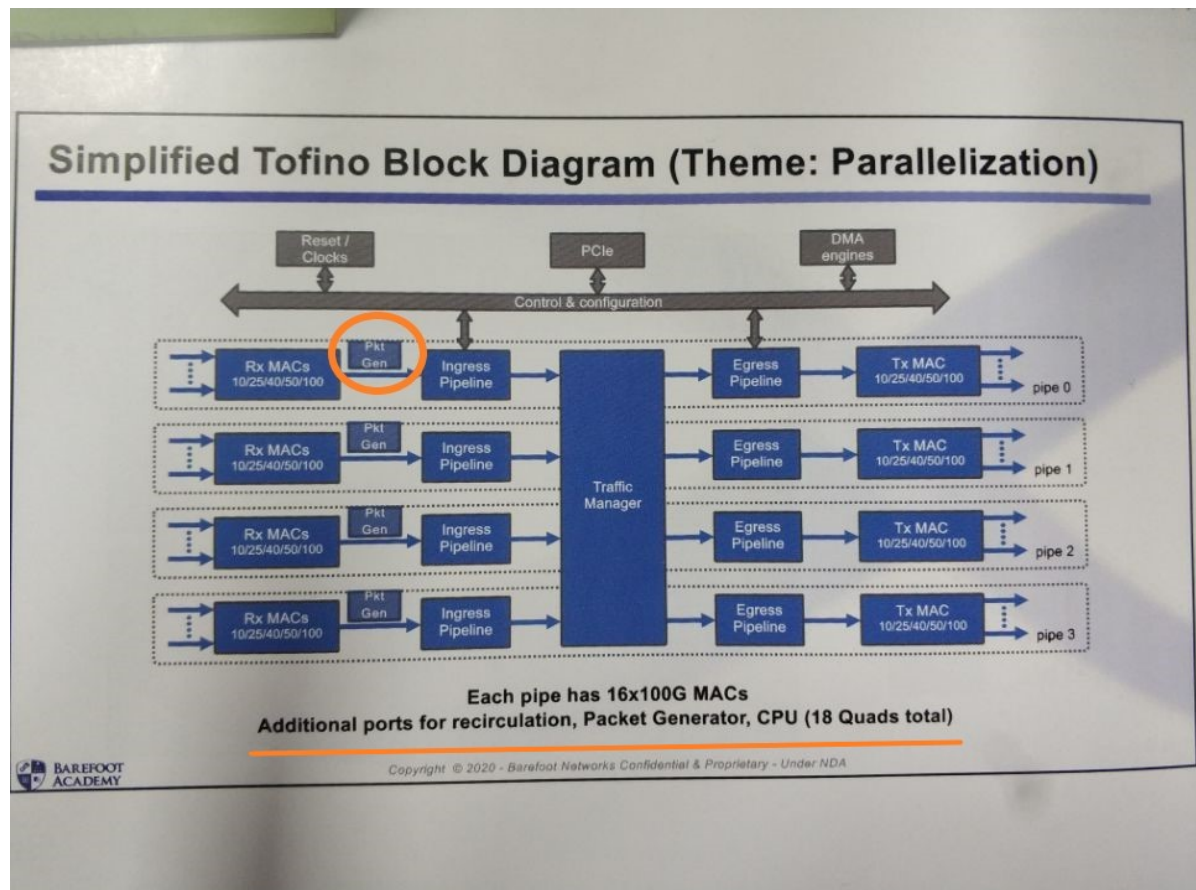


# Tofino数据面数据包生成器——Pkt Gen

## Pkt Gen

Tofino上实现了数据面的数据包生成器Pkt Gen，理论上可以大大提高数据包的生成速率，但是在Barefoot的P4讲义中对于Tofino上的数据面的数据包生成器Pkt Gen只是一句话带过，所以针对Pkt Gen进行了进一步的调研。



## 不同的触发模式

```
// -----  
// PACKET GENERATION  
// -----  
// Packet generator supports up to 8 applications and a total of 16KB packet  
// payload. Each application is associated with one of the four trigger types:  
// - One-time timer  
// - Periodic timer  
// - Port down  
// - Packet recirculation  
// For recirculated packets, the event fires when the first 32 bits of the  
// recirculated packet matches the application match value and mask.  
// A triggered event may generate programmable number of batches with  
// programmable number of packets per batch.
```

Pkt Gen支持四种不同的触发模式：

- One-time timer
- Periodic timer
- Port down
- Packet recirculation

前两种是计时器触发，第三种是通过切换交换机端口的状态（up/down）触发生成数据包，第四种触发模式是数据包的循环触发。

不同的触发模式对应生成的数据包头部字段略有差异，但都是6个字节：

```
header pktgen_timer_header_t {
    @padding bit<3> _pad1;
    bit<2> pipe_id;           // Pipe id
    bit<3> app_id;            // Application id
    @padding bit<8> _pad2;

    bit<16> batch_id;         // Start at 0 and increment to a
                             // programmed number
    bit<16> packet_id;        // Start at 0 and increment to a
                             // programmed number
}

header pktgen_port_down_header_t {
    @padding bit<3> _pad1;
    bit<2> pipe_id;           // Pipe id
    bit<3> app_id;            // Application id
    @padding bit<15> _pad2;
    bit<9> port_num;          // Port number

    bit<16> packet_id;        // Start at 0 and increment to a
                             // programmed number
}

header pktgen_recirc_header_t {
    @padding bit<3> _pad1;
    bit<2> pipe_id;           // Pipe id
    bit<3> app_id;            // Application id
    bit<24> key;              // Key from the recirculated packet

    bit<16> packet_id;        // Start at 0 and increment to a
                             // programmed number
}
```

在计时器触发模式（其他模式类似）中，可以通过自定义batch\_id和packet\_id决定生成数据包的数量，数据包生成总数=batch\_id \* packet\_id，理论上控制面一条发包指令，最多在数据面生成数据包数量为 $2^{16} * 2^{16}$ ，这是一个非常大的数字。

## 数据面如何生成数据包

首先通过控制面生成一个指定类型（可以实现自定义某些字段）的数据包，在Barefoot SDE中已经实现了以下类型的数据包生成函数：

- TCP
- TCPv6
- UDP
- SRv6

- GENEVE
- GRE
- GREv6
- VXLAN
- VXLANv6
- GRE ERSPAN
- IP
- IPv6
- ICMP
- ICMPv6
- ARP
- L2 Packet
- MPLS
- QINQ
- IGMP

然后将该数据包截去前6个字节（目的MAC地址）放入到缓存中，暂且称其为“模板”数据包。

```
logger.info("configure packet buffer")
pktgen_pkt_buffer_table.entry_add(
    target,
    [pktgen_pkt_buffer_table.make_key([gc.KeyTuple('pkt_buffer_offset', buff_offset),
                                          gc.KeyTuple('pkt_buffer_size', (pktlen - 6))]),
    [pktgen_pkt_buffer_table.make_data([gc.DataTuple('buffer', str(p)[6:])])])]
```

为什么要截去前6个字节呢？因为Pkt Gen生成的数据包头部字段恰好是6个字节，数据面将这6个字节和缓存中的“模板”数据包拼接到一起，就生成了一个数据包。

疑问比较大的一点是，这前6个字节是如何生成的呢？因为我们发现控制面并没有做这件事，只是进行了校验的工作。通过后续的抓包分析，我们推断出这6个字节是数据面自动生成的。

校验时前6个字节的生成过程如下：

```
pipe_shift = 4
h '%02x:00:%02x:%02x:%02x:%02x' % ((pipe_id << pipe_shift) | app_id,
down_port >> 8,
down_port & 0xFF,
packet_id >> 8,
packet_id & 0xFF)
```

## 运行程序，抓包分析

tna\_pktgen给出了两种触发模式的测试程序：

- TimerPktgenTest (One - time timer)
- PortDownPktgenTest (Port down)

在PTF测试平台上运行测试（“模板为eth数据包”），抓取数据包：

在TimerPktgenTest测试中，自定义batch\_count(4)和packet\_count(2)，因此生成8个数据包。可以看到，数据包大小都是一致的，只是在目的MAC（即前6个字节）上有区别。进一步地，我们就可以发现目的MAC的规律，恰好是与batch\_count（0-3）、packet\_count（0-1）紧密相关。

The image shows a Wireshark capture of a file named 'TimerPktgenTest.pcap'. The packet list at the top shows 8 packets, all of type LLDP, with a length of 100 bytes. The destination MAC address for all packets is 00:00:00\_00:00:00. The source MAC address for all packets is OmniDire\_08:09:0a. The info column for all packets indicates 'Invalid Chassis ID TLV'.

The packet details pane for the first packet (Frame 1) shows the following information:

- Encapsulation type: Ethernet (1)
- Arrival Time: Dec 15, 2020 16:20:21.732353000 中国标准时间
- [Time shift for this packet: 0.000000000 seconds]
- Epoch Time: 1608020421.732353000 seconds
- [Time delta from previous captured frame: 0.000000000 seconds]
- [Time delta from previous displayed frame: 0.000000000 seconds]
- [Time since reference or first frame: 0.000000000 seconds]
- Frame Number: 1
- Frame Length: 100 bytes (800 bits)
- Capture Length: 100 bytes (800 bits)
- [Frame is marked: False]
- [Frame is ignored: False]
- [Protocols in frame: eth:ethertype:lldp]
- [Coloring Rule Name: Broadcast]
- [Coloring Rule String: eth[0] & 1]

The packet details pane for the Ethernet II layer shows the following information:

- Destination: 00:00:00\_00:00:00 (01:00:00:00:00:00)
- Address: 00:00:00\_00:00:00 (01:00:00:00:00:00)
- .... ..0. .... = LG bit: Globally unique address (factory default)
- .... ..1. .... = IG bit: Group address (multicast/broadcast)
- Source: OmniDire\_08:09:0a (00:06:07:08:09:0a)
- Address: OmniDire\_08:09:0a (00:06:07:08:09:0a)
- .... ..0. .... = LG bit: Globally unique address (factory default)
- .... ..0. .... = IG bit: Individual address (unicast)
- Type: 802.1 Link Layer Discovery Protocol (LLDP) (0x88cc)

The packet details pane for the Link Layer Discovery Protocol layer shows the following information:

- Invalid Chassis ID (0x18), expected (0x01)
- [Expert Info (Warning/Malformed): Invalid Chassis ID (0x18), expected (0x01)]
- [Invalid Chassis ID (0x18), expected (0x01)]
- [Severity level: Warning]
- [Group: Malformed]

The packet bytes pane shows the raw data of the first packet, which is a valid LLDP packet with a valid Chassis ID TLV.

在PortDownPktgenTest测试中，扫描可用的端口数量，自定义packet\_count(2)/端口，扫描到可用端口数量为17（port1-16，port64），因此生成34个数据包。

```
enable pktgen port
configure pktgen application
configure packet buffer
enable pktgen
Clear port down
Take port 1 down
()
Take port 2 down
()
Take port 3 down
()
Take port 4 down
()
Take port 5 down
()
Take port 6 down
()
Take port 7 down
()
Take port 8 down
()
Take port 9 down
()
Take port 10 down
()
Take port 11 down
()
Take port 12 down
()
Take port 13 down
()
Take port 14 down
()
Take port 15 down
()
Take port 16 down
()
Take port 64 down
()
Bring port 1 up
Bring port 2 up
Bring port 3 up
Bring port 4 up
Bring port 5 up
Bring port 6 up
Bring port 7 up
Bring port 8 up
Bring port 9 up
Bring port 10 up
Bring port 11 up
Bring port 12 up
Bring port 13 up
Bring port 14 up
Bring port 15 up
Bring port 16 up
Bring port 64 up
disable pktgen
disable port for pktgen
ok
-----
Ran 1 test in 1.743s
OK
- bf-sde-9.1.1 sudo ./run_p4_tests.sh -p tna_pktgen -t ~/bf-sde-9.1.1/pkgsrc/p4-examples/p4_16_programs/tna_pktgen -s test.TimerPktgenTest
```

PortDownPktgenTest.pcap

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(T) 帮助(H)

应用显示过滤器 ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	OmniDire_08:09:0a	02:00:00:01:00:00	LLDP	100	Invalid Chassis ID TLV
2	0.016757	OmniDire_08:09:0a	02:00:00:01:00:01	LLDP	100	Invalid Chassis ID TLV
3	0.033528	OmniDire_08:09:0a	02:00:00:02:00:00	LLDP	100	Invalid Chassis ID TLV
4	0.050265	OmniDire_08:09:0a	02:00:00:02:00:01	LLDP	100	Invalid Chassis ID TLV
5	0.065843	OmniDire_08:09:0a	02:00:00:03:00:00	LLDP	100	Invalid Chassis ID TLV
6	0.082226	OmniDire_08:09:0a	02:00:00:03:00:01	LLDP	100	Invalid Chassis ID TLV
7	0.098643	OmniDire_08:09:0a	02:00:00:04:00:00	LLDP	100	Invalid Chassis ID TLV
8	0.114886	OmniDire_08:09:0a	02:00:00:04:00:01	LLDP	100	Invalid Chassis ID TLV
9	0.131634	OmniDire_08:09:0a	02:00:00:05:00:00	LLDP	100	Invalid Chassis ID TLV
10	0.148102	OmniDire_08:09:0a	02:00:00:05:00:01	LLDP	100	Invalid Chassis ID TLV
11	0.164375	OmniDire_08:09:0a	02:00:00:06:00:00	LLDP	100	Invalid Chassis ID TLV
12	0.180607	OmniDire_08:09:0a	02:00:00:06:00:01	LLDP	100	Invalid Chassis ID TLV
13	0.196795	OmniDire_08:09:0a	02:00:00:07:00:00	LLDP	100	Invalid Chassis ID TLV
14	0.212629	OmniDire_08:09:0a	02:00:00:07:00:01	LLDP	100	Invalid Chassis ID TLV
15	0.229381	OmniDire_08:09:0a	02:00:00:08:00:00	LLDP	100	Invalid Chassis ID TLV
16	0.246404	OmniDire_08:09:0a	02:00:00:08:00:01	LLDP	100	Invalid Chassis ID TLV
17	0.263011	OmniDire_08:09:0a	02:00:00:09:00:00	LLDP	100	Invalid Chassis ID TLV
18	0.279402	OmniDire_08:09:0a	02:00:00:09:00:01	LLDP	100	Invalid Chassis ID TLV
19	0.295448	OmniDire_08:09:0a	02:00:00:0a:00:00	LLDP	100	Invalid Chassis ID TLV
20	0.311883	OmniDire_08:09:0a	02:00:00:0a:00:01	LLDP	100	Invalid Chassis ID TLV
21	0.328454	OmniDire_08:09:0a	02:00:00:0b:00:00	LLDP	100	Invalid Chassis ID TLV
22	0.344566	OmniDire_08:09:0a	02:00:00:0b:00:01	LLDP	100	Invalid Chassis ID TLV
23	0.360717	OmniDire_08:09:0a	02:00:00:0c:00:00	LLDP	100	Invalid Chassis ID TLV
24	0.376624	OmniDire_08:09:0a	02:00:00:0c:00:01	LLDP	100	Invalid Chassis ID TLV
25	0.392792	OmniDire_08:09:0a	02:00:00:0d:00:00	LLDP	100	Invalid Chassis ID TLV
26	0.409156	OmniDire_08:09:0a	02:00:00:0d:00:01	LLDP	100	Invalid Chassis ID TLV
27	0.425570	OmniDire_08:09:0a	02:00:00:0e:00:00	LLDP	100	Invalid Chassis ID TLV
28	0.442120	OmniDire_08:09:0a	02:00:00:0e:00:01	LLDP	100	Invalid Chassis ID TLV
29	0.458701	OmniDire_08:09:0a	02:00:00:0f:00:00	LLDP	100	Invalid Chassis ID TLV
30	0.475033	OmniDire_08:09:0a	02:00:00:0f:00:01	LLDP	100	Invalid Chassis ID TLV
31	0.491019	OmniDire_08:09:0a	02:00:00:10:00:00	LLDP	100	Invalid Chassis ID TLV
32	0.507602	OmniDire_08:09:0a	02:00:00:10:00:01	LLDP	100	Invalid Chassis ID TLV
33	0.524733	OmniDire_08:09:0a	02:00:00:40:00:00	LLDP	100	Invalid Chassis ID TLV
34	0.524986	OmniDire_08:09:0a	02:00:00:40:00:01	LLDP	100	Invalid Chassis ID TLV

< >

> Frame 1: 100 bytes on wire (800 bits), 100 bytes captured (800 bits)

> Ethernet II, Src: OmniDire\_08:09:0a (00:06:07:08:09:0a), Dst: 02:00:00:01:00:00 (02:00:00:01:00:00)

> Link Layer Discovery Protocol

0000 02 00 00 01 00 00 06 07 08 09 0a 88 cc 30 30 .....00

0010 30 30 30 30 30 30 30 30 30 30 30 30 30 30 00000000 00000000

0020 30 30 30 30 30 30 30 30 30 30 30 30 30 30 00000000 00000000

0030 30 30 30 30 30 30 30 30 30 30 30 30 30 30 00000000 00000000

0040 30 30 30 30 30 30 30 30 30 30 30 30 30 30 00000000 00000000

0050 30 30 30 30 30 30 30 30 30 30 30 30 30 30 00000000 00000000

0060 30 30 30 30 0000

PortDownPktgenTest.pcap 分组: 34 · 已显示: 34 (100.0%) 配置: Default

后续，我又尝试修改“模板”数据包为TCP的SYN包，同样抓取到了预想数目的数据包：

tcp\_test.pcap

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(I) 帮助(H)

应用显示过滤器 ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.1	192.168.0.2	TCP	100	1234 → 80 [SYN] Seq=0 Win=8192 Len=46
2	0.016392	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=
3	0.032529	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=
4	0.049068	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=
5	0.065303	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=
6	0.081522	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=
7	0.097391	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=
8	0.097640	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=

< >

> Frame 1: 100 bytes on wire (800 bits), 100 bytes captured (800 bits)

> Ethernet II, Src: OmniDir\_08:09:0a (00:06:07:08:09:0a), Dst: 00:00:00\_00:00:00 (01:00:00:00:00:00)

> Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.2

> Transmission Control Protocol, Src Port: 1234, Dst Port: 80, Seq: 0, Len: 46

0000 01 00 00 00 00 00 06 07 08 09 0a 08 00 45 00 .....E-

0010 00 56 00 01 00 00 40 06 f9 4d c0 a8 00 01 c0 a8 .V...@.M...

0020 00 02 04 d2 00 50 00 00 00 00 00 00 00 50 02 .....P.....P.

0030 20 00 0d 2c 00 00 00 01 02 03 04 05 06 07 08 09 .,.....

0040 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 .....

0050 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 ..... ! " \$ % & ' ( )

0060 2a 2b 2c 2d ..... \* , -

tcp\_test.pcap

分组: 8 · 已显示: 8 (100.0%)

配置: Default



tcp\_test1.pcap

文件(F) 编辑(E) 视图(V) 跳转(G) 捕获(C) 分析(A) 统计(S) 电话(Y) 无线(W) 工具(I) 帮助(H)

应用显示过滤器

<Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.1	192.168.0.2	TCP	100	1234 → 80 [SYN] Seq=0 Win=8192 Len=46
2	0.019805	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
3	0.039555	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
4	0.059434	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
5	0.077939	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
6	0.102820	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
7	0.122266	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
8	0.141423	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
9	0.160325	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
10	0.179713	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
11	0.198629	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
12	0.218474	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
13	0.237598	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
14	0.256847	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
15	0.275547	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
16	0.299258	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
17	0.319306	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
18	0.338448	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
19	0.357574	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
20	0.375844	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
21	0.394120	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
22	0.412135	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
23	0.429992	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
24	0.448499	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
25	0.466882	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
26	0.484712	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
27	0.507970	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
28	0.526222	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
29	0.544360	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
30	0.562531	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
31	0.580621	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
32	0.599076	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
33	0.617889	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46
34	0.618193	192.168.0.1	192.168.0.2	TCP	100	[TCP Retransmission] 1234 → 80 [SYN] Seq=0 Win=8192 Len=46

<

>

> Frame 1: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface 0

> Ethernet II, Src: OmniDir\_08:09:0a:00:06:07, Dst: 02:00:00:01:00:00 (02:00:00:01:00:00)

> Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.2

> Transmission Control Protocol, Src Port: 1234, Dst Port: 80, Seq: 0, Len: 46

0000 02 00 00 01 00 00 06 07 08 09 0a 08 00 45 00 .....E-

0010 00 56 00 01 00 00 40 06 f9 4d c0 a8 00 01 c0 a8 .V...@..M.....

0020 00 02 04 d2 00 50 00 00 00 00 00 00 00 50 02 ....P.....P.

0030 20 00 0d 2c 00 00 00 01 02 03 04 05 06 07 08 09 .,.....

0040 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 .....!

0050 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 .....! "%&'()

0060 2a 2b 2c 2d .....\*+,-

tcp\_test1.pcap

分组: 34 · 已显示: 34 (100.0%) 配置: Default