WizFi360

Application – Throughput

Version 1.2

WIZnet Co.,Ltd

Copyright© 2019

History

Ver	Date	Description
1.0	Aug.2019	Initial version
1.1	Sep.2019	Add command mode throughput test result
1.2	Oct.2019	Modify contents about command mode

Contents

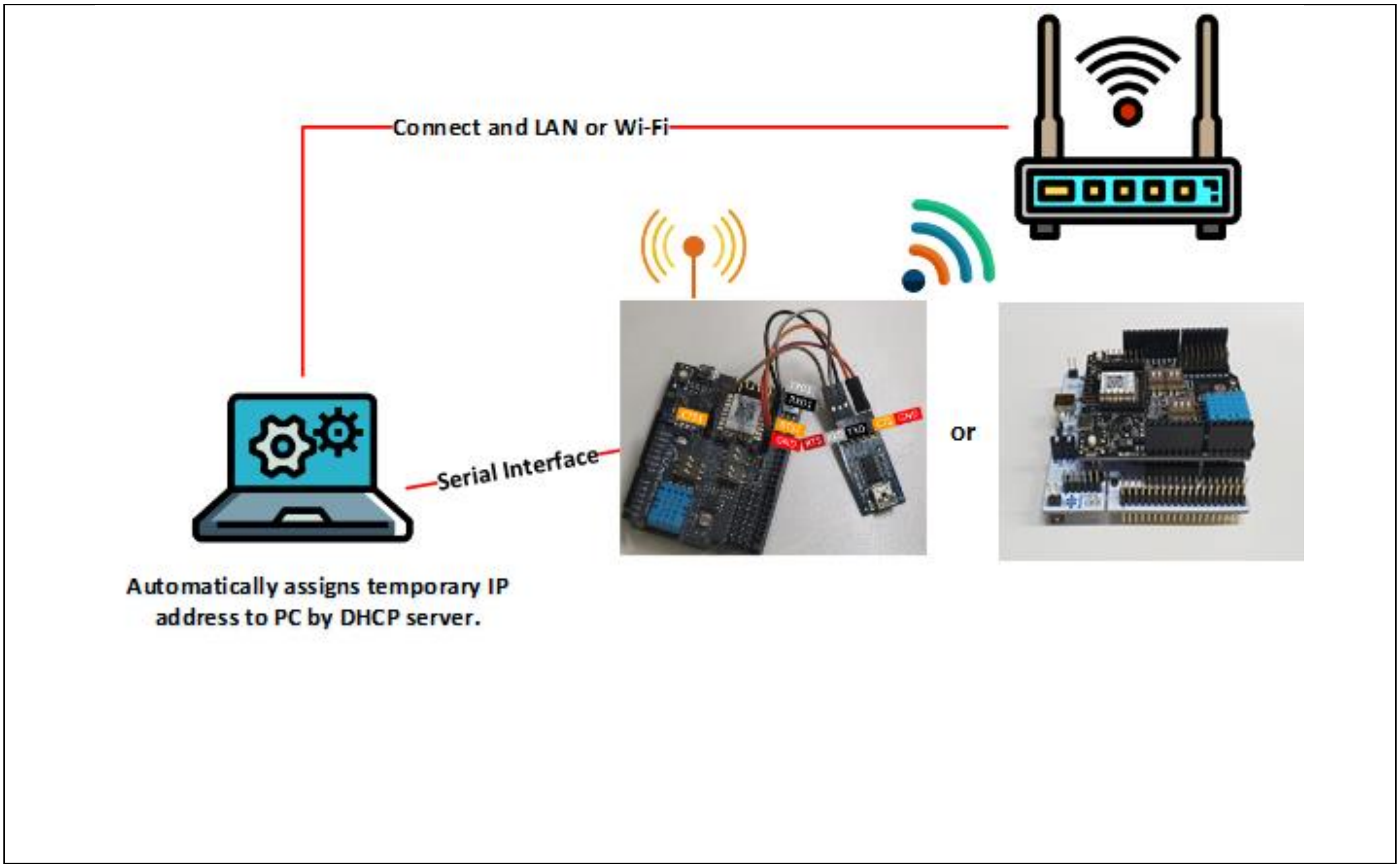
1. Test environment

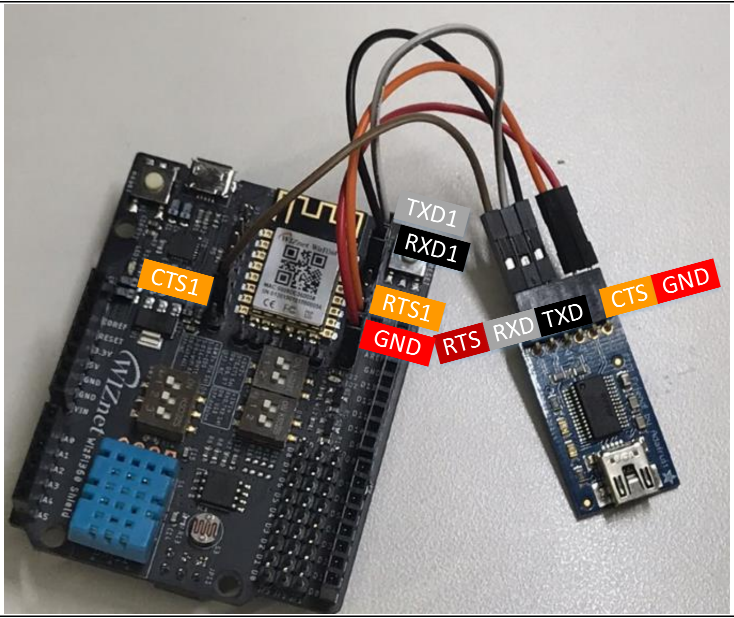
UART Throughput Test 를 하기 위해서는 CTS/RTS 를 이용한 제어 및 WizFi360 제어 Software 가 필요하다.

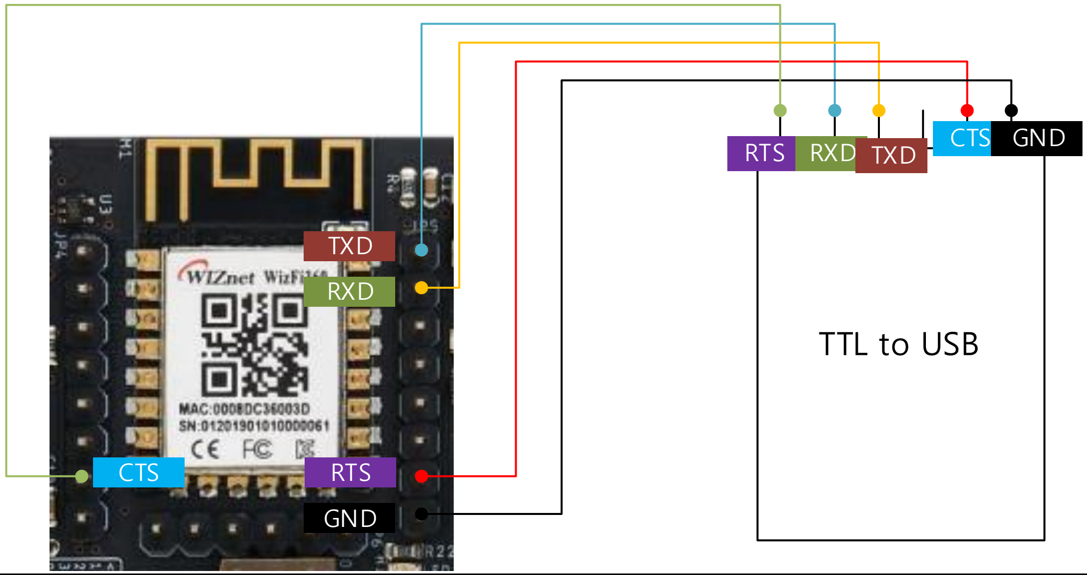
- WizFi360 EVB or WizFi360io
- STM32Fxxx EVB(NUCLEO-F401RE)
- PC
- Serial Tool
  - YAT Serial Tool(Data Mode)
- WizFi360 제어 Software(Command Mode)
- 1Mbyte data file
- WiFi Router(SoftAP mode 를 사용할 경우 제외)

Data Mode 일 경우에는 YAT Serial Tool 를 이용하여 RTS/CTS 를 설정하고, DTR 로 Data Read 신호 설정해준다. 그런 후 RTS/CTS 가 제어하면서 데이터 전송이 이루어진다.

Command Mode 일 경우에는 AT+CIPSENDBUF command 를 이용하여 한번에 최대 보낼 수 있는 데이터 Byte 수인 2048 을 설정한 후 2048 Byte 크기의 Data 를 보내고, 또다시 AT+CIPSENDBUF 와 데이터를 반복해가면서 데이터 전송이 이루어진다.

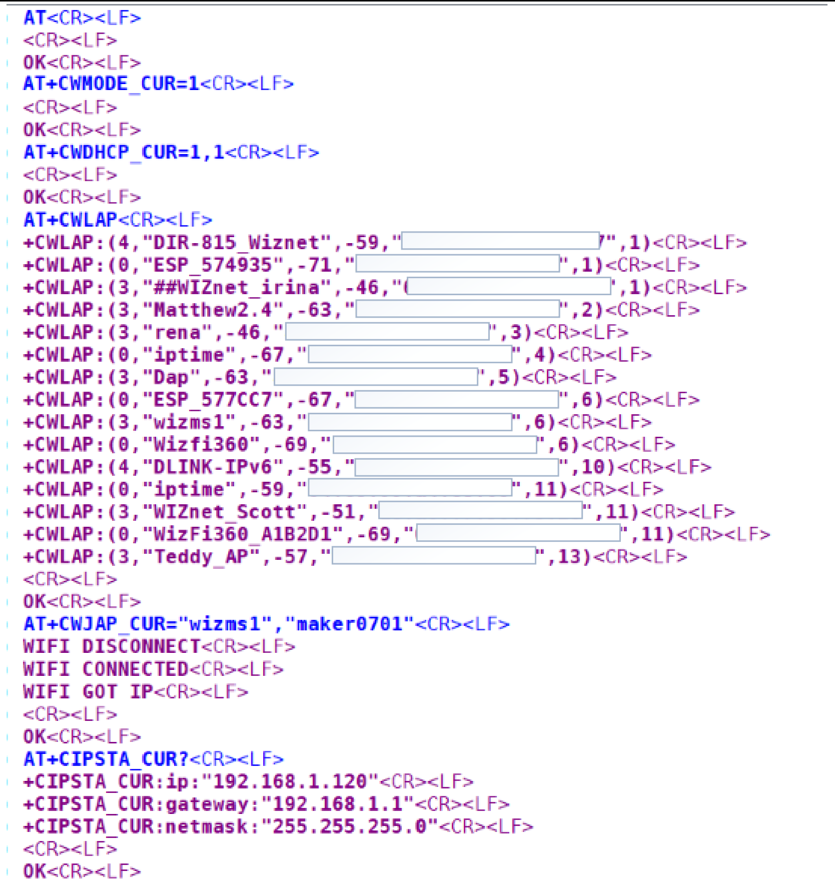

Connect and LAN or Wi-Fi
Serial Interface
or
Automatically assigns temporary IP address to PC by DHCP server.


TXD1
RXD1
CTS1
RTS1
GND
RTS
RXD
TXD
CTS
GND

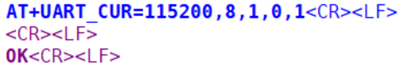

TXD
RXD
CTS
RTS
GND
RTS
RXD
TXD
CTS
GND
TTL to USB

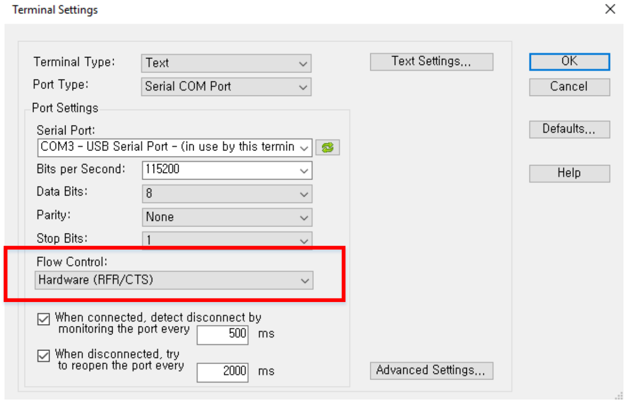
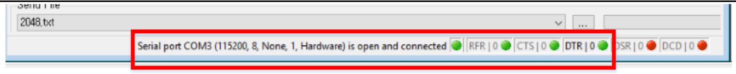
2. Using Serial command

- Station Mode

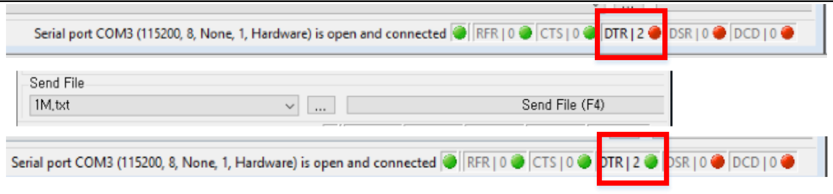
AT command	Terminal
AT AT+CWMODE_CUR=1 AT+CWDHCP_CUR=1,1 AT+CWLAP AT+CWJAP_CUR="wizms1","maker0701" AT+CIPSTA_CUR?	 AT<CR><LF> <CR><LF> OK<CR><LF> AT+CWMODE_CUR=1<CR><LF> <CR><LF> OK<CR><LF> AT+CWDHCP_CUR=1,1<CR><LF> <CR><LF> OK<CR><LF> AT+CWLAP<CR><LF> +CWLAP:(4,"DIR-815_Wiznet",-59,"[redacted]",1)<CR><LF> +CWLAP:(0,"ESP_574935",-71,"[redacted]",1)<CR><LF> +CWLAP:(3,"##WIZnet_irina",-46,"[redacted]',1)<CR><LF> +CWLAP:(3,"Matthew2.4",-63,"[redacted]",2)<CR><LF> +CWLAP:(3,"rena",-46,"[redacted]",3)<CR><LF> +CWLAP:(0,"iptime",-67,"[redacted]",4)<CR><LF> +CWLAP:(3,"Dap",-63,"[redacted]',5)<CR><LF> +CWLAP:(0,"ESP_577CC7",-67,"[redacted]",6)<CR><LF> +CWLAP:(3,"wizms1",-63,"[redacted]",6)<CR><LF> +CWLAP:(0,"Wizfi360",-69,"[redacted]",6)<CR><LF> +CWLAP:(4,"DLINK-IPv6",-55,"[redacted]",10)<CR><LF> +CWLAP:(0,"iptime",-59,"[redacted]",11)<CR><LF> +CWLAP:(3,"WIZnet_Scott",-51,"[redacted]",11)<CR><LF> +CWLAP:(0,"WizFi360_A1B2D1",-69,"[redacted]",11)<CR><LF> +CWLAP:(3,"Teddy_AP",-57,"[redacted]",13)<CR><LF> <CR><LF> OK<CR><LF> AT+CWJAP_CUR="wizms1","maker0701"<CR><LF> WIFI DISCONNECT<CR><LF> WIFI CONNECTED<CR><LF> WIFI GOT IP<CR><LF> <CR><LF> OK<CR><LF> AT+CIPSTA_CUR?<CR><LF> +CIPSTA_CUR:ip:"192.168.1.120"<CR><LF> +CIPSTA_CUR:gateway:"192.168.1.1"<CR><LF> +CIPSTA_CUR:netmask:"255.255.255.0"<CR><LF> <CR><LF> OK<CR><LF>

- UART CTS/RTS Setting

AT command	Terminal
AT+CWUART_CUR = 115200,8,1,0,1	 AT+UART_CUR=115200,8,1,0,1<CR><LF> <CR><LF> OK<CR><LF>
Terminal Setting	

1. Ctrl+Shift+S > Open the Settings 2. Flow Control안에 Hardware(RFR/CTS)로 변경	 Terminal Settings Terminal Type: Text Port Type: Serial COM Port Port Settings Serial Port: COM3 - USB Serial Port - (in use by this termin Bits per Second: 115200 Data Bits: 8 Parity: None Stop Bits: 1 Flow Control: Hardware (RFR/CTS) ☑ When connected, detect disconnect by monitoring the port every 500 ms ☑ When disconnected, try to reopen the port every 2000 ms
3. Terminal창 아래에 오면 CTS/DTR이 초록으로 들어온 것을 확인할 수 있다.	 2048.txt Serial port COM3 (115200, 8, None, 1, Hardware) is open and connected 🟢 RFR | 0 🟢 CTS | 0 🟢 DTR | 0 🟢 DSR | 0 🔴 DCD | 0 🔴

- TCP Client /Data mode

AT command	Terminal
AT+CIPSTART="TCP","192.168.100.27",5001 AT+CIPMODE=1 AT+CIPSEND	AT+CIPSTART="TCP","192.168.100.27",5001<CR><LF> CONNECT<CR><LF> <CR><LF> OK<CR><LF> AT+CIPMODE=1<CR><LF> <CR><LF> OK<CR><LF> AT+CIPSEND<CR><LF> <CR><LF> >
	Terminal Setting
1. DTR이 빨간불일 때, 1M.txt를 보내고, 2. DTR를 클릭해서 초록불로 바뀌면 데이터가 Serial을 통해 전송되게 된다.	 Serial port COM3 (115200, 8, None, 1, Hardware) is open and connected 🟢 RFR | 0 🟢 CTS | 0 🟢 DTR | 2 🔴 DSR | 0 🔴 DCD | 0 🔴 Send File 1M.txt Serial port COM3 (115200, 8, None, 1, Hardware) is open and connected 🟢 RFR | 0 🟢 CTS | 0 🟢 DTR | 2 🟢 DSR | 0 🔴 DCD | 0 🔴

- TCP Client / Command mode

AT command	Example Code
AT+CIPSTART="TCP","192.168.100.27",5001 AT+CIPMODE=0 AT+CIPSENDBUF=2048 Send the 2048byte data * 512times = 1Mbyte	

```c
int8_t deviceTestThroughput_WizFi360(char *data, int len)
{
    int8_t ret = RET_NOK;
    int cnt;
    int segid = 0;

    for(cnt = 0; cnt < (len / 4); cnt++)  // 2k * 512 = 1M
    {
        if(ATCmdParser_send("AT+CIPSENDBUF=%d", len)&& ATCmdParser_recv("OK") && ATCmdParser_recv(">"))
        {
                if(ATCmdParser_send("%s", data) && ATCmdParser_recv("%d,SEND OK", &segid))
                {
                    ret = RET_OK;
                }
                else
                {
                    printf("Write data : failed\r\n");
                }
        }
        else
        {
            printf("Set buffer : failed\r\n");
        }
    }

    return ret;
}
```

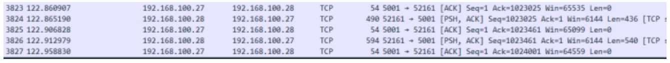
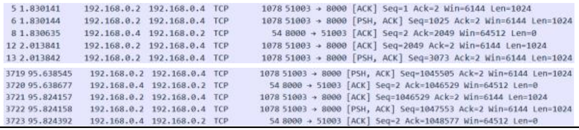
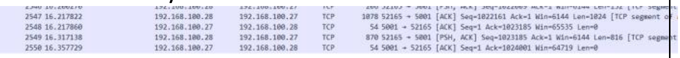
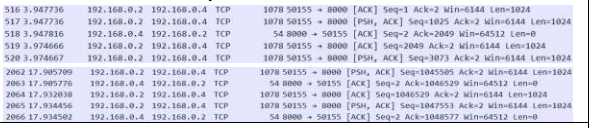
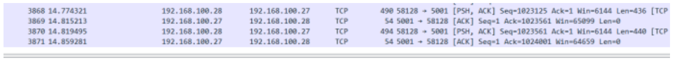
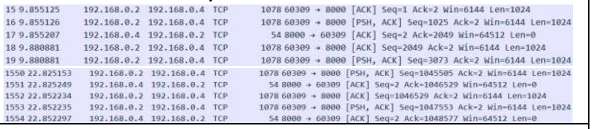
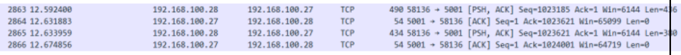
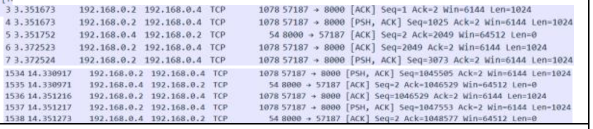
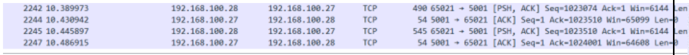
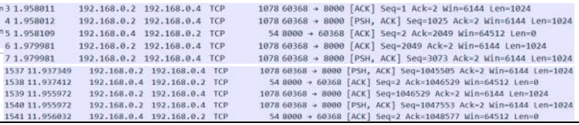
3. The result of UART Throughput

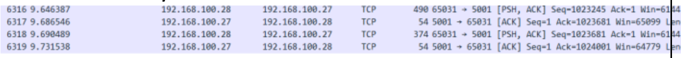
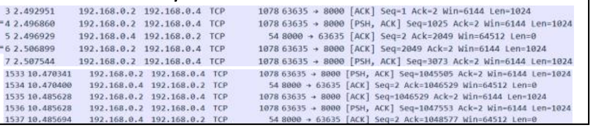
1Mbyte를 PC 혹은 WizFi360을 제어하는 MCU에서 WizFi360의 Serial(UART1)로 데이터를 보내고, TCP Server로 데이터를 전송한다.

Baud rate	Data mode		Command mode	
	Time	Speed(bit/s)	Time	Speed(bit/s)
115200	123s	66K	93.9s	87.2K
921600	16.3s	502K	14.0s	585.1K
1000000	14.9s	550K	13.0s	630.2K
1250000	12.7s	645K	11.0s	744.7K
1500000	10.5s	780K	10.0s	819.2K
2000000	9.7s	845K	8.0s	1.0M

해당 속도는 Wireshark를 이용하여, 데이터 전송시작부터 완료되는 시점까지의 시간을 측정한 것은 Appendix 1을 보면 된다.

Appendix 1

Baud rate	Data mode	Command mode
115200	123s : 66Kbit/s  3823 122.860907 192.168.100.27 192.168.100.28 TCP 54 5001 → 52161 [ACK] Seq=1 Ack=1023025 Win=65535 Len=0 3824 122.865190 192.168.100.28 192.168.100.27 TCP 490 52161 → 5001 [PSH, ACK] Seq=1023025 Ack=1 Win=6144 Len=436 [TCP s 3825 122.906828 192.168.100.27 192.168.100.28 TCP 54 5001 → 52161 [ACK] Seq=1 Ack=1023461 Win=65099 Len=0 3826 122.912979 192.168.100.28 192.168.100.27 TCP 594 52161 → 5001 [PSH, ACK] Seq=1023461 Ack=1 Win=6144 Len=540 [TCP s 3827 122.958830 192.168.100.27 192.168.100.28 TCP 54 5001 → 52161 [ACK] Seq=1 Ack=1024001 Win=64559 Len=0	93.9s : 87.2Kbit/s  5 1.830141 192.168.0.2 192.168.0.4 TCP 1078 51003 → 8000 [ACK] Seq=1 Ack=2 Win=6144 Len=1024 6 1.830144 192.168.0.2 192.168.0.4 TCP 1078 51003 → 8000 [PSH, ACK] Seq=1025 Ack=2 Win=6144 Len=1024 8 1.830635 192.168.0.4 192.168.0.2 TCP 54 8000 → 51003 [ACK] Seq=2 Ack=2049 Win=64512 Len=0 12 2.013841 192.168.0.2 192.168.0.4 TCP 1078 51003 → 8000 [ACK] Seq=2049 Ack=2 Win=6144 Len=1024 13 2.013842 192.168.0.2 192.168.0.4 TCP 1078 51003 → 8000 [PSH, ACK] Seq=3073 Ack=2 Win=6144 Len=1024 3719 95.638545 192.168.0.2 192.168.0.4 TCP 1078 51003 → 8000 [PSH, ACK] Seq=1045505 Ack=2 Win=6144 Len=1024 3720 95.638677 192.168.0.4 192.168.0.2 TCP 54 8000 → 51003 [ACK] Seq=2 Ack=1046529 Win=64512 Len=0 3721 95.824157 192.168.0.2 192.168.0.4 TCP 1078 51003 → 8000 [ACK] Seq=1046529 Ack=2 Win=6144 Len=1024 3722 95.824158 192.168.0.2 192.168.0.4 TCP 1078 51003 → 8000 [PSH, ACK] Seq=1047553 Ack=2 Win=6144 Len=1024 3723 95.824392 192.168.0.4 192.168.0.2 TCP 54 8000 → 51003 [ACK] Seq=2 Ack=1048577 Win=64512 Len=0
921600	16.3s :502Kbit/s  2547 16.217822 192.168.100.28 192.168.100.27 TCP 1078 52165 → 5001 [ACK] Seq=1022161 Ack=1 Win=6144 Len=1024 [TCP segment of 2548 16.217860 192.168.100.27 192.168.100.28 TCP 54 5001 → 52165 [ACK] Seq=1 Ack=1023185 Win=65535 Len=0 2549 16.317138 192.168.100.28 192.168.100.27 TCP 870 52165 → 5001 [PSH, ACK] Seq=1023185 Ack=1 Win=6144 Len=816 [TCP segment 2550 16.357729 192.168.100.27 192.168.100.28 TCP 54 5001 → 52165 [ACK] Seq=1 Ack=1024001 Win=64719 Len=0	14.0s : 585.1Kbit/s  516 3.947736 192.168.0.2 192.168.0.4 TCP 1078 50155 → 8000 [ACK] Seq=1 Ack=2 Win=6144 Len=1024 517 3.947736 192.168.0.2 192.168.0.4 TCP 1078 50155 → 8000 [PSH, ACK] Seq=1025 Ack=2 Win=6144 Len=1024 518 3.947816 192.168.0.4 192.168.0.2 TCP 54 8000 → 50155 [ACK] Seq=2 Ack=2049 Win=64512 Len=0 519 3.974666 192.168.0.2 192.168.0.4 TCP 1078 50155 → 8000 [ACK] Seq=2049 Ack=2 Win=6144 Len=1024 520 3.974667 192.168.0.2 192.168.0.4 TCP 1078 50155 → 8000 [PSH, ACK] Seq=3073 Ack=2 Win=6144 Len=1024 2062 17.905709 192.168.0.2 192.168.0.4 TCP 1078 50155 → 8000 [PSH, ACK] Seq=1045505 Ack=2 Win=6144 Len=1024 2063 17.905776 192.168.0.4 192.168.0.2 TCP 54 8000 → 50155 [ACK] Seq=2 Ack=1046529 Win=64512 Len=0 2064 17.932038 192.168.0.2 192.168.0.4 TCP 1078 50155 → 8000 [ACK] Seq=1046529 Ack=2 Win=6144 Len=1024 2065 17.934456 192.168.0.2 192.168.0.4 TCP 1078 50155 → 8000 [PSH, ACK] Seq=1047553 Ack=2 Win=6144 Len=1024 2066 17.934502 192.168.0.4 192.168.0.2 TCP 54 8000 → 50155 [ACK] Seq=2 Ack=1048577 Win=64512 Len=0
1000000	14.9s : 550Kbit/s  3868 14.774321 192.168.100.28 192.168.100.27 TCP 490 58128 → 5001 [PSH, ACK] Seq=1023125 Ack=1 Win=6144 Len=436 [TCP 3869 14.815213 192.168.100.27 192.168.100.28 TCP 54 5001 → 58128 [ACK] Seq=1 Ack=1023561 Win=65099 Len=0 3870 14.819495 192.168.100.28 192.168.100.27 TCP 494 58128 → 5001 [PSH, ACK] Seq=1023561 Ack=1 Win=6144 Len=440 [TCP 3871 14.859281 192.168.100.27 192.168.100.28 TCP 54 5001 → 58128 [ACK] Seq=1 Ack=1024001 Win=64659 Len=0	13.0s : 630.2Kbit/s  15 9.855125 192.168.0.2 192.168.0.4 TCP 1078 60309 → 8000 [ACK] Seq=1 Ack=2 Win=6144 Len=1024 16 9.855126 192.168.0.2 192.168.0.4 TCP 1078 60309 → 8000 [PSH, ACK] Seq=1025 Ack=2 Win=6144 Len=1024 17 9.855207 192.168.0.4 192.168.0.2 TCP 54 8000 → 60309 [ACK] Seq=2 Ack=2049 Win=64512 Len=0 18 9.880881 192.168.0.2 192.168.0.4 TCP 1078 60309 → 8000 [ACK] Seq=2049 Ack=2 Win=6144 Len=1024 19 9.880881 192.168.0.2 192.168.0.4 TCP 1078 60309 → 8000 [PSH, ACK] Seq=3073 Ack=2 Win=6144 Len=1024 1550 22.825153 192.168.0.2 192.168.0.4 TCP 1078 60309 → 8000 [PSH, ACK] Seq=1045505 Ack=2 Win=6144 Len=1024 1551 22.825249 192.168.0.4 192.168.0.2 TCP 54 8000 → 60309 [ACK] Seq=2 Ack=1046529 Win=64512 Len=0 1552 22.852234 192.168.0.2 192.168.0.4 TCP 1078 60309 → 8000 [ACK] Seq=1046529 Ack=2 Win=6144 Len=1024 1553 22.852235 192.168.0.2 192.168.0.4 TCP 1078 60309 → 8000 [PSH, ACK] Seq=1047553 Ack=2 Win=6144 Len=1024 1554 22.852297 192.168.0.4 192.168.0.2 TCP 54 8000 → 60309 [ACK] Seq=2 Ack=1048577 Win=64512 Len=0
1250000	12.7s : 645Kbit/s  2863 12.592400 192.168.100.28 192.168.100.27 TCP 490 58136 → 5001 [PSH, ACK] Seq=1023185 Ack=1 Win=6144 Len=436 [T 2864 12.631883 192.168.100.27 192.168.100.28 TCP 54 5001 → 58136 [ACK] Seq=1 Ack=1023621 Win=65099 Len=0 2865 12.633959 192.168.100.28 192.168.100.27 TCP 434 58136 → 5001 [PSH, ACK] Seq=1023621 Ack=1 Win=6144 Len=380 [T 2866 12.674856 192.168.100.27 192.168.100.28 TCP 54 5001 → 58136 [ACK] Seq=1 Ack=1024001 Win=64719 Len=0	11.0s : 744.7Kbit/s  3 3.351673 192.168.0.2 192.168.0.4 TCP 1078 57187 → 8000 [ACK] Seq=1 Ack=2 Win=6144 Len=1024 4 3.351673 192.168.0.2 192.168.0.4 TCP 1078 57187 → 8000 [PSH, ACK] Seq=1025 Ack=2 Win=6144 Len=1024 5 3.351752 192.168.0.4 192.168.0.2 TCP 54 8000 → 57187 [ACK] Seq=2 Ack=2049 Win=64512 Len=0 6 3.372523 192.168.0.2 192.168.0.4 TCP 1078 57187 → 8000 [ACK] Seq=2049 Ack=2 Win=6144 Len=1024 7 3.372524 192.168.0.2 192.168.0.4 TCP 1078 57187 → 8000 [PSH, ACK] Seq=3073 Ack=2 Win=6144 Len=1024 1534 14.330917 192.168.0.2 192.168.0.4 TCP 1078 57187 → 8000 [PSH, ACK] Seq=1045505 Ack=2 Win=6144 Len=1024 1535 14.330971 192.168.0.4 192.168.0.2 TCP 54 8000 → 57187 [ACK] Seq=2 Ack=1046529 Win=64512 Len=0 1536 14.351216 192.168.0.2 192.168.0.4 TCP 1078 57187 → 8000 [ACK] Seq=1046529 Ack=2 Win=6144 Len=1024 1537 14.351217 192.168.0.2 192.168.0.4 TCP 1078 57187 → 8000 [PSH, ACK] Seq=1047553 Ack=2 Win=6144 Len=1024 1538 14.351273 192.168.0.4 192.168.0.2 TCP 54 8000 → 57187 [ACK] Seq=2 Ack=1048577 Win=64512 Len=0
1500000	10.5s : 780Kbit/s  2242 10.389973 192.168.100.28 192.168.100.27 TCP 490 65021 → 5001 [PSH, ACK] Seq=1023074 Ack=1 Win=6144 Len 2244 10.430942 192.168.100.27 192.168.100.28 TCP 54 5001 → 65021 [ACK] Seq=1 Ack=1023510 Win=65099 Len=0 2245 10.445897 192.168.100.28 192.168.100.27 TCP 545 65021 → 5001 [PSH, ACK] Seq=1023510 Ack=1 Win=6144 Len 2247 10.486915 192.168.100.27 192.168.100.28 TCP 54 5001 → 65021 [ACK] Seq=1 Ack=1024001 Win=64608 Len=0	10.0s : 819.2Kbit/s  3 1.958011 192.168.0.2 192.168.0.4 TCP 1078 60368 → 8000 [ACK] Seq=1 Ack=2 Win=6144 Len=1024 4 1.958012 192.168.0.2 192.168.0.4 TCP 1078 60368 → 8000 [PSH, ACK] Seq=1025 Ack=2 Win=6144 Len=1024 5 1.958109 192.168.0.4 192.168.0.2 TCP 54 8000 → 60368 [ACK] Seq=2 Ack=2049 Win=64512 Len=0 6 1.979981 192.168.0.2 192.168.0.4 TCP 1078 60368 → 8000 [ACK] Seq=2049 Ack=2 Win=6144 Len=1024 7 1.979981 192.168.0.2 192.168.0.4 TCP 1078 60368 → 8000 [PSH, ACK] Seq=3073 Ack=2 Win=6144 Len=1024 1537 11.937349 192.168.0.2 192.168.0.4 TCP 1078 60368 → 8000 [PSH, ACK] Seq=1045505 Ack=2 Win=6144 Len=1024 1538 11.937412 192.168.0.4 192.168.0.2 TCP 54 8000 → 60368 [ACK] Seq=2 Ack=1046529 Win=64512 Len=0 1539 11.955972 192.168.0.2 192.168.0.4 TCP 1078 60368 → 8000 [ACK] Seq=1046529 Ack=2 Win=6144 Len=1024 1540 11.955972 192.168.0.2 192.168.0.4 TCP 1078 60368 → 8000 [PSH, ACK] Seq=1047553 Ack=2 Win=6144 Len=1024 1541 11.956032 192.168.0.4 192.168.0.2 TCP 54 8000 → 60368 [ACK] Seq=2 Ack=1048577 Win=64512 Len=0

2000000	9.7s : 845Kbit/s  6316 9.646387 192.168.100.28 192.168.100.27 TCP 490 65031 → 5001 [PSH, ACK] Seq=1023245 Ack=1 Win=6144 6317 9.686546 192.168.100.27 192.168.100.28 TCP 54 5001 → 65031 [ACK] Seq=1 Ack=1023681 Win=65099 Len= 6318 9.690489 192.168.100.28 192.168.100.27 TCP 374 65031 → 5001 [PSH, ACK] Seq=1023681 Ack=1 Win=6144 6319 9.731538 192.168.100.27 192.168.100.28 TCP 54 5001 → 65031 [ACK] Seq=1 Ack=1024001 Win=64779 Len=	8.0s : 1.0Mbit/s  3 2.492951 192.168.0.2 192.168.0.4 TCP 1078 63635 → 8000 [ACK] Seq=1 Ack=2 Win=6144 Len=1024 4 2.496860 192.168.0.2 192.168.0.4 TCP 1078 63635 → 8000 [PSH, ACK] Seq=1025 Ack=2 Win=6144 Len=1024 5 2.496929 192.168.0.4 192.168.0.2 TCP 54 8000 → 63635 [ACK] Seq=2 Ack=2049 Win=64512 Len=0 6 2.506899 192.168.0.2 192.168.0.4 TCP 1078 63635 → 8000 [ACK] Seq=2049 Ack=2 Win=6144 Len=1024 7 2.507544 192.168.0.2 192.168.0.4 TCP 1078 63635 → 8000 [PSH, ACK] Seq=3073 Ack=2 Win=6144 Len=1024 1533 10.470341 192.168.0.2 192.168.0.4 TCP 1078 63635 → 8000 [PSH, ACK] Seq=1045505 Ack=2 Win=6144 Len=1024 1534 10.470400 192.168.0.4 192.168.0.2 TCP 54 8000 → 63635 [ACK] Seq=2 Ack=1046529 Win=64512 Len=0 1535 10.485628 192.168.0.2 192.168.0.4 TCP 1078 63635 → 8000 [ACK] Seq=1046529 Ack=2 Win=6144 Len=1024 1536 10.485628 192.168.0.2 192.168.0.4 TCP 1078 63635 → 8000 [PSH, ACK] Seq=1047553 Ack=2 Win=6144 Len=1024 1537 10.485694 192.168.0.4 192.168.0.2 TCP 54 8000 → 63635 [ACK] Seq=2 Ack=1048577 Win=64512 Len=0