. In this way, ACK Compression is prevented.

- Enable UL data flow control.
  - It is periodically checked whether there are downlink TCP ACK packets. If there are none, uplink data flow control is enabled.
  - Set NV66051(Throughput Optimization Parameters) as follows:
    - back_pressure_removal = 1 (enable)
    - tx_timestamp_timer_val = 2 (seconds)

Appendix D: UDP and iperf Test

UDP and Iperf Test

- TCP flow control mechanisms and reliability transmission ACK mechanisms are comparatively complex while UDP is comparatively simple:
  - Acknowledgment is not required from peer; no ACK mechanism.
  - Can transmit data flow through any specified bandwidth by using tools.

- Based on UDP characteristics, Iperf can be used to do the test. The goal is to verify whether the bandwidth of the physical channel is enough.
  - If the Iperf UDP test result does not meet the requirement, then TCP cannot meet the requirement. In this case, the problem may be the physical channel or that the bandwidth of the network is not enough.
  - If the Iperf UDP test result meets the requirement while TCP does not, the problem cannot be with the physical layer. Check the upper-layer data services.

For details about how to use Iperf, see 80-N2363-1 (Appendix. C).

Iperf Test Case on Android

- Goal: To test the downlink data throughput of WiFi-Tethering

- Install iperf into the Android device.
  - ```
    c:\adb>adb root
    c:\adb>adb remount
    c:\adb>adb push iperf /etc/iperf
    c:\adb> adb shell chmod 777 /etc/iperf
    ```

- Run the iperf TCP client on the Android device.
  - ```
    c:\adb>adb shell
    # /etc/iperf -c <IP-address of PC client> -w 2M -t 60 -i 1
    ```

- Run the iperf server on the Laptop to test the download throughput of WiFi-Tethering.
  - ```
    c:\iperf>iperf -s -w 2M -i 1
    ```
    ```

Test Cases

- Use WiFi-Tethering and test the download speed. After 60 seconds, the average throughput counted by iperf is 45.4 Mbps.

```
C:iperf>iperf -s -w 2M -i 1
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 2.00 MByte
------------------------------------------------------------
[264] local 192.168.43.39 port 5001 connected with 192.168.43.1 port 60232
[ ID] Interval       Transfer     Bandwidth
[264]  0.0- 1.0 sec   539 KBytes  4.41 Mbits/sec
[264]  1.0- 2.0 sec   309 KBytes  2.53 Mbits/sec
[264]  2.0- 3.0 sec  2.73 MBytes  22.9 Mbits/sec
[264]  3.0- 4.0 sec  4.44 MBytes  37.2 Mbits/sec
[264]  4.0- 5.0 sec  5.82 MBytes  48.8 Mbits/sec
...
[264] 54.0-55.0 sec  5.21 MBytes  43.7 Mbits/sec
[264] 55.0-56.0 sec  4.84 MBytes  40.6 Mbits/sec
[264] 56.0-57.0 sec  5.22 MBytes  43.8 Mbits/sec
[264] 57.0-58.0 sec  4.72 MBytes  39.6 Mbits/sec
[264] 58.0-59.0 sec  5.68 MBytes  47.6 Mbits/sec
[264] 59.0-60.0 sec  5.25 MBytes  44.0 Mbits/sec
[ ID] Interval       Transfer     Bandwidth
[264]  0.0-60.0 sec   325 MBytes  45.4 Mbits/sec
```

Appendix E: CPU Performance Setting

CPU Performance Setting

- The default CPU running mode is “on-demand”, the CPU frequency is changed according to loading.
- For peak data throughput test, sometimes need to set the CPU to max speed.
- Set the CPU to “Performance” mode, to get max performance.
  - Cool the device (turn off Data Service/LCD etc.)
  - Run the following commands in ADB shell (Qual-Core example)
    - `su`
    - `stop mpdecision`
    - `stop thermal-engine`
    - `echo 1 > /sys/devices/system/cpu/cpu1/online`
    - `echo 1 > /sys/devices/system/cpu/cpu2/online`
    - `echo 1 > /sys/devices/system/cpu/cpu3/online`
    - `echo "performance" > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor`
  - Check if governor was changed to "performance" mode
    - `cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor`

CPU Usage (1)

- Get the CPU Usage, and see if the CPU is almost running at 100% usage. If yes, check why the CPU usage is too high.

- Get the CPU usage during DL/UL Tput test
  - `top -d 1 -n 5`

- Example: CPU usage is almost 100%

```
# top -d 1 -n 5
top -d 1 -n 5
[Mem: 37512K used, 131160K free, 0K shrd, 620K buff, 8548K cached
CPU:  0.0% usr 58.3% sys  0.0% nic  0.0% idle  0.0% io  0.0% irq 41.6% sirq
Load average: 3.84 1.84 0.72 3/208 1112
[7m  PID  PPID USER     STAT   VSZ %MEM CPU %CPU COMMAND[0m
   32     2 root     RW       0  0.0   0 23.0 [kworker/u:2]
   13     2 root     RW       0  0.0   0 23.0 [kworker/0:1]
   39     2 root     SW       0  0.0   0 23.0 [kworker/u:3]
    3     2 root     RW       0  0.0   0 15.3 [ksoftirqd/0]
 1112   994 root     R     2188  1.3   0  7.6 top -d 1 -n 5
    6     2 root     RW       0  0.0   0  7.6 [kworker/u:0]
  324     1 root     S <   189m114.6   0  0.0 /usr/bin/thermal-engine
319     1 root     S    77624 45.9   0  0.0 /usr/bin/qmuxd
…
```

CPU Usage

- Get the overall CPU usage during DL/UL Tput test
  - `mpstat 2 30` // this is busybox command

- Example: Print the CPU usage per 2s, only ~2% idle.

```
# mpstat 2 30
mpstat 2 30
Linux 3.4.0+ (mdm9x25)  01/06/80        _armv7l_        (1 CPU)

00:04:06     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
00:04:08     all    0.00    0.00   40.00    0.00    0.00   58.00    0.00    0.00    2.00

00:04:08     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
00:04:10     all    1.00    0.00   42.79    0.00    0.00   53.23    0.00    0.00    2.99

00:04:10     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
00:04:12     all    0.50    0.00   40.20    0.00    0.00   57.79    0.00    0.00    1.51

00:04:12     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
00:04:14     all    0.00    0.00   39.90    0.00    0.00   58.08    0.00    0.00    2.02

00:04:14     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
00:04:16     all    0.00    0.00   46.23    0.00    0.00   51.76    0.00    0.00    2.01
```

Questions?

https://support.cdmatech.com

目录

- 1. 准备工作和基本步骤
- 2. Data Service层分析
- 3. CDMA 1x/EVDO层分析
- 4. WCDMA/HSPA层分析
- 5. TD-SCDMA/HSPA层分析
- 6. LTE层分析
- 7. RFSW分析
- 附录

1. 准备工作和基本步骤

准备工作

准备如下分析工具：

- QXDM ；

  用QXDM回放QXDM LOG，观察物理层View，可以得知物理层速率。例如LTE的观察：

  - LTE ML1 DL Throughput and BLER
  - LTE ML1 UL Throughput and BLER

- Wireshark

  WIRESHARK LOG可以看出TCP/IP层的速率和数据包发送接收情况

  - PC 侧的WIRESHARK LOG（推荐）
  - UE侧的WIRESHARK LOG

- DU Meter

  这类工具安装到PC上，可以观测PC端口的速率

排查基本步骤

1. 确定吞吐量的目标值，最好有商用的参考机(REF)在相同环境下进行对比，以便进行详细对比分析，从而收窄可能出现问题的模块；

2. 先从应用层(HTTP/TCP/IP)查看，使用Wireshark查看数据TCP/IP层吞吐量是否达标，是否有TCP报文丢包，是否有较多重传；

3. 如果TCP层有较多丢包或重传，查看Data Service层是否有流控频繁启动或丢包。如果没有，则继续往底层(RLC/MAC/PHY)查看。检查信道质量、误码率、网络调度数据等；

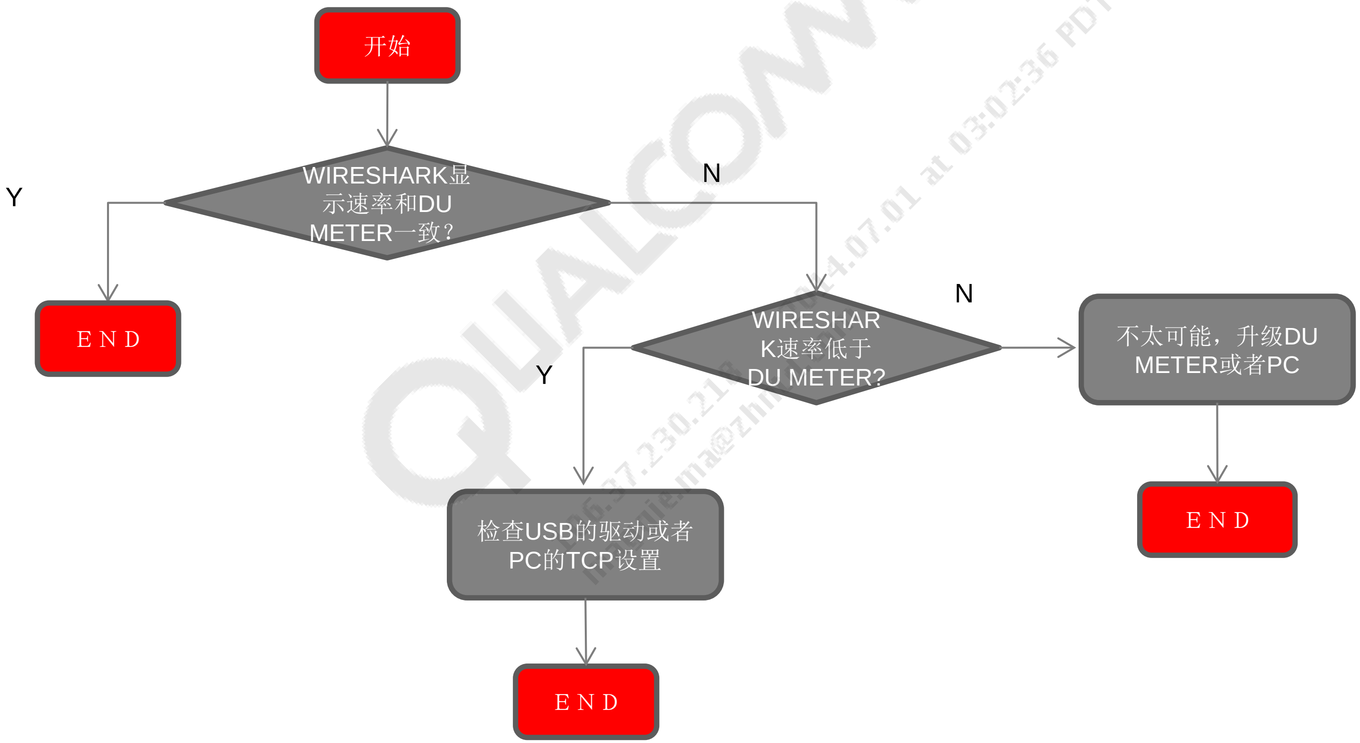
4. 必要时，在DUT和REF使用ping包来检查RTT时延，DUT是否比REF的时延大、甚至出现ping包超时。相对于TCP，查ping包(ICMP)问题相对容易些；

5. 如果怀疑底层带宽是否足够，可以使用iperf工具进行UDP测试。从而确定问题是在底层还是上层。

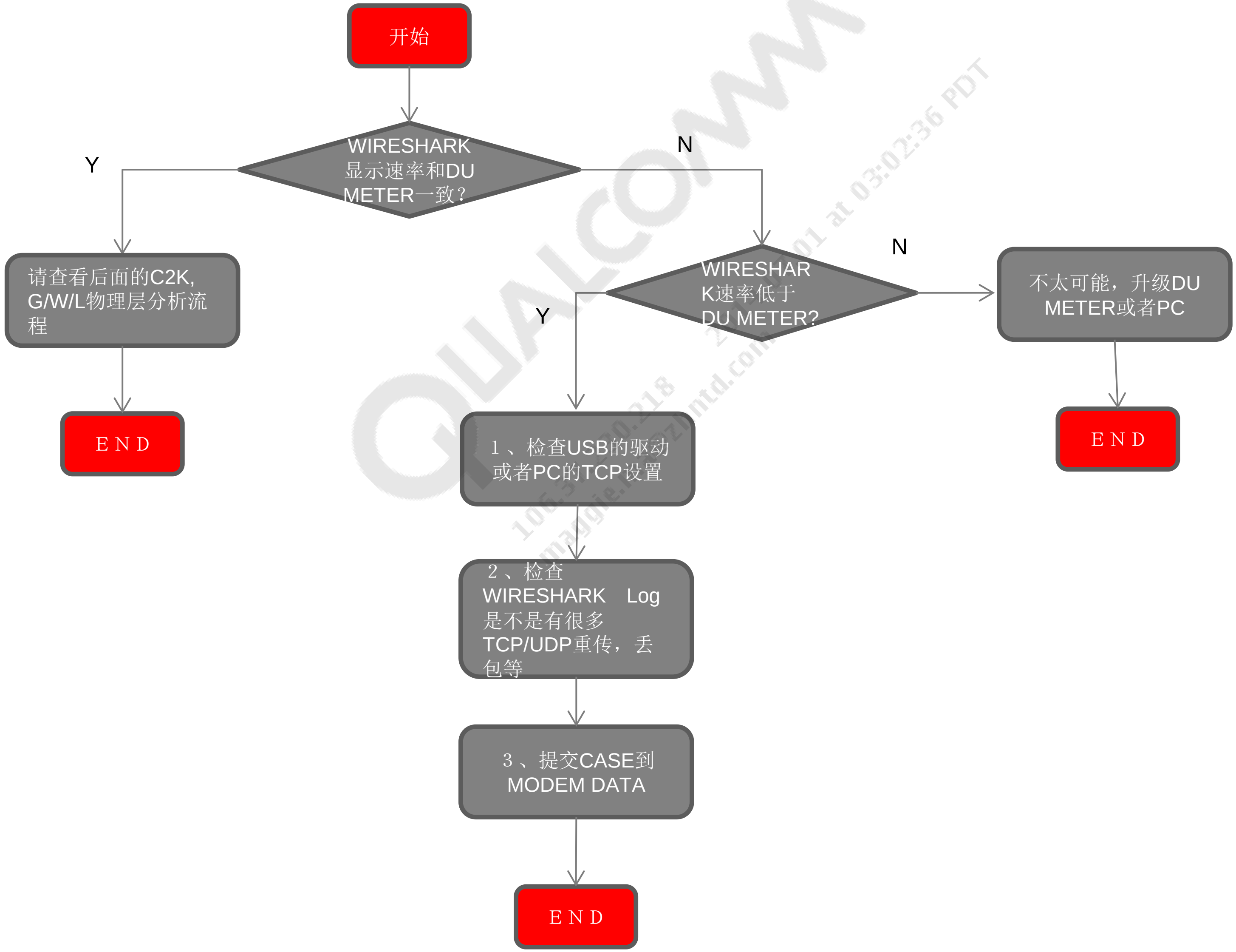
2. Data Service层分析

2.1 Wireshark工具分析

PC/Android侧Log的Wireshark检查流程

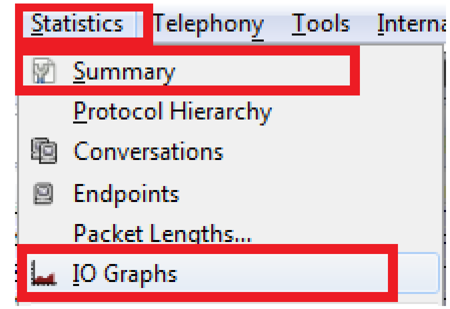


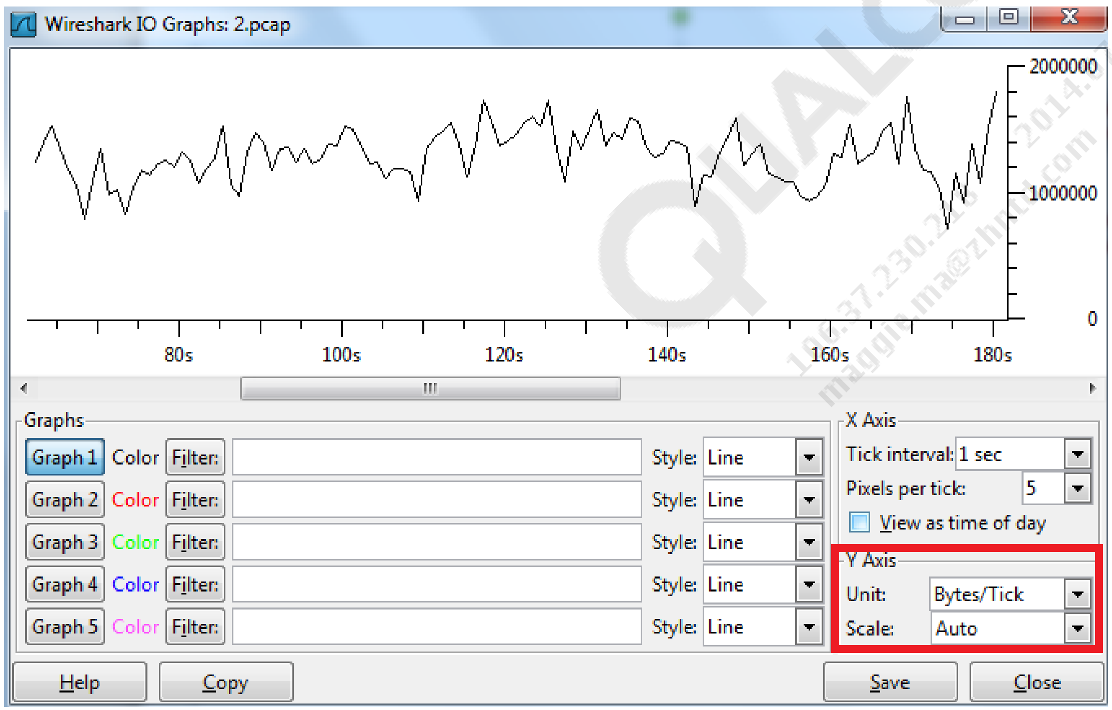
UE侧Log的Wireshark检查流程



使用Wireshark工具分析报文

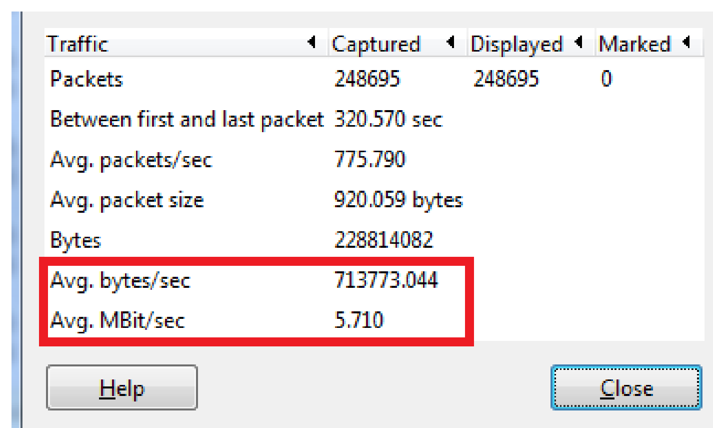
- 首先用Wireshark打开pcap log
- 点击“Statistics->IO Graphs”或者“Summary”
- 获得数据吞吐量的基本情况：速率快/慢、波动大/小等





IO Graphs

Summary



Wireshark的IO Graphs更多组合用法

- 可以使用多个过滤条件，绘制不同的曲线对比。例如下图对比得知：
  - 黑色曲线：上图(WiFi)的吞吐量波动率，较下图(USB)的大；
  - 红色曲线：上图(WiFi)的重复ACK数量，较下图(USB)的多

Wireshark TCP报文Log的实例分析

- 检查TCP数据包丢失

2840	21:30:34.211	90.130.66.198	192.168.0.153	TCP	103	ftp > 63066 [PSH, ACK] Seq=293 Ack=83 Win=32120 Len=49
2841	21:30:34.212	192.168.0.153	90.130.66.198	TCP	60	63066 > ftp [PSH, ACK] Seq=83 Ack=342 Win=64979 Len=6
2879	21:30:34.531	90.130.66.198	192.168.0.153	TCP	54	[TCP Dup ACK 2840#1] ftp > 63066 [ACK] Seq=342 Ack=83 Win=32120 Len=0
2884	21:30:34.621	90.130.66.198	192.168.0.153	TCP	54	ftp > 63066 [ACK] Seq=342 Ack=89 Win=32114 Len=0
2892	21:30:35.472	90.130.66.198	192.168.0.153	TCP	78	[TCP Previous segment lost] ftp > 63066 [PSH, ACK] Seq=381 Ack=89 Win=32120 Len=24
2893	21:30:35.472	192.168.0.153	90.130.66.198	TCP	54	[TCP Dup ACK 2841#1] 63066 > ftp [ACK] Seq=89 Ack=342 Win=64979 Len=0
2925	21:30:36.217	90.130.66.198	192.168.0.153	TCP	93	[TCP Retransmission] ftp > 63066 [PSH, ACK] Seq=342 Ack=89 Win=32120 Len=39
2926	21:30:36.217	192.168.0.153	90.130.66.198	TCP	54	63066 > ftp [ACK] Seq=89 Ack=405 Win=64916 Len=0

  - 在2884行, 上次发送序号为#342.
  - 在2892行, 此次发送序号为#381. 得出丢失的TCP报文序号为[342-380]
  - 在2925行，重传TCP序号为#342的报文
  - 在2926行，收到ACK(381+24 = 405)，成功！

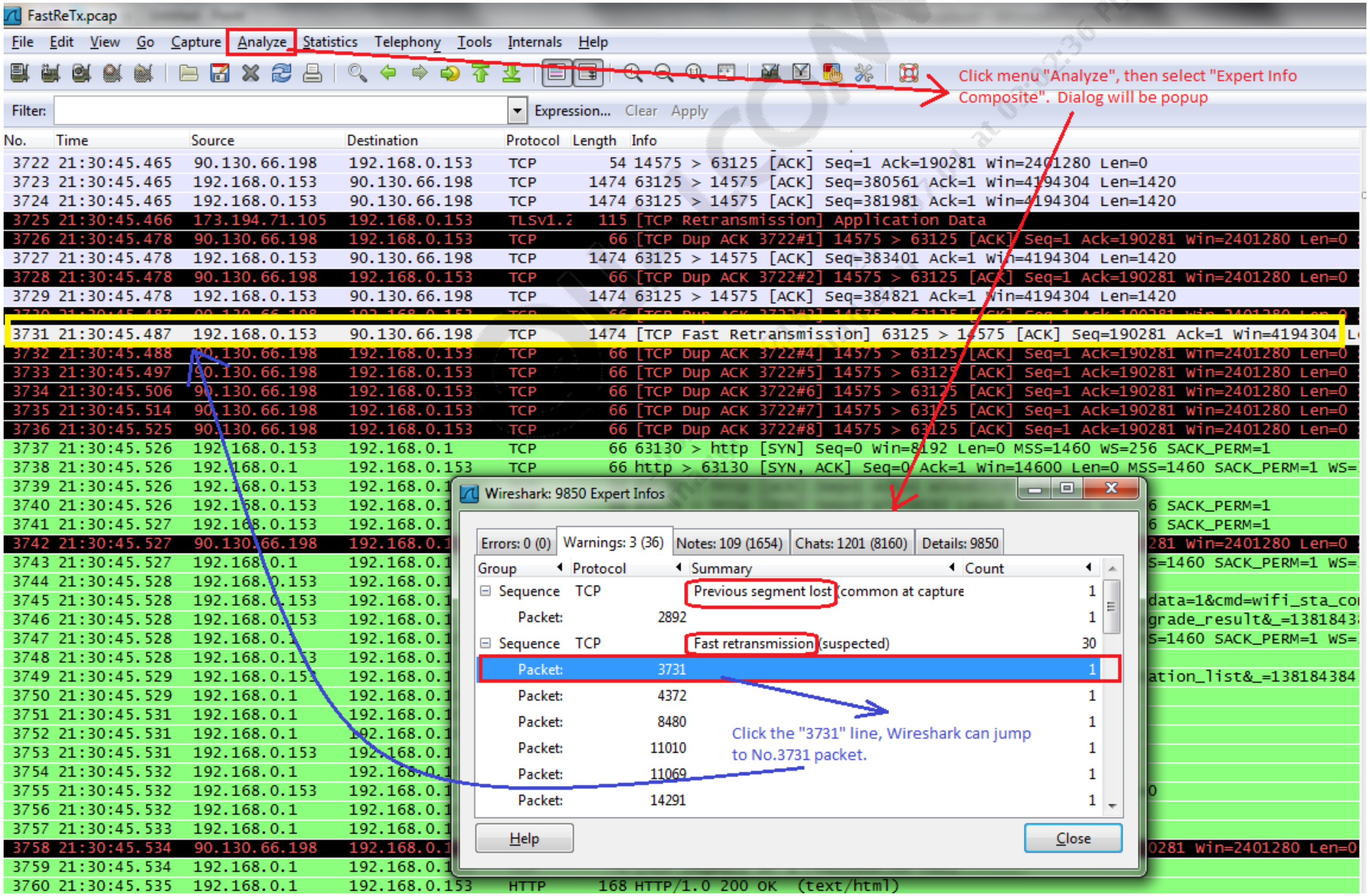
- DupACKs和快速重传

No.	Time	Source	Destination	Protocol	Length	Info
3523	21:30:45.028	192.168.0.153	90.130.66.198	TCP	1474	63125 > 14575 [ACK] Seq=190281 Ack=1 Win=4194304 Len=1420
3722	21:30:45.465	90.130.66.198	192.168.0.153	TCP	54	14575 > 63125 [ACK] Seq=1 Ack=190281 Win=2401280 Len=0
3726	21:30:45.478	90.130.66.198	192.168.0.153	TCP	66	[TCP Dup ACK 3722#1] 14575 > 63125 [ACK] Seq=1 Ack=190281 Win=2401280 Len=0 SLE=195961
3728	21:30:45.478	90.130.66.198	192.168.0.153	TCP	66	[TCP Dup ACK 3722#2] 14575 > 63125 [ACK] Seq=1 Ack=190281 Win=2401280 Len=0 SLE=195961
3730	21:30:45.487	90.130.66.198	192.168.0.153	TCP	66	[TCP Dup ACK 3722#3] 14575 > 63125 [ACK] Seq=1 Ack=190281 Win=2401280 Len=0 SLE=195961
3731	21:30:45.487	192.168.0.153	90.130.66.198	TCP	1474	[TCP Fast Retransmission] 63125 > 14575 [ACK] Seq=190281 Ack=1 Win=4194304 Len=1420

  - 在3523行发送TCP序号为#190281的报文;
  - 在3726/3728/2730 行，收到3个重复 ACK;
  - 在3731行，快速重传TCP序号为#190281的报文 (大约在460ms).
  - 由此看出，不需要等待RTO超时那么长时间，快速重传可以较快地恢复TCP流量，保持吞吐量在较高传输水平。附录C有更详细信息可以参考

Wireshark快速查找TCP丢包/重传

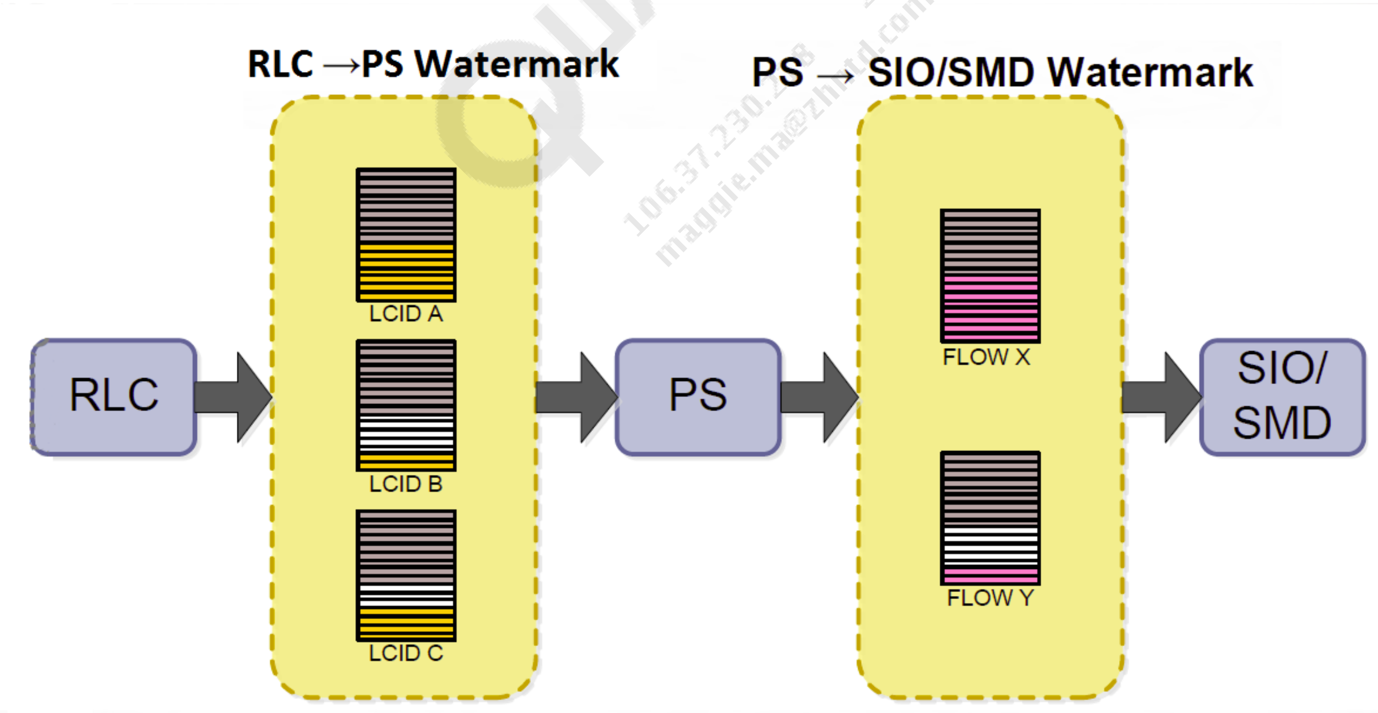
- 如何在Wireshark中快速地找到TCP丢包和重传，如下图所示：



2.2. Watermark流控分析

Watermark-based Flow Control(WM流控)

- Um Watermark
  - 缓存RLC <-> PS的数据包报文
- Rm Watermark
  - 缓存PS <-> SIO/SMD的数据报文
- 如果WM流控频繁启动，数据吞吐量会受影响，需要排查发起流控的原因



Watermark流控实例分析

- 检查QXDM log是否有流控频繁启动：
  - 关键字为 “disabling flow|enabling flow”，如下例：

```
// mask 0x800模块启动流控，关闭数据流(PS不能将数据送到Um Watermark)，时间持续了1.4s
Data Services/Medium        08:59:54.933        ps_phys_link.c  00864  client 0x800 disabling flow on phys link
0x02BDC4F8 -> mask 0x800

Data Services/Medium        08:59:56.316        ps_phys_link.c  00793  client 0x800 enabling flow on phys link
0x02BDC4F8 -> mask 0x0

// 通过mask = 0x800确认是Low Layer(LL)启动的流控，需要从RLC层查此问题
#define DS_FLOW_LL_MASK        0x00000800  /**< Mask for all LL events. */
```

- 检查QXDM log是否有Watermark满消息：
  - 关键字为“WM full,freeing packet ”，如下例：

```
Data Service Memory Buffers/Error      08:59:55.133       dsm_queue.c  00487  WM full,freeing packet
0x2261d6c:Watermark 0x2bdc720:Tried 1460

Data Service Memory Buffers/Error      08:59:55.133       dsm_queue.c  00487  WM full,freeing packet
0x2261d6c:Watermark 0x2bdc720:Tried 1460
...   // 此处发生多个WM full, freeing packet消息

Data Service Memory Buffers/Error      08:59:55.994       dsm_queue.c  00487  WM full,freeing packet
0x226200c:Watermark 0x2bdc720:Tried 1460
```


3. CDMA1x/EVDO层分析

EV-DO数据吞吐量

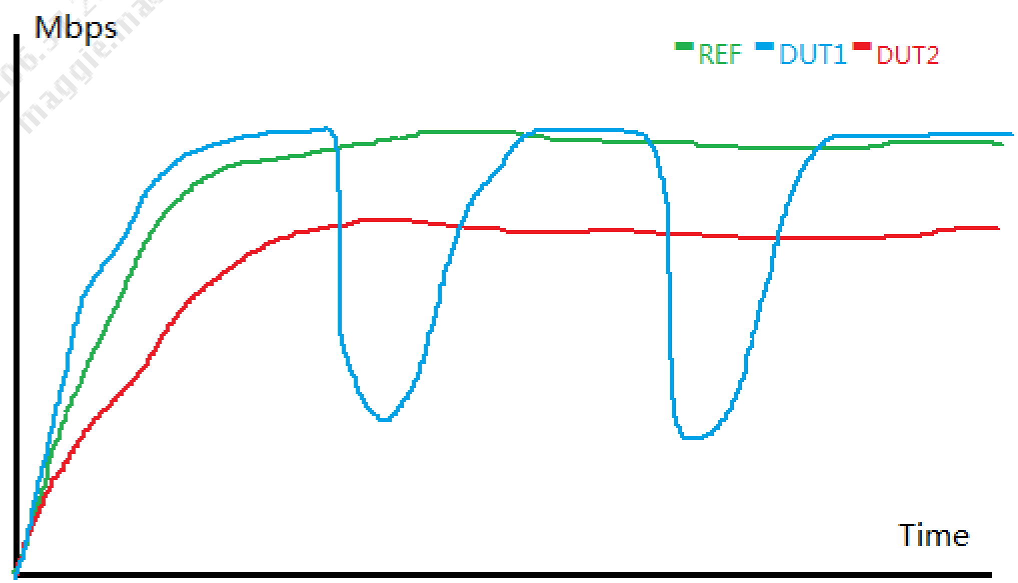
- EV-DO数据吞吐量排查问题清单：
  - 80-N2363-1：DOrA_Data_Tput_Chklst_Troubleshoot
  - 遇到EV-DO吞吐量问题，请先按照此文档列表仔细自查。如果仍然没有解决问题，请提交case

- 在CT测试时，按照如下设置进行测试.
  - Spirent APEX测试仪器的64 K Send buffer size设置：
    - Client Buffer Size Configurable True
    - Client Send Buffer Size(*1460) 44
    - Client Receive Buffer Size(*1460) 44
  - TCP ACK prioritization
    - NV#67208 to 1 (Enabled)
  - BP Removal ON
    - NV#69739 to 1, 1000
  - Network model call
    - AT+CRM=2

4. WCDMA/HSPA层分析

WCDMA/HSPA 数据吞吐量问题分析

- WCDMA/HSPA 数据的吞吐量测试依赖于网络环境，所以在测试前一定要清楚自己的目标速率。相同的测试机器在不同的测试环境下，速率表现会相差很远。因此分析问题的时候通常需要和参考手机做对比。
- 通常会有两类速率问题
  - 测试机(DUT1)在整个测试过程中都比对比机(REF)差
  - 测试机(DUT2)大多数时候都和对比机(REF)差不多，有时候会差一点
- DUT2处理比较简单，找到速率变低的时间点然后从TCP -→RLC-→Physical 检查原因
- 这个文档主要通过分析DUT1的现象来了解怎么样从TCP到物理层来分析吞吐量问题



4.1 RLC层分析

检查RLC层 (1)

- 如果测试机比参考机UDP测试差，那么需要进一步分析WCDMA RLC 层

1. 检查RB 速率是否满足要求

2013 Jul 18 12:22:00.429 [1C] 0x413A WCDMA Radio Bearer Rates

Number of User Plane Radio Bearers = 2

Radio Bearer ID	Downlink RB Rate	Uplink RB Rate
5	42000000	5742000

2013 Jul 18 12:24:42.650 [78] 0x413A WCDMA Radio Bearer Rates

Number of User Plane Radio Bearers = 2

Radio Bearer ID	Downlink RB Rate	Uplink RB Rate
5	16000	16000

2. 检查RRC 状态，RRC在CELL_PCH/CELL_FACH状态下速率会变得很低，或者为0

3. 检查RLC 层是否有错误重传

1980 Jan 6 00:35:28.642 [F2] 0x414A WCDMA RLC DL AM Statistics

Data Logical Channel ID = 19
Control Logical Channel ID = 19
Logical Channel Type = 5 = DTCH
Total Number of PDU Bytes Received = 78745983
Total Number of SDU Bytes to Upper Layer = 78303506
Total Number of Error PDUs Received = 503
Total Number of Data PDUs Received = 27869
Total Number of Control PDUs Received = 1587
Total Number of PDUs NAK'd by PE = 0
Sequence Number of Last PDU in Seq. PDU = 289
Highest Sequence Number PDU Received = 289

检查RLC层 (2)

4. 可以搜索关键字"RLC_ERR: RESET|Post DL reset msg|RESET: max_DAT|RLC RESET"

//UE发Reset给网络侧

01:25:39.696 rlculam.c 3119 H RESET: max_DAT 0x8e (vt_dat, ignore MSB)for retx_sn 1 on LC 19

01:25:39.768 rlcul.c 5283 H Post RLC RESET to RRC, LC 19

//UE收到网络侧的reset

MSG WCDMA RLC/High 17:52:07.850 rlcdlam.c 11165 RLC_DL: Rcvd RESET PDU, reset_sn -- 0, LC Id 28
MSG WCDMA RLC/High 17:52:07.850 rlcdlam.c 11202 RLC_DL: Rcvd RESET with same reset_sn 0 on LC Id 28
MSG WCDMA RLC/High 17:52:08.310 rlcdlam.c 11165 RLC_DL: Rcvd RESET PDU, reset_sn -- 0, LC Id 28
MSG WCDMA RLC/High 17:52:08.310 rlcdlam.c 11202 RLC_DL: Rcvd RESET with same reset_sn 0 on LC Id 28

5. 检查RLC层是否有流量控制(flow control). 目前主要有基于memory, CPU loading 和温度的三种流量控制.搜索关键字" rlcfc|RLC_FC"

1) **Memory-based flow control:**

//Window open

MSG [00000/02] Legacy/High 00:00:57.586 pdcp.c 00487 Sending PDCP open Rx window request

MSG [03004/03] WCDMA RLC/Error 00:00:57.587 rlcul.c 00543 RLC_FC: NA: LC 19, Rcvd 3 (3:OPEN_RX_WIN,1:CLOSE_RX_WIN) cmd frm PDCP

MSG [03004/02] WCDMA RLC/High 00:00:57.587 rlcul.c 06886 Submit New WinSufi, ws 2047, LC 19

//Window close

MSG [00000/02] Legacy/High 00:00:58.099 pdcp.c 00432 Sending PDCP close Rx window request

MSG [03004/03] WCDMA RLC/Error 00:00:58.099 rlcul.c 00543 RLC_FC: NA: LC 19, Rcvd 1 (3:OPEN_RX_WIN,1:CLOSE_RX_WIN) cmd frm PDCP

MSG [03004/02] WCDMA RLC/High 00:00:58.099 rlcul.c 06886 Submit New WinSufi, ws 1, LC 19

检查RLC层 (3)

2) **Temperature-based flow control:**

//Thermal flow control registered

MSG [03004/02] WCDMA RLC/High 15:02:33.087 rlcfc.c 01936 RLC_MULTI_FC::TEMPERATURE DL FC registerd for LC 19

//RLC window changed to 51

MSG [03004/02] WCDMA RLC/High 15:03:22.623 rlcfc.c 02446 TEMP_FC::Timer Callbk 0
MSG [03004/02] WCDMA RLC/High 15:03:22.623 rlcfc.c 02618 TEMP_FC::new window level 1
MSG [03004/02] WCDMA RLC/High 15:03:22.625 rlcfc.c 01772 MULTI_FC::prev output flow 2047 output flow 51
MSG [03004/02] WCDMA RLC/High 15:03:22.625 rlcfc.c 02351 RLC_MULTI_FC:: Action = TxWin(51)

//RLC window size changed to 20

MSG [03004/02] WCDMA RLC/High 15:05:22.622 rlcfc.c 02618 TEMP_FC::new window level 0
MSG [03004/02] WCDMA RLC/High 15:05:22.622 rlcfc.c 01772 MULTI_FC::prev output flow 51 output flow 20
MSG [03004/02] WCDMA RLC/High 15:05:22.622 rlcfc.c 02351 RLC_MULTI_FC:: Action = TxWin(20)

//Thermal flow control de-registered

MSG [03004/02] WCDMA RLC/High 15:06:15.147 rlcfc.c 01097 RLC DL De-registered with FC

3) **CPU loading flow control:**

//CPU loading is 100%

MSG [00043/02] Flow Controller/High 00:02:54.191 fc.c 00499 sleep 0, total 8196 samples -- loading 100

//Down the UL RLC window size

MSG [03004/02] WCDMA RLC/High 00:02:54.312 rlcfc.c 00637 Rxd DN CMD with cmd-id: 0 for 0 direction
MSG [03004/02] WCDMA RLC/High 00:02:54.312 rlcfc.c 00268 Step_dir 1, fc_ws_inc 0, fc_ws_stp 8
MSG [03004/02] WCDMA RLC/High 00:02:54.312 rlcfc.c 00722 FC WS: 20, STEP: 0, Step_WS_dir, 1

//Down the DL RLC window size

MSG [03004/02] WCDMA RLC/High 00:02:54.312 rlcfc.c 00637 Rxd DN CMD with cmd-id: 0 for 1 direction
MSG [03004/02] WCDMA RLC/High 00:02:54.312 rlcfc.c 00268 Step_dir 1, fc_ws_inc 0, fc_ws_stp 8
MSG [03004/02] WCDMA RLC/High 00:02:54.312 rlcfc.c 00722 FC WS: 20, STEP: 0, Step_WS_dir, 1
00:02:54.350 19 01 01 40 04 <-CONTROL PDU:: Type: STATUS SUFI[0]: WINDOW SIZE => 20 SUFI[1]: n/a

4.2 HSDPA物理层下行分析

分析物理层下行(HSDPA)

- 如果RLC层没有发现什么异常,那么就接着分析物理层
- 下行方向，检查HSDPA 的解码情况，可以通过QXDM-→WCDMA→HSDPA Decoding Statistics窗口检查
  - **Average physical layer requested rate –** 根据物理层测到的CQI得出的，能反映当前的无线环境信道质量
  - **Average physical layer scheduled rate –** 网络侧给手机调度块的大小，从而得出网络侧在每个周期的调度速率
  - **Average physical layer served rate –** 网络侧在一段时间里给手机的调度速率
  - **Average MAC layer rate –** 手机成功解调物理层块的速率
  - **SBLER –**每一个子帧的误块率（不包括重传）
  - **Res BLER –** 物理层经过重传的误块率
- 另外在QXDM log里可以检查packet 0x4222(WCDMA HS Decode Status With Data V3)

分析物理层下行(HSDPA)

WCDMA HSDPA Decoding Statistics

	Carrier 0	Carrier 1	Total
HS-SCCH DECODING STATISTICS			
Number of Subframes	8066	0	8066
HS-SCCH Attempts	8065	0	8065
HS-SCCH Successes	83	0	83
HS-SCCH Success Rate	1.03 %	0.00 %	1.03 %
ACK->NACK/DTX (Duplicate SB +) ...	1.32 %	0.00 %	1.32 %
TBS Changes During Retransmission	0	0	0
HS RATE STATISTICS			
Avg Physical Layer Rate (Requested)	1659.50 Kbps	-	1659.50 Kbps
Avg Physical Layer Rate P|S (Sched...	402.20 Kbps	-	402.20 Kbps
Avg Physical Layer Rate P|S (Served)	4.14 Kbps	-	4.14 Kbps
Avg MAC Layer Rate	3.54 Kbps	-	3.54 Kbps
MIMO STATISTICS			
Percentage Single Stream	0.00 %	0.00 %	0.00 %

MIMO	-
64QAM	-
UE Category	-
% Dual Carrier	0.00 %
HS MAC Resets	0
% 1 TB Requested	0.00 %
% 2 TBs Requested	0.00 %

Reset All

HS-DSCH Decoding Statistics Carrier 0

Carrier	TBS (bits)	QPSK P|S	16QAM ...	64QA...	SB- P|S	SB+ P|S	Dup. SB+ P|S	SBLER[1st] P|S (%)	Block-	Bloc...	Res. BLER (%)	1	2	3	4	5	>= 6
CO	Totals	81	2	0	7	75	1	8.54 [8.54]	7	75	8.54	75	0	0	0	0	0
CO	5782	0	1	0	0	1	0	0.00 [0.00]	0	1	0.00	1	0	0	0	0	0
CO	4748	0	1	0	0	1	0	0.00 [0.00]	0	1	0.00	1	0	0	0	0	0
CO	2404	5	0	0	2	3	0	40.00 [40.00]	2	3	40.00	3	0	0	0	0	0
CO	2046	5	0	0	1	4	0	20.00 [20.00]	1	4	20.00	4	0	0	0	0	0
CO	1711	1	0	0	0	1	0	0.00 [0.00]	0	1	0.00	1	0	0	0	0	0
CO	1380	3	0	0	1	2	0	33.33 [33.33]	1	2	33.33	2	0	0	0	0	0
CO	1036	2	0	0	0	2	0	0.00 [0.00]	0	2	0.00	2	0	0	0	0	0
CO	699	7	0	0	0	7	0	0.00 [0.00]	0	7	0.00	7	0	0	0	0	0
CO	365	58	0	0	3	54	1	5.26 [5.26]	3	54	5.26	54	0	0	0	0	0

分析物理层下行(HSDPA)

```
1980 Jan  6  00:33:50.647  [AA]  0x4222  HS Decode Status Log Packet with Data Edition 3
Version =  9
Number of Samples =  25
Start SFN = 850
Mac-ehs enabled = Yes
MIMO CQI reporting = FALSE
Transport Block Size Table = octet aligned
64QAM configured = TRUE
Max number of Harq Processes = 6
Dual Carrier Enabled = false
|--|-----|----|-----|---|---|----|----|-----|----|-----------|---|---|---|---|---|
| #| SCCH|DSCH|HS TB|XRV|New|Num |Code|     |HARQ|SCCH       |   |   |   |Sec|Sec|
|  |A/V/T|Stat| size|   | Tx|Code|Off | Mod |  Id|Type|Ord|Id|DTX|DRX|HSL| HS| UL|
|--|-----|----|-----|---|---|----|----|-----|----|----|---|--|---|---|---|---|---|
| 0|1 1 1|PASS|32264|  6|  1|  14|   1|64QAM|   0|   1|  0| 1|   |   |   |   |   |
| 1|1 1 1|PASS|24232|  6|  1|  15|   1|16QAM|   1|   1|  0| 1|   |   |   |   |   |
| 2|1 1 1|PASS|16352|  6|  1|  13|   1|16QAM|   5|   1|  0| 1|   |   |   |   |   |
| 3|1 1 1|PASS|31128|  6|  1|  15|   1|64QAM|   2|   1|  0| 1|   |   |   |   |   |
| 4|1 1 1|PASS|31128|  6|  1|  15|   1|64QAM|   3|   1|  0| 1|   |   |   |   |   |
| 5|1 1 1|PASS|28976|  6|  1|  15|   1|64QAM|   4|   1|  0| 1|   |   |   |   |   |
| 6|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   0|   1|  0| 1|   |   |   |   |   |
| 7|1 1 1|PASS|31128|  6|  1|  15|   1|64QAM|   1|   1|  0| 1|   |   |   |   |   |
| 8|1 1 1|PASS|26976|  6|  1|  15|   1|64QAM|   5|   1|  0| 1|   |   |   |   |   |
| 9|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   2|   1|  0| 1|   |   |   |   |   |
|10|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   3|   1|  0| 1|   |   |   |   |   |
|11|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   4|   1|  0| 1|   |   |   |   |   |
|12|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   0|   1|  0| 1|   |   |   |   |   |
|13|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   1|   1|  0| 1|   |   |   |   |   |
|14|1 1 1|PASS|31128|  6|  1|  15|   1|64QAM|   5|   1|  0| 1|   |   |   |   |   |
|15|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   2|   1|  0| 1|   |   |   |   |   |
|16|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   3|   1|  0| 1|   |   |   |   |   |
|17|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   4|   1|  0| 1|   |   |   |   |   |
|18|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   0|   1|  0| 1|   |   |   |   |   |
|19|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   1|   1|  0| 1|   |   |   |   |   |
|20|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   5|   1|  0| 1|   |   |   |   |   |
|21|1 1 1|PASS|34040|  6|  1|  15|   1|64QAM|   2|   1|  0| 1|   |   |   |   |   |
|22|1 1 1|PASS|31128|  6|  1|  15|   1|64QAM|   3|   1|  0| 1|   |   |   |   |   |
|23|1 1 1|PASS|31128|  6|  1|  15|   1|64QAM|   4|   1|  0| 1|   |   |   |   |   |
|24|1 1 1|PASS|31128|  6|  1|  15|   1|64QAM|   0|   1|  0| 1|   |   |   |   |   |
```

分析物理层下行(HSDPA)

- HSDPA速率低常见问题
  - RF没有校准
  - Equalizer, SCH-IC, QICE 没有打开。可能是修改NV 3852导致的，这个NV不应该被激活，软件代码里有相应的默认值
  - RF分集没有打开，分集没有校准，或者分集没有接测试线
  - 仪器侧设置了大的固定大小的传输块，从而所有的初次传输都解调不对，从而影响了速率
  - 手机的HSDPA category 和网络不匹配

4.3 HSUPA物理层上行分析

分析物理层上行(HSUPA)

- 上行方向，检查HSUPA 的传输情况，可以通过QXDM-→WCDMA→EUL Link Statistics窗口检查

WCDMA EUL Link Statistics

Throughput	Last Second	Total
Avg Raw TP	4437.16 kbps	4067.85 kbps
Avg Total TP	4369.15 kbps	3991.06 kbps
Avg Scheduled TP	4369.15 kbps	3991.06 kbps
Avg TP Supported by SG	4754.74 kbps	4184.90 kbps
Avg TP Supported by Avail Power	5738.81 kbps	5738.56 kbps
Avg TP Supported by Sched Data	8225.59 kbps	12081.56 kbps
% Limited by SG	69.67 %	58.40 %
% Limited by Power	0.00 %	0.01 %
% Limited by Sched Data	0.00 %	48.64 %

Grants	Last Second	Total
Number of AGs	12	1346
Mean AG	21.92	21.33
% Serving RG Down	0.00 %	0.00 %
% Serving RG Hold	100.00 %	100.00 %
% Serving RG Up	0.00 %	0.00 %
% Non-Serving RG Down	0.00 %	0.00 %
% Non-Serving RG Hold	0.00 %	0.00 %
Avg Serving Grant	28.89 dB	25.15 dB

Buffer Status	Last Second	Total
Avg Scheduled Buffer Status	13864.36 byte(s)	13152.19 byte(s)
Avg Non-Scheduled Buffer Status	0.00 byte(s)	0.00 byte(s)
% Scheduled Buffers Empty	0.00 %	23.77 %
% Non-Scheduled Buffers Empty	92.33 %	92.60 %
% Happy	0.00 %	27.25 %
Number of SIs	205	23783
% of TTIs with SI	68.33 %	55.57 %

UL Transmission	Last Second	Total
% New Transmissions	74.00 %	62.50 %
% Retransmissions	7.67 %	7.47 %
% DTX	18.33 %	30.02 %
BLER	10.20 %	10.85 %
Residual BLER	0.90 %	0.19 %
MAC-e Resets	0	4

UL Power	Last Second	Total
DPCCH	-54.27 dBm	-54.81 dBm
DPCCH %	0.0000 %	0.0000 %
DPDCH T/P	-	-
DPDCH %	0.0000 %	0.0000 %
HS-DPCCH T/P	-0.0070 dB	-0.0070 dB
HS-DPCCH %	0.0000 %	0.0000 %
E-DPCCH T/P	-0.8913 dB	-1.5587 dB
E-DPCCH %	0.0000 %	0.0000 %
E-DPDCH T/P	19.2094 dB	18.1457 dB
E-DPDCH %	0.0001 %	0.0001 %
Power Remaining	58.91dB	60.49dB
Power Remaining %	99.9999 %	99.9999 %

分析物理层上行(HSUPA)

- 另外可以分析log packet 0x4309

1980 Jan 6 00:34:01.909 [10] 0x4309 EUL Combined L1/MAC

Number of Samples = 40

TTI = 2ms

ETFCI Table = 1

Start CFN = 84

Number of Cells = 1

Serv Cell = 0

EDCHcell	RG_ID	TPC_ID	PSC
0 (serv)	0	0	111

SuFN	HQ	Serv RGCH	NS RGCH	Cell_RGCH 111 0			AGCH V	S	I	MAC_D	Reas	SG	LUPR	ReTx CTR	SI	HP	ETFCI	TBS	CM	Comb HICH	Cell_HICH 111 0		
109	5	HLD		HLD						00000010	BO	14	13	0		1	3	354		ACK	ACK		
110	6	HLD		HLD						00000010	BO	14	0	0		1	3	354		ACK	ACK		
111	7	HLD		HLD							BO	14	0	DTX						NAK	DTX		
112	0	HLD		HLD						00000010	BO	14	0	0		1	3	354		ACK	ACK		
113	1	HLD		HLD						00000010	SG	14	0	0		1	3	354		ACK	ACK		
114	2	HLD		HLD								14	13	1		1	3	354		ACK	ACK		
115	3	HLD		HLD						00000010	SG	14	13	0		1	3	354		ACK	ACK		
116	4	HLD		HLD						00000010	SG	14	13	0		1	3	354		ACK	ACK		
117	5	HLD		HLD						00000010	SG	14	13	0		1	3	354		ACK	ACK		
118	6	HLD		HLD						00000010	SG	14	13	0		1	3	354		ACK	ACK		
119	7	HLD		HLD						00000010	SG	14	0	0		1	3	354		ACK	ACK		
120	0	HLD		HLD						00000010	SG	14	13	0		1	3	354		ACK	ACK		
121	1	HLD		HLD						00000010	SG	14	13	0		1	3	354		ACK	ACK		
122	2	HLD		HLD						00000010	SG	14	13	0		1	3	354		ACK	ACK		
123	3	HLD		HLD						00000010	SG	14	13	0		1	3	354		NAK	DTX		

分析物理层上行(HSUPA)

- HSUPA速率低常见问题
  - 网络分配SG太低

SuFN	HQ	Serv RGCH	NS RGCH	Cell_RGCH 111 0				AGCH V	S	I	MAC_D	Reas	SG
198	6	HLD		HLD							00000010	SG	27
199	7	HLD		HLD							00000010	SG	27
200	0	HLD		HLD							00000010	SG	27
201	1	HLD		HLD							00000010	SG	27

- 可用于HSUPA传输的功率受限，需要检查手机发送校准和手机的最大发送功率

SuFN	HQ	Serv RGCH	NS RGCH	Cell_RGCH 309 0				AGCH V	S	I	MAC_D	Reas
1019	3	HLD		HLD							00000010	MP
1020	4	HLD		HLD							00000010	MP
1021	5	HLD		HLD							00000010	MP
1022	6	HLD		HLD							00000010	MP

分析物理层上行(HSUPA)

- 上层给物理层的数据太少，需要检查DATA层是否有流控，或者丢包

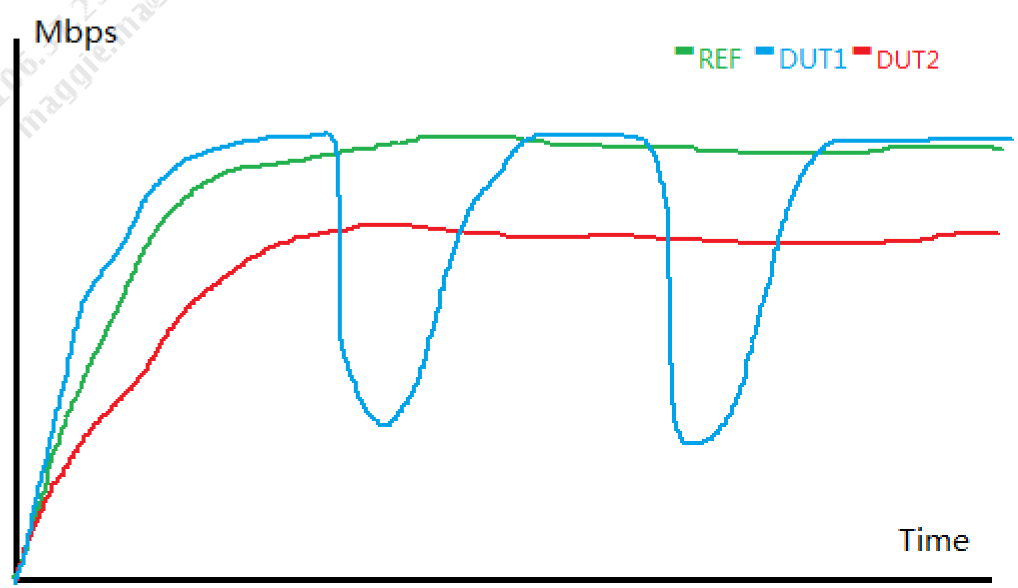
SuFN	HQ	Serv RGCH	NS RGCH	Cell_RGCH 111 0				AGCH V	S	I	MAC_D	Reas
148	4	HLD		HLD								BO
149	5	HLD		HLD								BO
150	6	HLD		HLD								BO
151	7	HLD		HLD								BO

- 传输误码率太高，需要检查RF 发送是否准确，另外要和参考机做对比测试

5. TD-SCDMA/HSPA 层分析

TD-SCDMA/HSPA 数据吞吐量问题分析

- TD-SCDMA/HSPA 数据的吞吐量测试依赖于网络环境，所以在测试前一定要清楚自己的目标速率。相同的测试机器在不同的测试环境下，速率表现会相差很远。因此分析问题的时候通常需要和参考手机做对比。
- 通常会有两类速率问题
  - 测试机(DUT1)在整个测试过程中都比对比机(REF)差
  - 测试机(DUT2)大多数时候都和对比机(REF)差不多，有时候会差一点
- DUT2处理比较简单，找到速率变低的时间点然后从TCP -→RLC-→Physical 检查原因
- 这个文档主要通过分析DUT1的现象来了解怎么样从TCP到物理层来分析吞吐量问题



5.1 RLC Layer 分析

分析RLC层(1)

- 如果测试机比参考机UDP测试差，那么需要进一步分析TDSCDMA RLC 层

1.检查RB 速率是否满足要求.

1980 Jan 8 09:09:11.389 [00] 0xD0E4 TDSCDMA RRC RB Rate Info

```
Version = 1
Version 1 {
  num_rbs = 1
  Rb Rate {
    rb_id = 5
    dl_rb_rate = 2800000//2.8Mbps 表示网络配置为下行为HSDPA.
    ul_rb_rate = 16000//UL is 16kbps
  }
}
```

2.检查RRC 状态，RRC在CELL_PCH/CELL_FACH状态下速率会变得很低，或者为0.

3.检查RLC 层是否有错误重传.

```
Rlc Dl Am Stat[3] {
   log_data_dl_rlc_id = 27
   log_ctl_dl_rlc_id = 27
   log_chan_type = UE_LOGCHAN_DTCH
   tot_num_pdu_byte_rxd = 1148902
   tot_num_sdu_byte_rxd = 1095428
   tot_num_error_pdu_rxd = 35
   tot_num_data_pdu_rxd = 13744
   tot_num_ctl_pdu_rxd = 232
   tot_num_pdu_nak = 0
   sn_last_inseq_pdu_rxd = 1456
   highest_sn_pdu_rxd = 1456
 }
```


分析RLC层(2)

4.可以搜索关键字”RLC_ERR: RESET|Post DL reset msg|RESET: max_DAT|RLC RESET”.

//UE发Reset给网络侧

MSG [10254/02] TDSCDMA RLC/High 02:01:22.970 tdsrlcul.c 04360 Post RLC RESET to RRC, LC 24

MSG [10255/02] TDSCDMA RRC/High 02:01:22.971 tdsrrccu.c 22719 RLC RESET type:0,LC_id:24,dir:1

//UE收到网络侧的reset

MSG [10254/02] TDSCDMA RLC/High 02:17:11.350 tdsrlcdlam.c 05831 RLC_DL: Rcvd RESET PDU, reset_sn -- 0, LC Id 25

MSG [10254/02] TDSCDMA RLC/High 02:17:11.350 tdsrlcdlam.c 01725 Post UL RESET Msg, LC 25

5.2 HSDPA物理层下行分析

分析物理层下行(HSDPA)(1 of 5)

- 如果RLC层没有发现什么异常,那么就接着分析物理层.
- 首先在APEX里检查HSPDA Summary

CQI Summary		
	Average RTBS	46.64
	% of 16QAM	98
	% of QPSK	1
	No. of times seEff = 0 / Total Sample, (%)	0 / 3727, (0)
	No. of times RTBS = 0 / Total Sample, (%)	0 / 3727, (0)
	Average Requested code rate	0.86
Average Target SIR	2.60	
Average HS-SCCH SIR	17.80	
Average SIR	9.46	
Average RSCP	-87.98	
Average Tx Power		
	Slot	Tx Power
Percentage of Scheduling: (HS-SCCH Successes / Attempts, (Success Rate in %))	4612 / 14336, (32.1708)	
Percentage of allocated modulation scheme		
	QPSK	21.39
	16 QAM	78.61
Average TBS index	42.9612	
HS-SCCH BLER HCSN out of sequence	1	
Percentage of Ts Hs Pdsch		
	Ts Hs Pdsch	Percentage
	4	0.24
	6	0.38
	8	22.15
	12	11.92
	14	65.31

4:HS-PDSCH TS is 4
6:HS-PDSCH TS is 4,5
8:HS-PDSCH TS is 3
12:HS-PDSCH TS is 3,4
14:HS-PDSCH TS is 3,4,5

分析物理层下行(HSDPA)(2 of 5)

- 在APEX->TDSCDMA->L1->0xD032 TDSCDMA L1 HSDPA HSSCCH Statistics 检查 TDS HSSCCH 调度

1980 Jan 8 09:09:18.025 [00] 0xD032 TDSCDMA L1 HSDPA HSSCCH Statistics

```
Version = 1
Version 1 {
  cell_id = 0
  start_sys_frame_no = 2902
  reserved = 0
  num_sub_frames = 512
  num_scch_valid = 512//表示网侧在512个子帧内调度的SCCH个数
}
```

- 如果num_scch_valid 非常小, 表示网络调度HSSCCH不足.

分析物理层下行(HSDPA)(3 of 5)

- 在APEX->TDSCDMA->L1->0xD031 TDSCDMA L1 HSDPA Decode Status检查HCSN

1980 Jan 8 09:09:11.575 [00] 0xD031 TDSCDMA L1 HSDPA Decode Status

```
Version = 1
Version 1 {
  cell_id = 0
  reserved = 0
  num_samples = 10
  info = 976
  Decode Status[0] {
    phyChType = 0
    Scch {
      arriving_sys_frame_no = 2248
      arriving_sub_frame_no = 4497
      crc_pass = 1
      hcsn = 0
```

Time	hcsn
09:09:12.285	0
09:09:12.285	1
09:09:12.285	2
09:09:12.285	3
09:09:12.490	4
09:09:12.490	5
09:09:12.490	6
09:09:12.490	7
09:09:12.490	0
09:09:12.725	1
09:09:12.725	2
09:09:12.725	3
09:09:12.725	4
09:09:12.725	5
09:09:12.905	6
09:09:12.905	7
09:09:12.905	0
09:09:12.905	1
09:09:12.905	2
09:09:13.210	3
09:09:13.210	4
09:09:13.210	5
09:09:13.210	0

miss 6 and 7, SCCH not continuous

- HCSN 应该是0到7的循环, 否则UE接收的SCCH是不连续的.

分析物理层下行(HSDPA)(4 of 5)

- 在 APEX->TDSCDMA->L1->0xD033 TDSCDMA L1 HSDPA HARQ Statistics 检查BLER

1980 Jan 8 09:09:25.705 [00] 0xD033 TDSCDMA L1 HSDPA HARQ Statistics

Version = 3
Version 3 {
  cell_id = 0
  reserved = 0
  num_sub_frames = 512
  start_sys_frame_no = 3670
  num_harq_proc = 5
  bler_overall_newtx = 6
  bitrate_avg_kbps = 578
  bler_2secs_newtx = 0
  bitrate_2secs_kbps = 149

Time	bler_2secs_newtx
09:08:05.975	0
09:09:12.885	8
09:09:15.465	1
09:09:18.025	8
09:09:20.585	8
09:09:23.145	11
09:09:25.705	0
09:09:28.265	0
09:09:30.825	0
09:09:33.385	10
09:09:35.945	1
09:09:38.505	1
09:09:41.065	1
09:09:43.625	1
09:09:46.185	3
09:09:48.745	4
09:09:51.305	3
09:09:53.865	1
09:09:56.425	0

- 如果 bler_2secs_newtx 非常高, 可能和RF接收相关.

分析物理层下行(HSDPA)(5 of 5)

- Common causes of a low HSDPA rate:
  - RF 没有校准.
  - 如果有功率或者信号的问题, 有可能是RF是射频天线开关时间被改变了.
  - 仪器侧设置了大的固定大小的传输块，从而所有的初次传输都解调不对，从而影响了速率
  - 手机的HSDPA category 和网络不匹配

5.3 HSUPA 物理层分析

分析物理层上行(HSUPA)

- 在APEX->TDSCDMA->0xD044 TDSCDMA L1 UPA ETFC Restriction Info检查ETFC.

1980 Jan 6 02:42:10.075 [0/0x00] 0xD044 TDSCDMA L1 UPA ETFC Restriction Info

```
Version = 2
Version2 {
   num_samples = 40
   first_sample_sub_fn = 1169
   max_allowed_tx_pwr = 24
   Etfc Restrict Info
```

#	subfn_offset	sg_type	snpl	max_epuch_pwr_avail	other_ch_type	prri	uph	pebase	dtx_flag	max_sup_etfci	max_sg_etfci	etfci	harq_id
0	0	0	31	26	0	20	30	-126	0	59	59	59	0
1	1	0	31	26	0	20	30	-126	0	59	59	59	2
2	2	0	31	26	0	20	30	-125	0	59	59	59	1
3	3	0	31	26	0	20	30	-125	0	59	59	59	3
4	4	0	31	26	0	20	30	-123	0	59	59	59	0
5	5	0	31	26	0	20	30	-123	0	59	59	59	2
6	6	0	31	26	0	20	30	-121	0	59	59	59	1
7	7	0	31	26	0	20	30	-121	0	59	59	59	3
8	8	0	31	26	0	20	30	-121	0	59	59	59	0
9	9	0	31	26	0	20	30	-121	0	59	59	59	2
10	10	0	31	26	0	20	30	-121	0	59	59	59	1
11	11	0	31	26	0	20	30	-121	0	59	59	59	3
12	12	0	31	26	0	20	30	-121	0	59	59	59	0
13	13	0	31	26	0	20	30	-121	0	59	59	59	2
14	14	0	31	26	0	20	30	-121	0	59	59	59	1

如果etfci过小，可能是上次发送的数据不足，网侧分配的授权太小和UE的发送功率相关

分析物理层上行(HSUPA)

- 在0xD046 TDSCDMA L1 UPA EDL Dec And SG Info检查授权信息.

1980 Jan 6 02:42:10.075 [0/0x00] 0xD046 TDSCDMA L1 UPA EDL Dec And SG Info

```
Version = 1
Version 1 {
   num_samples = 40
   first_sample_sub_fn = 1168
   Sample Info
      ----------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------
      |#
|sg_type|ag_flag|prri|crri|trri|eni|rdi|hich_rvd_flag|hich_ack_nack|hich_harq_id|select_harq_id|new_tx|etfci|curr_tx_nb|max_tx_nb|rtx_tmr|rt
x_tmr_exp|erucch_flag|
      ----------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------
      |  0|       1|       1|  20|   0|   8|  1|  0|            1|            1|           0|             0|     1|   59|         0|        5|
0|       72|           0|
      |  1|       1|       1|  20|   0|   8|  1|  0|            1|            1|           2|             2|     1|   59|         0|        5|
0|       72|           0|
      |  2|       1|       1|  20|   0|   8|  1|  0|            1|            1|           1|             1|     1|   59|         0|        5|
0|       72|           0|
      |  3|       1|       1|  20|   0|   8|  1|  0|            1|            1|           3|             3|     1|   59|         0|        5|
0|       72|           0|
      |  4|       1|       1|  20|   0|   8|  1|  0|            1|            0|           0|             0|     0|   59|         1|        5|
0|       72|           0|
      |  5|       1|       1|  20|   0|   8|  1|  0|            1|            0|           2|             2|     0|   59|         1|        5|
0|       72|           0|
      |  6|       1|       1|  20|   0|   8|  1|  0|            1|            1|           1|             1|     1|   59|         0|        5|
0|       72|           0|
      |  7|       1|       1|  20|   0|   8|  1|  0|            1|            0|           3|             3|     0|   59|         1|        5|
0|       72|           0|
```

- 如果PRRI过小, 表示网侧分配的绝对授权很低.

分析物理层上行(HSUPA)

- 在0xD048 TDSCDMA L1 UPA Statistics Info检查BLER.

0xD048 TDSCDMA L1 UPA Statistics Info

```
Version = 3
Version 3 {
   curr_pathloss = 107
   snpl = 31
   schld_stats_included = 1
   non_schld_stats_included = 0
   Schld Stats Info {
      start_sub_fn = 1008
      num_sub_frames = 200
      num_sg_avail_sub_frames = 196
      num_new_tx = 184
      num_re_tx = 12
      num_ack = 184
      num_nack = 12
      num_nack_new_tx = 12
      num_failed_harq = 0
      total_bler = 6.52 %
      new_tx_bler = 6.52 %
      residual_bler = 0.00 %
      upa_l1_tput = 415840
      sum_raw_bits = 442960
      sum_pwr_bits = 442960
      sum_sg_bits = 442960
      sum_s_buf = 0
      sum_etfci = 10856
      num_agch = 196
      num_spacing_tti = 0
      sum_prri = 1568
      pebase = -123
      num_tpc_up = 22
      num_tpc_down = 174
      num_dtx = 0
      sum_num_vu = 3136
      num_rtx_timeout = 0
      num_tx_num_exceed = 0
      num_first_tx_acked = 172
      num_pwr_limited = 0
      num_sg_limited = 184
      num_si_epuch = 0
      sum_epuch_pwr = 510
      num_success_erucch = 0
      num_failed_erucch = 0
      num_abort_erucch = 0
      num_ehich_lost = 0
      num_eagch_lost = 0
      sum_num_ts = 196
      num_sg_maller_than_uph = 0
      num_harq_fail_per_id = { 0, 0, 0, 0 }
```

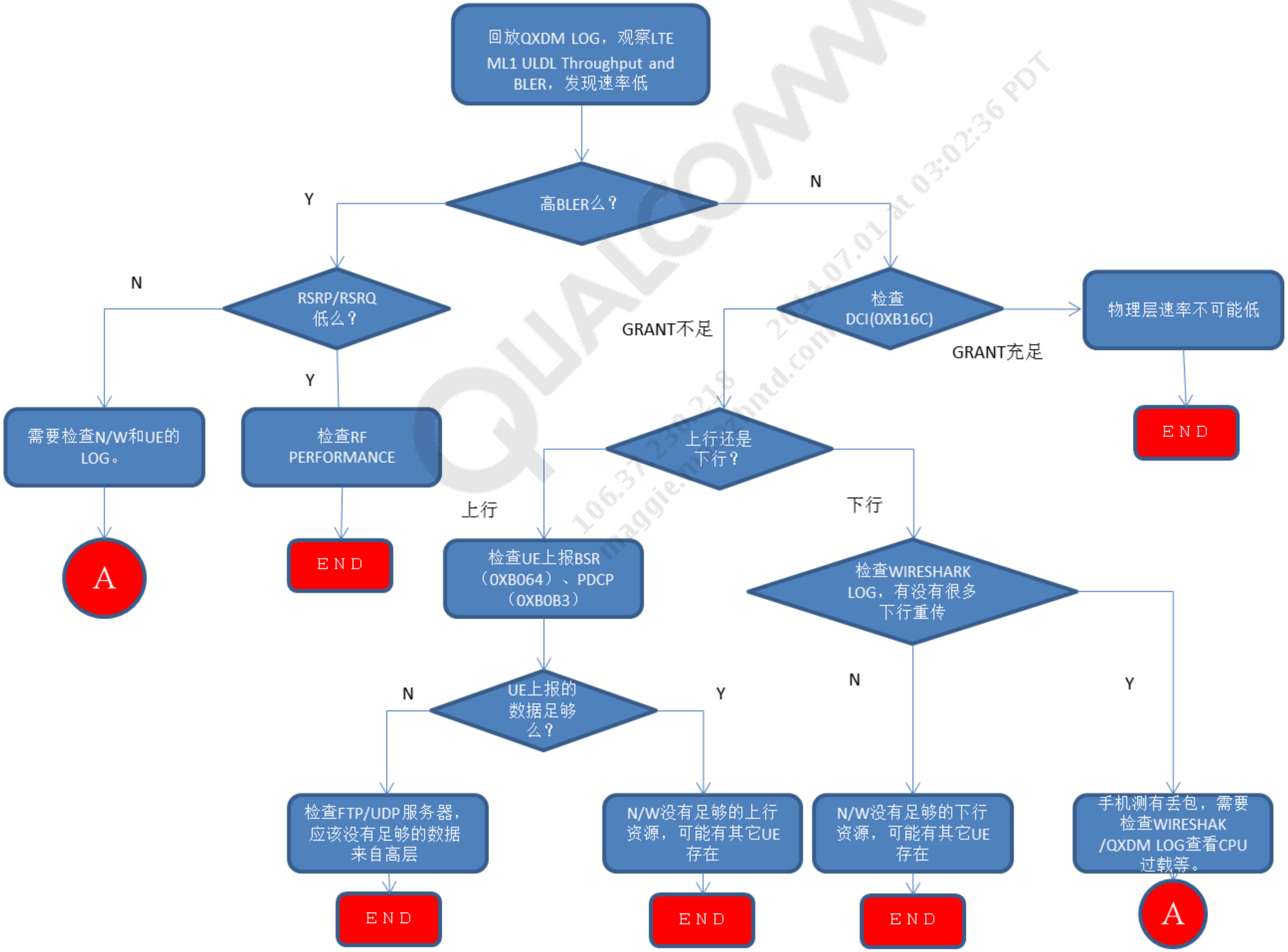
如果bler太高，可能和RF发送相关.

分析物理层上行(HSUPA)

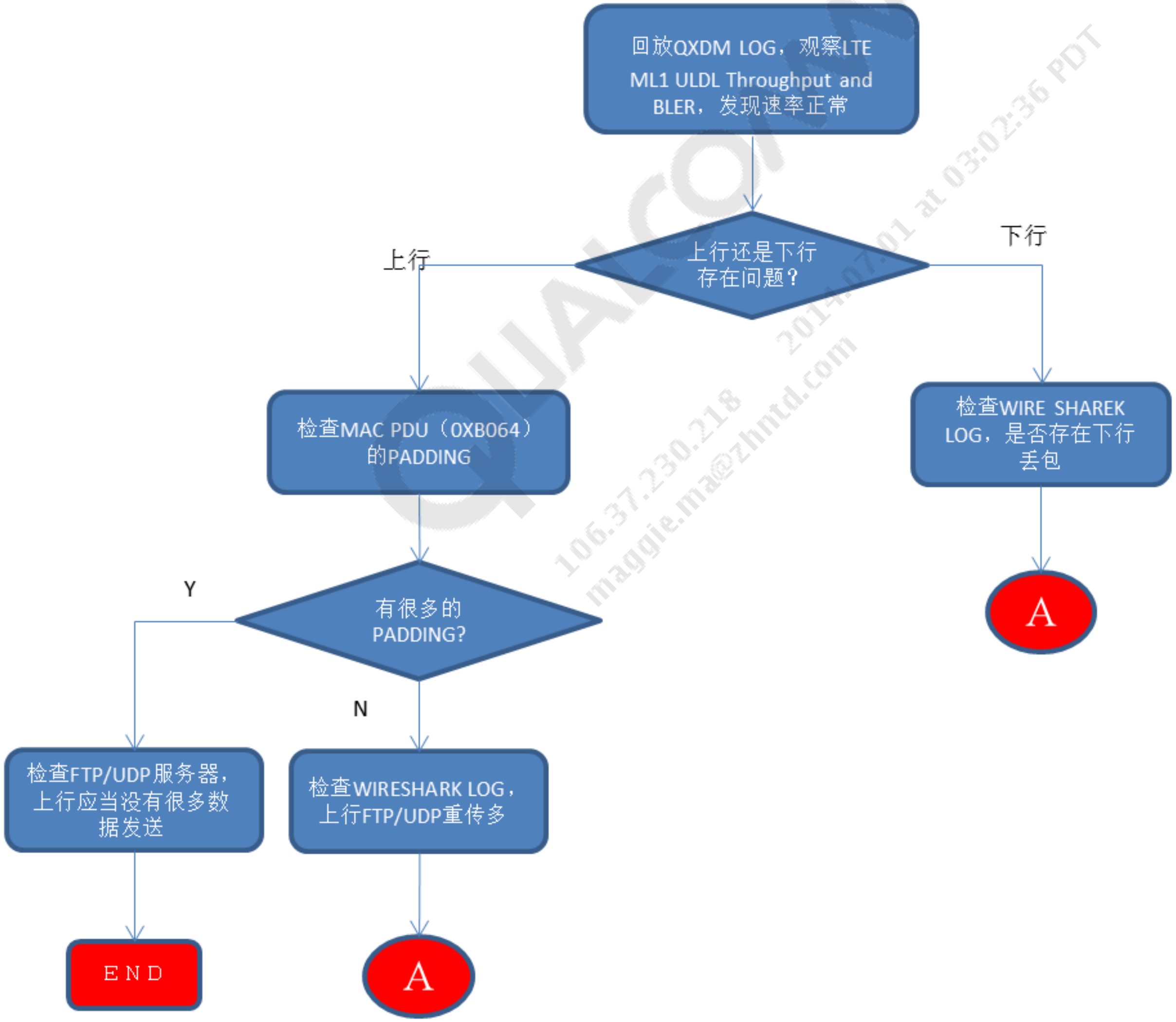
- HSUPA速率低常见问题
  - 网络分配AG太低
- 可用于HSUPA传输的功率受限，需要检查手机发送校准和手机的最大发送功率.
- 上层给物理层的数据太少，需要检查DATA层是否有流控，或者丢包.
- 传输误码率太高，需要检查RF 发送是否准确，另外要和参考机做对比测试.

6. LTE层分析

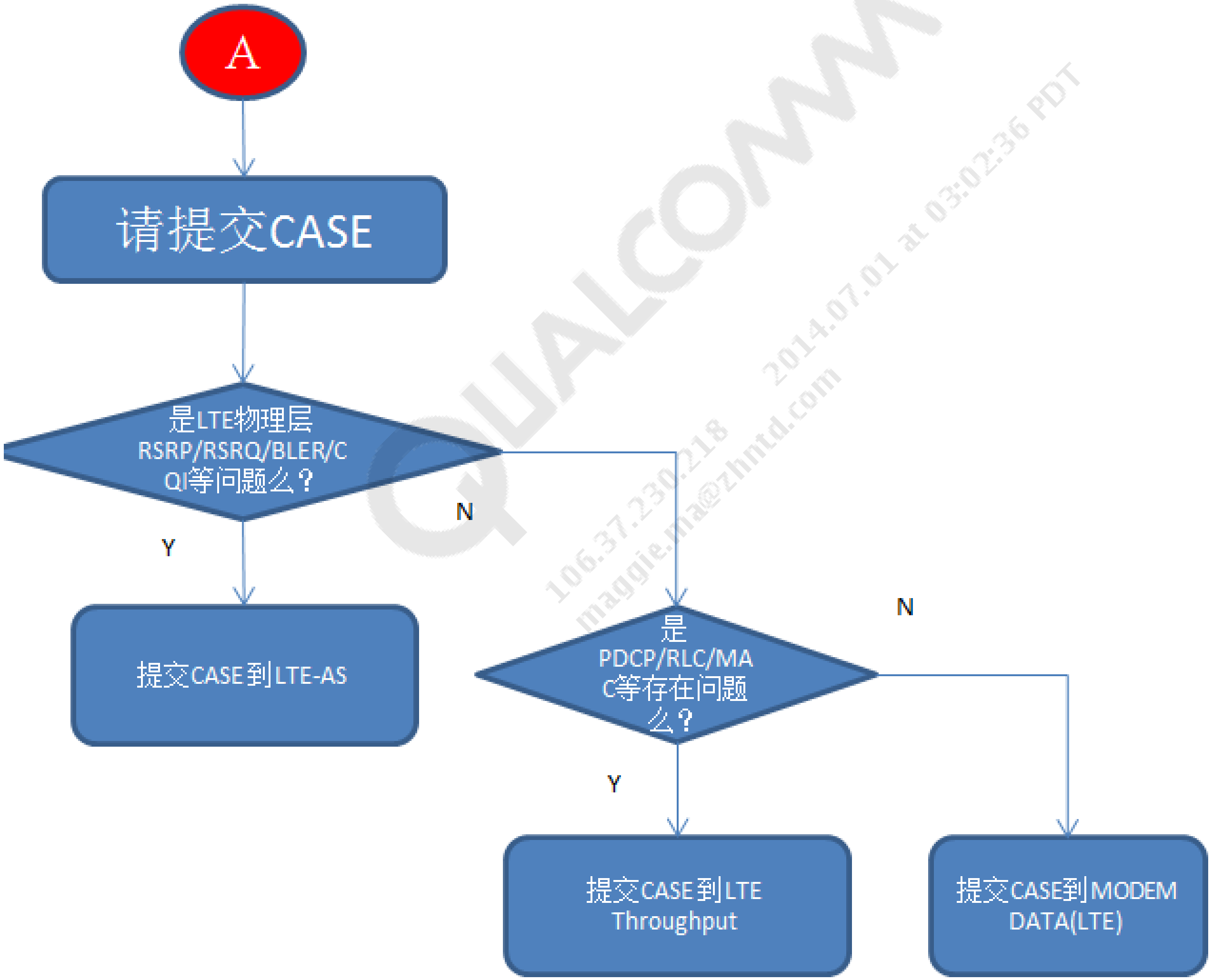
LTE物理层速率低检查流程



LTE物理层速率正常检查流程



提交CASE


A
请提交CASE
是LTE物理层RSRP/RSRQ/BLER/CQI等问题么？
Y
N
提交CASE到LTE-AS
是PDCP/RLC/MAC等存在问题么？
Y
N
提交CASE到LTE Throughput
提交CASE到MODEM DATA(LTE)

7. RFSW分析

RFSW分析内容

- 适用平台
- 射频时序
- 错误举例
- 参数调整原则

适用平台

- 使用GPIO做为控制PA和ASM的平台，包括
  8960,8930, 9x15, Fusion3, 8974, 8974AB, 9x25, 8x26,8926,8x10,8x12..

接收时序

以9x15代码为例: QCT 默认RX ASM天线时序是(-5,0), RX_ON0 时序是 (-150,0)，这些时序的单位是us, -5代表在一帧的帧头提前5us打开天线接收通路。

```
rfc_signal_gsm_cfg_type rf_card_wtr1605_chn3_rx0_tdscdma_b34_sig_cfg =
{
  {
    WTR1605, /* rf_asic_name */
    0, /* rf_asic_id */
    ( RFDEVICE_PA_LUT_MAPPING_INVALID ), /* rf_asic_band_pa_lut_map */
    (int)WTR1605_TDSCDMA_PRX_BAND34_PMB1, /* rf_asic_sw_port_map */
    FALSE, /* txlut, config */
  },
  0x01230111,   /* Revision: V1.35.273 */
  {
    {  { (int)RFC_WTR1605_CHN3_RX_ON0_DEFAULT, RFC_HIGH}, {-150,0}  },
    {  { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_10_DEFAULT, RFC_HIGH}, {-5,0}  },
    {  { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_11_DEFAULT, RFC_LOW}, {-5,0}  },
    {  { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_12_DEFAULT, RFC_HIGH}, {-5,0}  },
    {    { (int)RFC_SIG_LIST_END, RFC_LOW  },  {0, 0}} ,
  },
};
```

发射时序

- 以9x15代码为例: QCT 默认TX ASM发射时序是(-5,-6)，PA_ON 时序是 (-5,-4), TX ON 时序是(-25,12)

```
rfc_signal_gsm_cfg_type rf_card_wtr1605_chn3_tx0_tdscdma_b34_sig_cfg =
{
  {
    WTR1605, /* rf_asic_name */
    0, /* rf_asic_id */
    ( RFDEVICE_PA_LUT_MAPPING_INVALID ), /* rf_asic_band_pa_lut_map */
    (int)WTR1605_TDSCDMA_TXWSAW_BAND34_TMB1, /* rf_asic_sw_port_map */
    TRUE, /* txlut, config */
  },
  0x01230111,    /* Revision: V1.35.273 */
  {
    {   { (int)RFC_WTR1605_CHN3_PA_ON_2_TX0_DEFAULT, RFC_HIGH}, {-5,-4}  },
    {   { (int)RFC_WTR1605_CHN3_TX_ON0_DEFAULT, RFC_CONFIG_ONLY}, {-25,12}  },
    {   { (int)RFC_WTR1605_CHN3_PA0_R0_DEFAULT, RFC_CONFIG_ONLY}, {-5,-4}  },
    {   { (int)RFC_WTR1605_CHN3_PA0_R1_DEFAULT, RFC_CONFIG_ONLY}, {-5,-4}  },
    {   { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_10_DEFAULT, RFC_HIGH}, {-5,-6}  },
    {   { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_11_DEFAULT, RFC_HIGH}, {-5,-6}  },
    {   { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_12_DEFAULT, RFC_HIGH}, {-5,-6}  },
    {   { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_13_DEFAULT, RFC_LOW}, {-5,-6}  },
    {   { (int)RFC_WTR1605_CHN3_PA0_SMPS_PDM_DEFAULT, RFC_CONFIG_ONLY}, {-54,-4}  },
    {   { (int)RFC_WTR1605_CHN3_GNSS_BLANK_TDS, RFC_HIGH}, {-6,-3}  },
    {   { (int)RFC_WTR1605_CHN3_INTERNAL_03_DEFAULT, RFC_CONFIG_ONLY}, {-11,-3}  },
    {     { (int)RFC_SIG_LIST_END, RFC_LOW  },  {0, 0}} ,
  },
```

错误举例 (1)

1. 下面例子为了通过射频硬件发射杂散测试，错误的将发射ASM时序从 (-5,-6) 改成 (-30,0)

   这种错误的改法会造成TD-HSDPA吞吐量显著下降

```
rfc_signal_gsm_cfg_type rf_card_wtr1605_chn3_tx0_tdscdma_b34_sig_cfg =
{
  {
    WTR1605, /* rf_asic_name */
    0, /* rf_asic_id */
    ( RFDEVICE_PA_LUT_MAPPING_INVALID ), /* rf_asic_band_pa_lut_map */
    (int)WTR1605_TDSCDMA_TXWSAW_BAND34_TMB1, /* rf_asic_sw_port_map */
    TRUE, /* txlut, config */
  },
  0x01230111,   /* Revision: V1.35.273 */
  {
    {  { (int)RFC_WTR1605_CHN3_PA_ON_2_TX0_DEFAULT, RFC_HIGH}, {-5,-4}  },
    {  { (int)RFC_WTR1605_CHN3_TX_ON0_DEFAULT, RFC_CONFIG_ONLY}, {-25,12}  },
    {  { (int)RFC_WTR1605_CHN3_PA0_R0_DEFAULT, RFC_CONFIG_ONLY}, {-5,-4}  },
    {  { (int)RFC_WTR1605_CHN3_PA0_R1_DEFAULT, RFC_CONFIG_ONLY}, {-5,-4}  },
    {  { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_10_DEFAULT, RFC_HIGH}, {-30,0}  },
    {  { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_11_DEFAULT, RFC_HIGH}, {-30,0}  },
    {  { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_12_DEFAULT, RFC_HIGH}, {-30,0}  },
    {  { (int)RFC_WTR1605_CHN3_RF_PATH_SEL_13_DEFAULT, RFC_LOW}, {-30,0}  },
    {  { (int)RFC_WTR1605_CHN3_PA0_SMPS_PDM_DEFAULT, RFC_CONFIG_ONLY}, {-54,-4}  },
    {  { (int)RFC_WTR1605_CHN3_GNSS_BLANK_TDS, RFC_HIGH}, {-6,-3}  },
    {  { (int)RFC_WTR1605_CHN3_INTERNAL_03_DEFAULT, RFC_CONFIG_ONLY}, {-11,-3}  },
    {    { (int)RFC_SIG_LIST_END, RFC_LOW  },  {0, 0}} ,
  },
};
```

错误举例 (2)

- 在DIME平台下，高通默认的ASM是MIPI的，所以如果客户使用GRFC的天线开关，不能直接使用高通默认的时序

For RX:

Error code

```
rfc_sig_tdd_info_type rf_card_wtr1605_sglte_cmcc_4mode_rx0_tdscdma_b34_sig_cfg =
{
 0x02220204,  /* Revision: v2.34.516 */
 {
  //{ (int)RFC_WTR1605_SGLTE_CMCC_4MODE_TIMING_ASM_CTL,  { RFC_CONFIG_ONLY /*Warning: Not specified*/ , -5 }, {RFC_LOW, -6 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RX_ON0,  { RFC_HIGH, -150 }, {RFC_LOW, 0 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_20,  { RFC_LOW, -25 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_01,  { RFC_HIGH, -25 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_02,  { RFC_LOW, -25 }, {RFC_LOW, -4 } },
  { (int)RFC_SIG_LIST_END,  { RFC_LOW, 0 }, {RFC_LOW, 0 } }
 },
};
```

Right code

```
rfc_sig_tdd_info_type rf_card_wtr1605_sglte_cmcc_4mode_rx0_tdscdma_b34_sig_cfg =
{
 0x02220204,  /* Revision: v2.34.516 */
 {
  //{ (int)RFC_WTR1605_SGLTE_CMCC_4MODE_TIMING_ASM_CTL,  { RFC_CONFIG_ONLY /*Warning: Not specified*/ , -5 }, {RFC_LOW, -6 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RX_ON0,  { RFC_HIGH, -150 }, {RFC_LOW, 0 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_20,  { RFC_LOW, -5 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_01,  { RFC_HIGH, -5 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_02,  { RFC_LOW, -5 }, {RFC_LOW, -4 } },
  { (int)RFC_SIG_LIST_END,  { RFC_LOW, 0 }, {RFC_LOW, 0 } }
 },
};
```

错误举例 (3)

- 发射也是一样的道理

For TX：

```
Wrong code:
rfc_sig_tdd_info_type rf_card_wtr1605_sglte_cmcc_4mode_tx0_tdscdma_b34_sig_cfg =
{
 0x02220204,  /* Revision: v2.34.516 */
 {...
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_PA0_R0,  { RFC_CONFIG_ONLY, -7 }, {RFC_LOW, -4 } },//PA range
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_PA_ON_16,  { RFC_HIGH, 0 }, {RFC_LOW, 0 } },//PA enable
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_07,  { RFC_HIGH, -25 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_11,  { RFC_LOW, -25 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_20,  { RFC_HIGH, -25 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_01,  { RFC_LOW, -25 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_02,  { RFC_HIGH, -25 }, {RFC_LOW, -4 } },
  //{ (int)RFC_WTR1605_SGLTE_CMCC_4MODE_INTERNAL_GNSS_BLANK_CONCURRENCY,  { RFC_HIGH, -5 }, {RFC_LOW, -6 } },
  //{ (int)RFC_WTR1605_SGLTE_CMCC_4MODE_TX_GTR_TH,  { RFC_CONFIG_ONLY, -5 }, {RFC_LOW, -6 } },
  //{ (int)RFC_WTR1605_SGLTE_CMCC_4MODE_PA_IND,  { RFC_HIGH, -5 }, {RFC_LOW, -6 } },
  { (int)RFC_SIG_LIST_END,  { RFC_LOW, 0 }, {RFC_LOW, 0 } }
 },
};


Right code:
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_07,  { RFC_HIGH, -5 }, {RFC_LOW, -4 } },
 { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_11,  { RFC_LOW, -5 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_20,  { RFC_HIGH, -5 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_01,  { RFC_LOW, -5 }, {RFC_LOW, -4 } },
  { (int)RFC_WTR1605_SGLTE_CMCC_4MODE_RF_PATH_SEL_02,  { RFC_HIGH, -5 }, {RFC_LOW, -4 } },
```


时序调整原则

- 原则上说，上述时序不需要调整，因为QCT使用这些时序可以通过所有的射频硬件和软件测试
- 如果因为客户选择器件的特性必须要进行调整，那么尽可能微调，最好限制在1-2个单位内，而且调整后必须要做下面的回归测试，确保调整前和调整后没有影响。
- 测试一：使用8820C做TD-HSDPA 吞吐量测试
- 测试二：发射时隙2的EVM，接收时隙3,4,5的灵敏度
  以上两个测试的配置为:时隙2配置为上行，时隙3，4，5配置为下行，做HSDPA

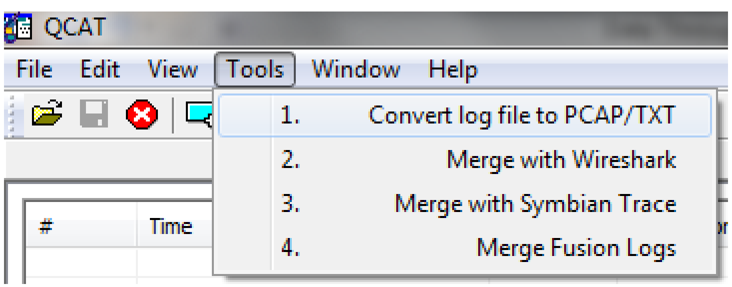
附录A：如何抓取log

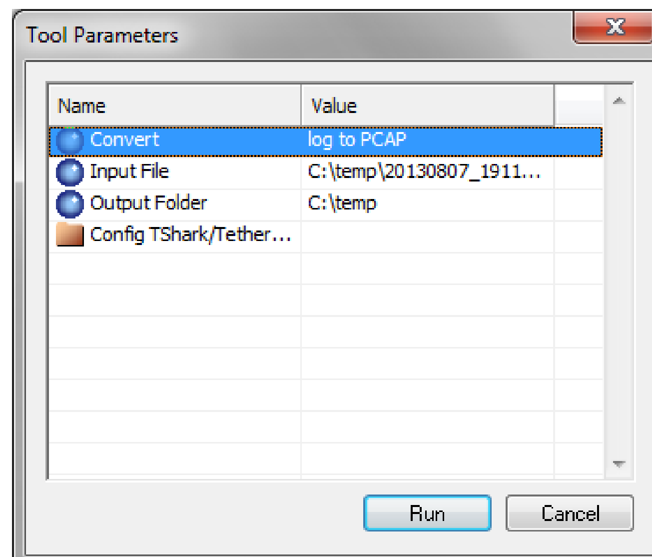
TCP/IP/UDP/ICMP等报文的抓取

- PC侧的pcap log是通过Wireshark工具获得，请在PC侧开启Wireshark，监测USB(虚拟以太网卡/PPP)口的流量

- 如果是Android上测试，则在adb shell上运行tcpdump 抓取rmnet/wlan设备的pcap;
  - 命令：adb shell tcpdump -i any -s 0 -w /data/tcpdump.pcap

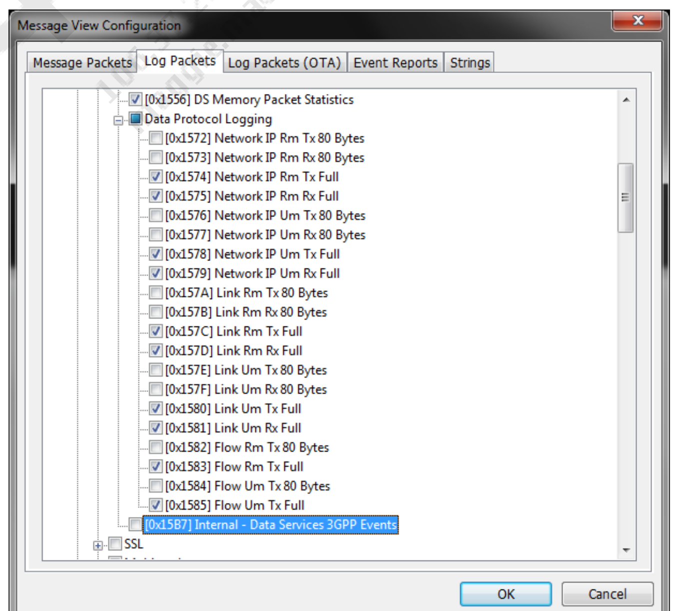
- UE侧的pcap log是通过QXDM Log转换的，但是必须开启"Data Protocol Logging(DPL)"的LOG PACEKT.
  - 转换工具：QCAT/Tools/Convert log file…
  - 如下图：





如何抓取DPL log

- DPL Logging
  - QXDM Log mask路径: Message View/Known Log Items/Common/Data Services/Data Protocol Logging
  - DPL full packet logging: 抓取数据包的全部字节(例如：1500 字节)
    - 在需要查看全部报文内容时使用，例如需要跟踪完整的HTTP数据流。
    - 在数据流量大时，由于抓取的Log很大，可能对吞吐量有影响
  - DPL packet headers logging: 抓取数据包的头部(例如：80字节).
    - 一般而言，TCP/IP报头为40字节，抓取报头前面的80字节已经够用了。此方法对高吞吐量测试影响相对小些。



附录B：ping包测试

使用ping包测试

- Ping包是一种非常简单而又实用的测试
  - 由Ping Request/Ping Reply组成；
  - 无论是否收到reply, 发送端都可以持续不断发送；
  - 可以测试客户端到服务器的环回时延(RTT)
  - 在PC上、或ADB shell上均可以测试

- 在数据出现断流(Data Stall)时，保持ping包，可以检查此时链路是否通畅
  - 典型案例：
    - **现象：**使用Android浏览器，偶然出现某些页面打不来，例如：http://m.baidu.com
    - **测试：**在打不开网页时，用ping包测试，所有ping包都收到Reply，证明链路是通的。用Wireshark分析打开网页的数据流，发现是网址跳转后，浏览器不再发后续的数据包。更换第三方浏览器可正常打开该网页；
    - **结论：**浏览器处理网页URL跳转有问题。Ping包很好地排除了底层链路问题

- 在数据吞吐量不达标时，在下载的同时保持ping包测试
  - 数据量大时，分析TCP数据流比较耗时。而ping包则相对简单；
  - 可以对比此时ping包的时延是否偏大、是否有丢包。
  - 把ping包与PDCP/RLC/MAC各层逐一对包，比较容易查看各层时延、是否丢包。如果数据流量大时检查TCP包，相当困难

附录C：TCP协议设置/关键算法分析

检查TCP参数设置

- TCP 的窗口大小必须大于测试设备的传输时延乘积(Bandwidth Delay Product, BDP)
  - TCP_WinSIZE>=Bandwidth X RTT(BDP)
  - WinXP – HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters->TcpWindowSize
  - Linux – TCP window size can be set in /etc/sysctl.conf by adding:
    - net.ipv4.tcp_rmem = <min> <default> <tcp max>
    - net.ipv4.tcp_wmem = <min> <default> <tcp max>

- 设置IP的最大传输块大小为1500: MTU ≈ TCP MSS + TCP Hdr + IP Hdr

- 检查TcpAckFrequency 的值 (默认是 2)

Linux TCP设置

- 获取缺省的TCP设置
  - cat /proc/sys/net/core/rmem_max
  - cat /proc/sys/net/core/wmem_max
  - cat /proc/sys/net/core/rmem_default
  - cat /proc/sys/net/core/wmem_default
  - cat /proc/sys/net/ipv4/tcp_rmem
  - cat /proc/sys/net/ipv4/tcp_wmem
  - cat /proc/sycat /proc/sys/net/ipv4/tcp_window_scalings

- 进行峰值数据吞吐量测试时，调整TCP参数
  - echo 4194304 > /proc/sys/net/core/rmem_max
  - echo 4194304 > /proc/sys/net/core/wmem_max
  - echo 2097152 > /proc/sys/net/core/rmem_default
  - echo 2097152 > /proc/sys/net/core/wmem_default
  - echo 524288 2097152 4194304 > /proc/sys/net/ipv4/tcp_rmem
  - echo 524288 2097152 4194304 > /proc/sys/net/ipv4/tcp_wmem
  - echo 1 > /proc/sys/net/ipv4/tcp_window_scaling

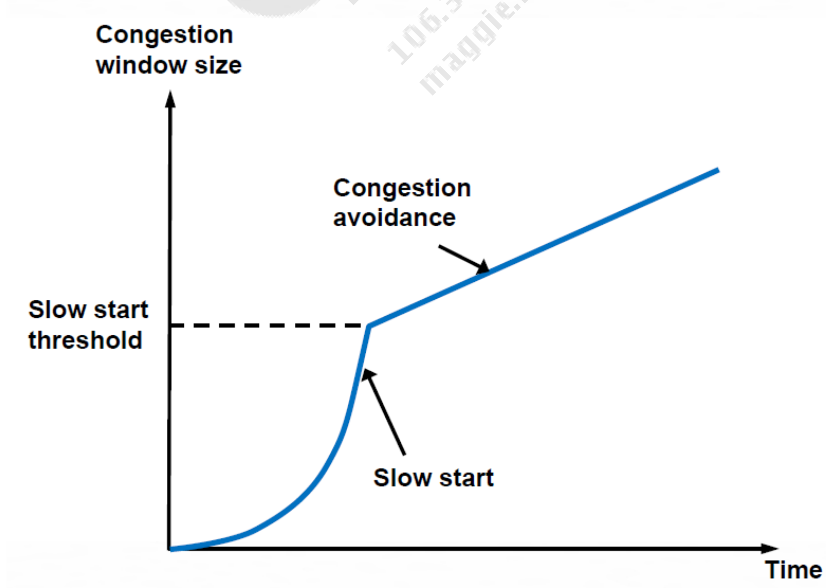
  - 注: 上述参数为调试参数，可以根据调试结果进行适当调整

TCP的特点

- TCP是可靠传输协议，采用滑动窗口来控制发送/接收过程；
- TCP基于ACK确认机制；如果有数据包丢失，接收端发送重复的ACK(DupACK);
- 几个重要的TCP窗口概念：
  - rx_win
    - 接收方的 "接收窗口"：包含在每一个TCP ACK数据包中，其大小是根据流控动态调整的
  - cwnd
    - 拥塞窗口：根据流控来动态调整
  - tx_win
    - 发送方的"发送窗口"：TCP发送方的一个本地变量，接收方无法得知该值
    - tx_win = MIN(rx_win, cwnd)

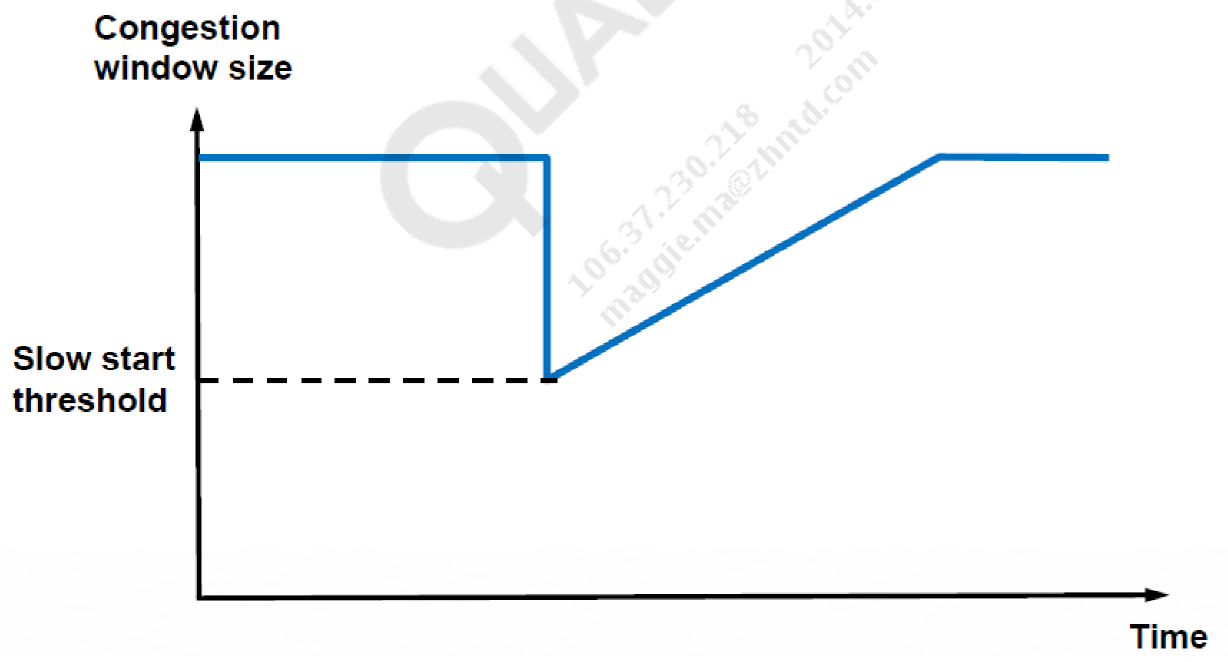
TCP拥塞控制(Congestion Control)

- TCP的拥塞窗口cwnd(Congestion Window Size) 是根据当前信道条件进行动态调整的：
  - 慢启动(Slow Start)阶段：在低于慢启动门限(Slow Start Threshold, ssthresh)时，cwnd呈指数级增长；
  - 拥塞避免(Congestion Avoidance)阶段：在高于ssthresh时，cwnd呈线性增长，以避免拥塞；
  - 一旦由于信道变化，而出现拥塞，则把tx_win设置为当前值的一半
- 下图是慢启动和拥塞避免的算法示意图



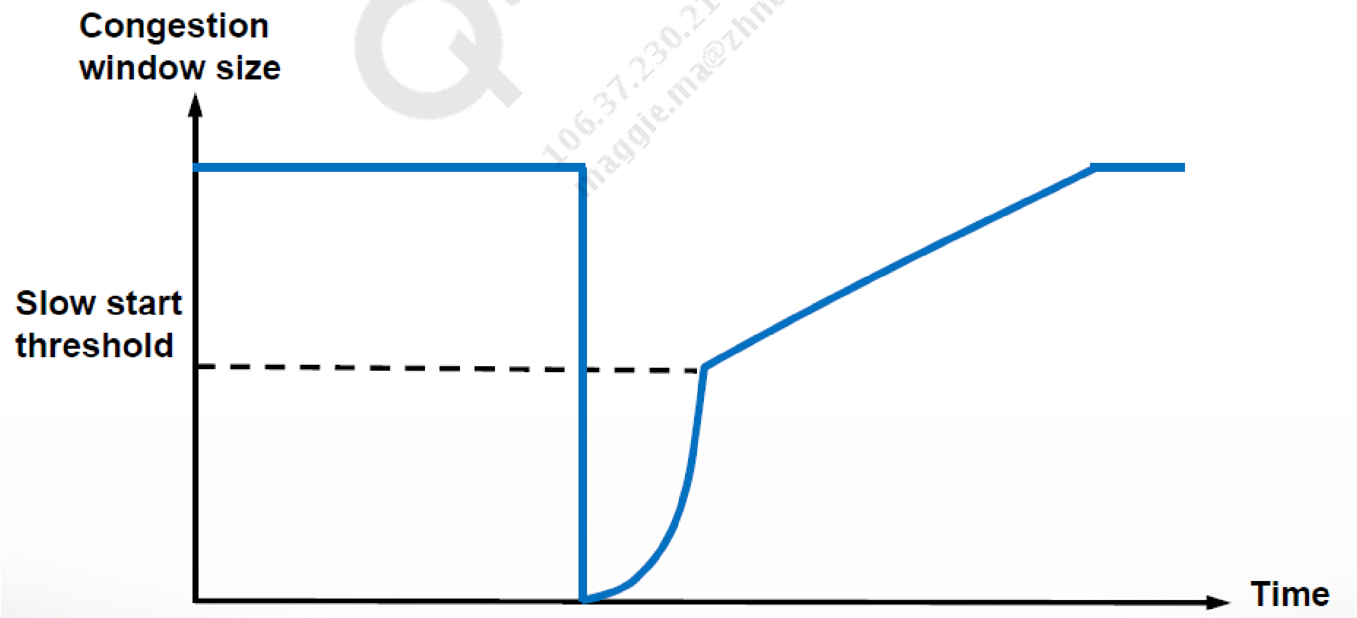
TCP拥塞控制 – 快速重传(Fast Retransmission)

- 当发送方连续收到三个序号相同(重复)的ACK(DupACK)，发送方立即重传该TCP报文，并调整如下TCP运行参数：
  - 将*ssthresh* 降为*tx_win*的一半；
  - 将*cwnd* 设置为*ssthresh* + 3 segment_size



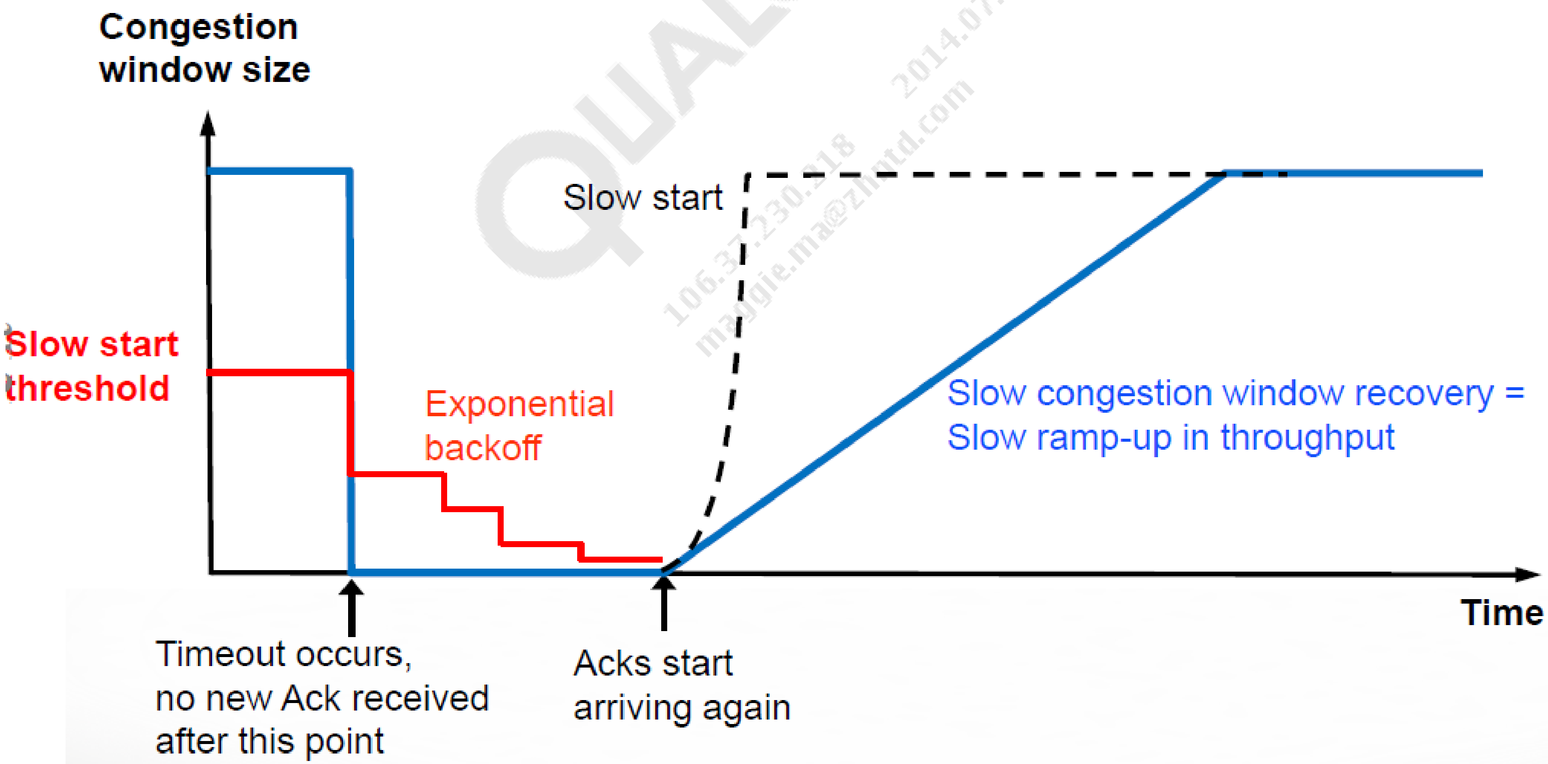
TCP拥塞控制– RTO超时重传

- 当发送方在RTO(ReTransmission Timeout)时间内，没有收到最后连续TCP序号的ACK，则重传该报文：
  - RTO = A + 4D (A – Average RTT, D – Mean deviation)
- 并将TCP运行参数设置如下：
  - 将ssthresh降为tx_win的一半
  - 将cwnd设置为 1 (TCP将重新进入慢启动状态)



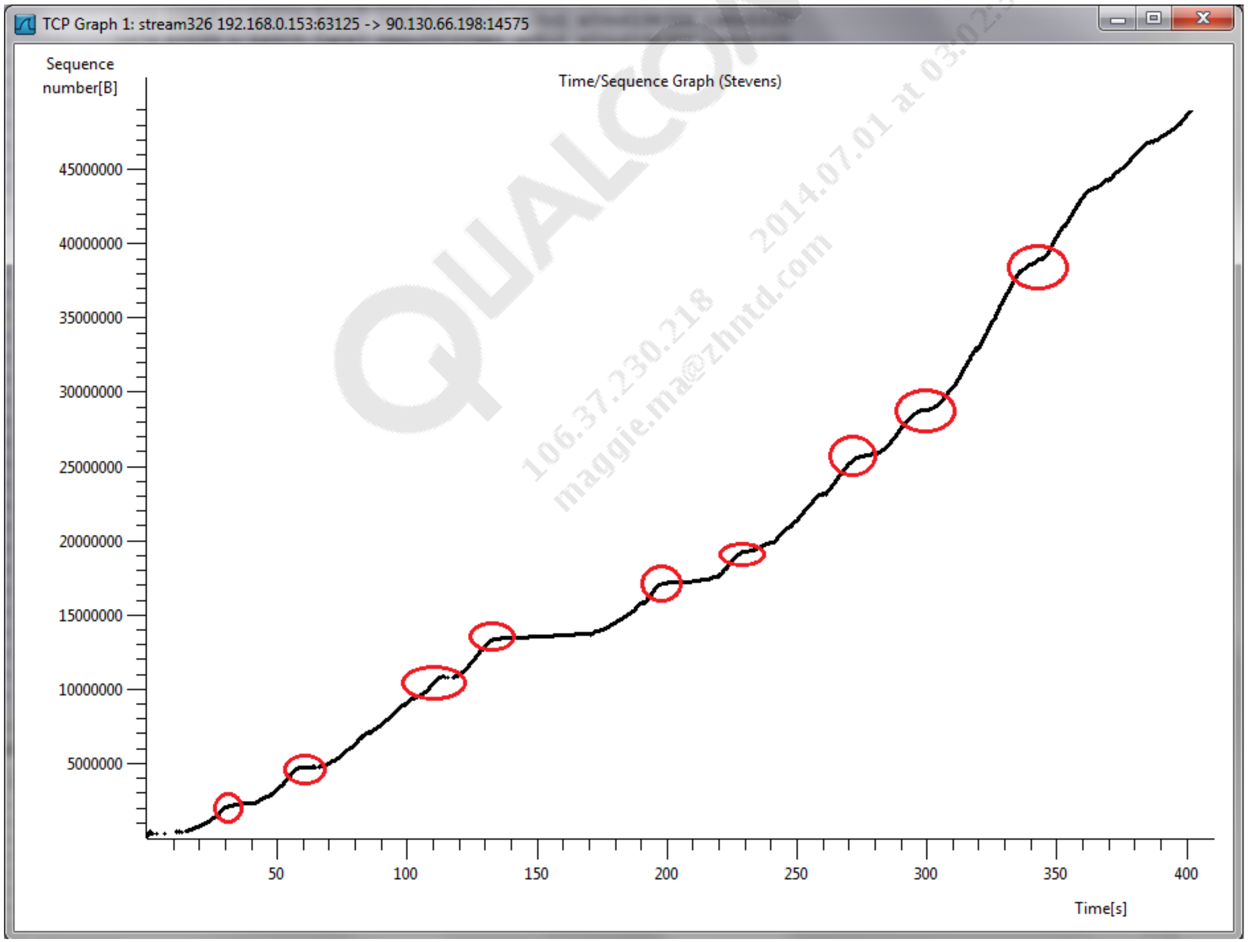
TCP拥塞控制– 连续多个TCP报文丢失

- 如果连续发生多个TCP报文丢失，ssthresh会迅速降低到一个非常低的值
  - 在每次RTO超时重传，ssthresh将降低一半(指数退避,exponential backoff)
  - 此时TCP流量急降到很低的值，且需要很长时间才能恢复



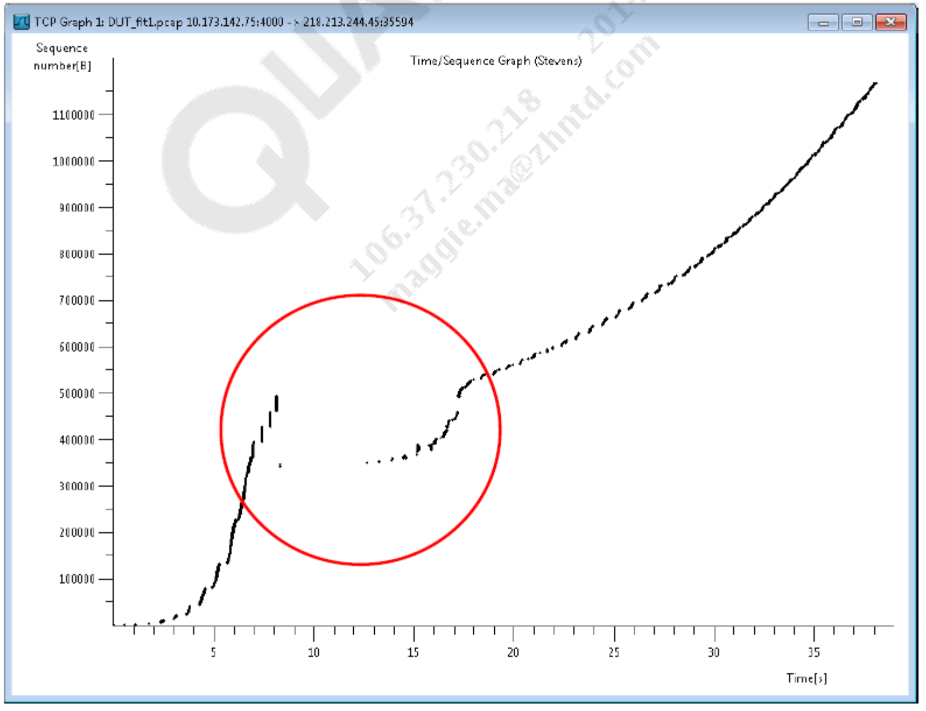
TCP拥塞避免的Wireshark示意图– 快速重传

- TCP快速重传可以快速地恢复TCP拥塞状况，并使得TCP流量不至于一下子降到很低水平。如下图所示：



TCP拥塞避免的Wireshark示意图-- RTO超时重传

- RTO超时重传将使得TCP流量迅速降到一个很低的值；
- 如果发生频繁TCP丢包，证明此时网络异常拥挤或信道条件很差，则TCP需要进入指数退避。此时TCP流量会降到非常低的值，且恢复起来较慢。
- 所以检查TCP是否频繁丢包，可以作为底层信道条件好坏的一个重要指标



双向TCP传输问题

- 问题1：双向(上传+下载)数据传输时，ACK数据包与Data数据包一起传输。如果ACK数据包的优先级与Data数据包一样，则会导致ACK数据包的传输时延比单向传输时慢，尤其是在上下行带宽不对称的条件下。
- 问题2：上行的流控机制可能会导致“ACK聚集(ACK Compression)”，原因如下：
  - 当上行数据传输过快，超过了Um口Watermark的高水位而触发流控；
  - 此时，ACK数据包就会聚集在Rm口的Watermark；
  - 随着Um口的数据逐渐传输，到达Watermark低水位而停止流控；
  - 此时，缓冲在Rm口Watermark的大量ACK数据包以很快的速度经Um口传输到网络；
  - 当网络收到一下子收到多个ACK数据包，TCP服务器端会发送发送大量下行TCP数据包，对下行网络造成冲击，以致于丢失部分TCP数据包而影响吞吐量

提高TCP双向传输吞吐量 方案– Prioritize TCP Acks

- 为了解决上述问题1，从UE侧引入上行TCP ACK数据包优先传输机制
  - 该Feature定义为：FEATURE_DATA_PS_TCP_ACK_PRIO；
  - 在Watermark队列中，优先传输TCP的ACK数据包。即TCP ACK数据包的优先级比TCP Data数据包优先级高；
  - 如果使能该功能：设置NV 67208 = 1 (缺省)

提高TCP双向传输吞吐量 方案 – Back Pressure Removal

- 为了解决上述问题2，从UE侧引入背压消除机制(Back pressure removal)

- 禁止上行(UL)流控机制
  - 周期性检测是否有上行TCP ACK数据包，从而确定是否有下行TCP数据流；
  - 如果有下行数据包，则禁止上行的流控机制。从而消除ACK聚集(ACK Compression)的可能 ；

- 使能上行(UL)流控机制
  - 周期性检测是否有下行TCP数据，如果没有，则使能上行流控机制
  - 设置NV66051(Throughput Optimization Parameters)为：
    - back_pressure_removal = 1 (enable)
    - tx_timestamp_timer_val = 2 (seconds)

附录D：UDP及iperf测试

UDP以及Iperf测试

- TCP的流控机制、以及可靠传输的ACK机制，相对而言比较复杂。而UDP则相对比较简单：
  - 不需要对端应答，无ACK机制；
  - 利用工具，可以任意指定带宽发送数据流；
- 利用上述UDP的特点，可以使用Iperf工具灌包测试。其目的是验证物理信道带宽是否足够。
  - 如果Iperf UDP测试结果不达标，那么TCP一定也不会达标。此时应该怀疑底层物理信道或网络带宽本身就不够大；
  - 如果Iperf UDP测试结果达标，而TCP未达标。证明瓶颈可能不在物理层，需要上层数据业务配合查问题
- 详细Iperf工具使用方法，参考80-N2363-1 (Appendix. C)

Android上进行iperf测试范例

- 目的：测试WiFi-Tethering的下行数据吞吐量 (DUT -> Laptop)

- 将iperf安装进Android手机
  - ```
    c:\adb>adb root
    c:\adb>adb remount
    c:\adb>adb push iperf /etc/iperf
    c:\adb> adb shell chmod 777 /etc/iperf
    ```

- 在Android上运行iperf TCP客户端
  - ```
    c:\adb>adb shell
    # /etc/iperf -c <IP-address of PC client> -w 2M -t 60 -i 1
    ```

- 在Laptop上运行iperf服务器，测试WiFi-Tethering下载吞吐量
  - ```
    c:\iperf>iperf -s -w 2M -i 1
    ```



    ```

# 测试例子

- WiFi-Tethering测试下载速率。测试60秒后，iperf统计的平均吞吐量为45.4Mbps

C:iperf>iperf -s -w 2M -i 1

------------------------------------------------------------

Server listening on TCP port 5001
TCP window size: 2.00 MByte

------------------------------------------------------------

```
[264] local 192.168.43.39 port 5001 connected with 192.168.43.1 port 60232
[ID] Interval   Transfer   Bandwidth
[264] 0.0- 1.0 sec   539 KBytes 4.41 Mbits/sec
[264] 1.0- 2.0 sec   309 KBytes 2.53 Mbits/sec
[264] 2.0- 3.0 sec 2.73 MBytes 22.9 Mbits/sec
[264] 3.0- 4.0 sec 4.44 MBytes 37.2 Mbits/sec
[264] 4.0- 5.0 sec 5.82 MBytes 48.8 Mbits/sec
...
[264] 54.0-55.0 sec 5.21 MBytes 43.7 Mbits/sec
[264] 55.0-56.0 sec 4.84 MBytes 40.6 Mbits/sec
[264] 56.0-57.0 sec 5.22 MBytes 43.8 Mbits/sec
[264] 57.0-58.0 sec 4.72 MBytes 39.6 Mbits/sec
[264] 58.0-59.0 sec 5.68 MBytes 47.6 Mbits/sec
[264] 59.0-60.0 sec 5.25 MBytes 44.0 Mbits/sec
[ID] Interval   Transfer   Bandwidth
[264] 0.0-60.0 sec   325 MBytes 45.4 Mbits/sec
```

附录E：CPU性能参数调整

CPU性能参数调整

- CPU缺省运行在”on-demand”模式，CPU的频率根据负荷动态调整。
- 在峰值吞吐量测试时，有时需要将CPU固定在最大频率以获得最佳性能。
- 设置CPU为“Performance”模式：
  - 将终端设备冷却为常温状态(例如：关闭数据连接/关闭屏幕等)
  - 在ADB Shell运行如下命令(四核为例)：
    - `su`
    - `stop mpdecision`
    - `stop thermal-engine`
    - `echo 1 > /sys/devices/system/cpu/cpu1/online`
    - `echo 1 > /sys/devices/system/cpu/cpu2/online`
    - `echo 1 > /sys/devices/system/cpu/cpu3/online`
    - `echo "performance" > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor`
  - 检查设置是否生效
    - `cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor`

CPU利用率 (1)

- 获取CPU的利用率，检查CPU是否运行在100%利用率上。如果是，需要检查CPU利用率过高的原因。

- 在上下行数据吞吐量测试时，获取CPU利用率的命令：
  - top -d 1 -n 5

- 例子: CPU利用率几乎到达100%

```
# top -d 1 -n 5
top -d 1 -n 5
[Mem: 37512K used, 131160K free, 0K shrd, 620K buff, 8548K cached
CPU:  0.0% usr 58.3% sys  0.0% nic  0.0% idle  0.0% io  0.0% irq 41.6% sirq
Load average: 3.84 1.84 0.72 3/208 1112
[7m  PID  PPID USER     STAT   VSZ %MEM CPU %CPU COMMAND[0m
   32     2 root     RW       0  0.0   0 23.0 [kworker/u:2]
   13     2 root     RW       0  0.0   0 23.0 [kworker/0:1]
   39     2 root     SW       0  0.0   0 23.0 [kworker/u:3]
    3     2 root     RW       0  0.0   0 15.3 [ksoftirqd/0]
 1112   994 root     R     2188  1.3   0  7.6 top -d 1 -n 5
    6     2 root     RW       0  0.0   0  7.6 [kworker/u:0]
  324     1 root     S <   189m114.6   0  0.0 /usr/bin/thermal-engine
319     1 root     S     77624 45.9    0  0.0 /usr/bin/qmuxd
…
```


CPU利用率 (2)

- 在上下行数据性能测试时，获取整体CPU利用率：
  - `mpstat 2 30`   // this is busybox command

- 例子: 每隔2秒，打印CPU利用率。CPU仅约2%空闲.

```
# mpstat 2 30
mpstat 2 30
Linux 3.4.0+ (mdm9x25)  01/06/80        _armv7l_       (1 CPU)

00:04:06     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
00:04:08     all    0.00    0.00   40.00    0.00    0.00   58.00    0.00    0.00    2.00

00:04:08     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
00:04:10     all    1.00    0.00   42.79    0.00    0.00   53.23    0.00    0.00    2.99

00:04:10     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
00:04:12     all    0.50    0.00   40.20    0.00    0.00   57.79    0.00    0.00    1.51

00:04:12     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
00:04:14     all    0.00    0.00   39.90    0.00    0.00   58.08    0.00    0.00    2.02

00:04:14     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
00:04:16     all    0.00    0.00   46.23    0.00    0.00   51.76    0.00    0.00    2.01
```

Questions?

https://support.cdmatech.com