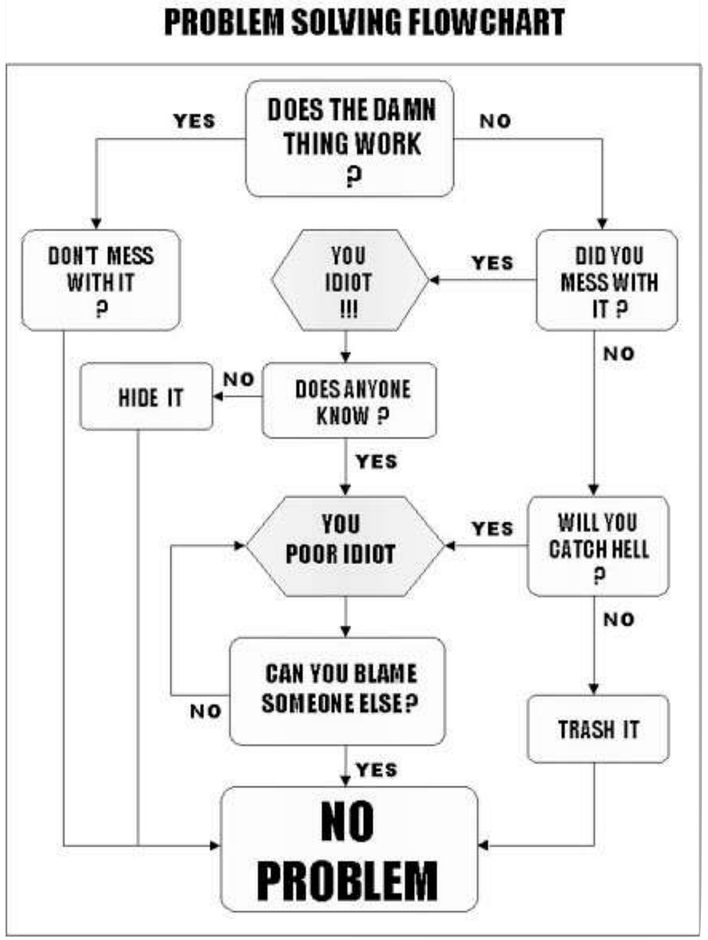
SharkFest '17 US

Digging Deep

Exploring real-life case studies

June 20th, 2017

Capture files are available at:
http://www.SYN-bit.nl/files/sf17us.zip



Sake Blok

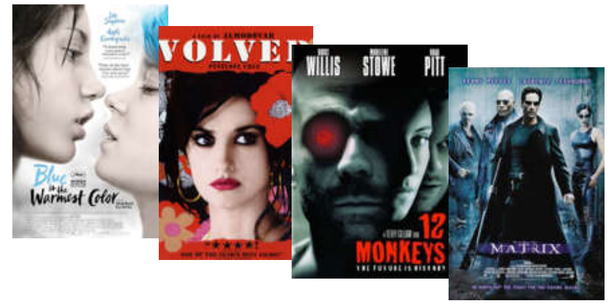
sake.blok@SYN-bit.nl

Relational Therapist for Computer Systems | SYN-bit

About me…

EURONET INTERNET

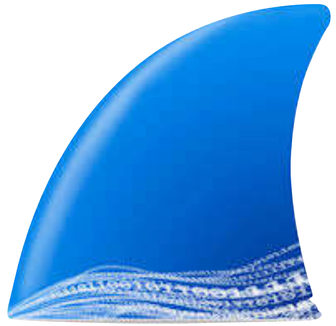
techno inside



ABN·AMRO

SYN-bit

deep traffic analysis



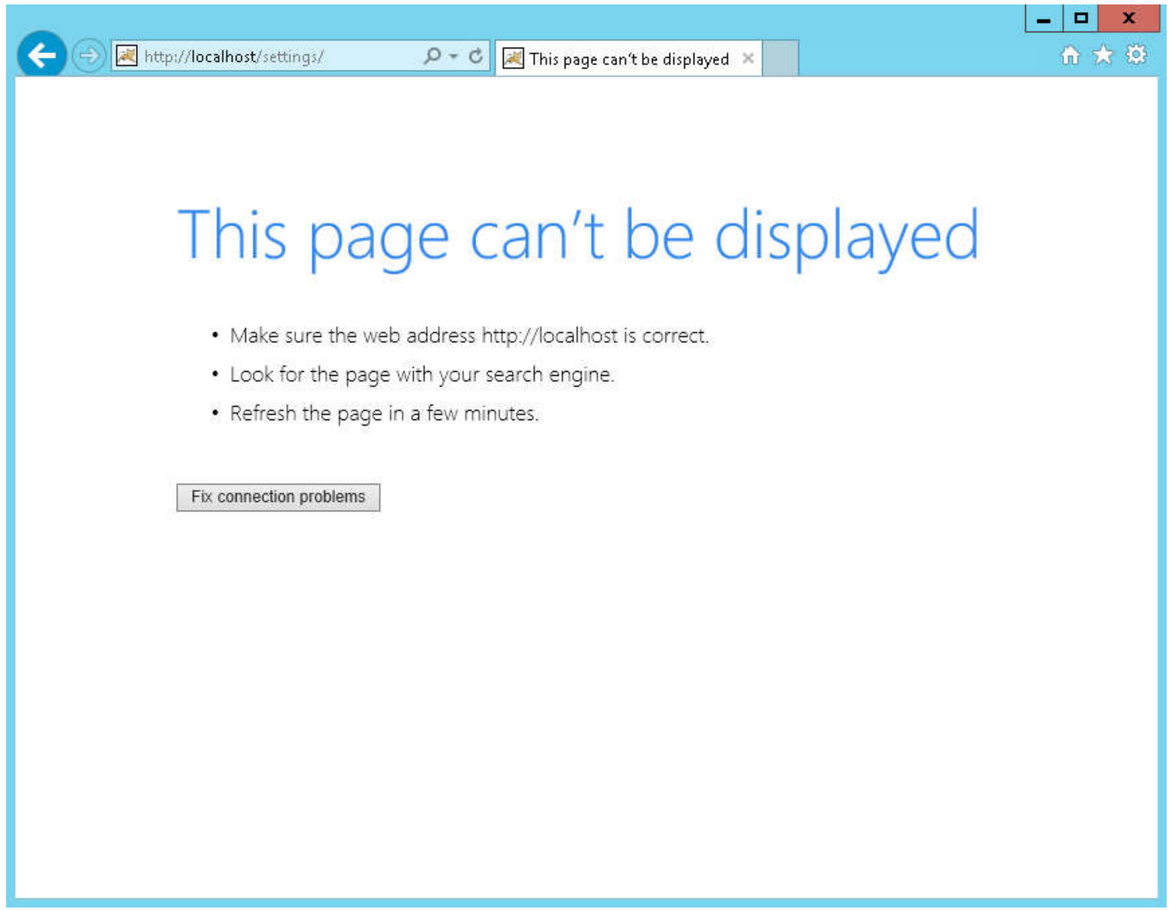
ion ip

f5

Haarlem

case00

- Customer calls on sunday night, their whole web infrastructure is failing
- I maintain their F5 loadbalancers and log in to make a trace



case00 (solved)

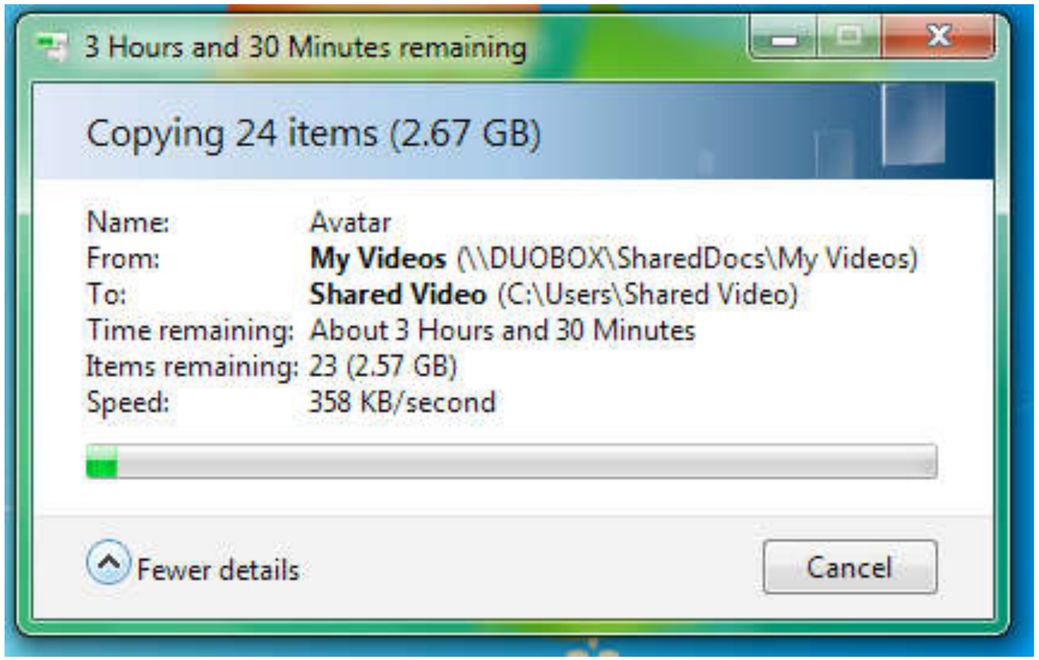
- Identified a SYN flood attack
- Double checked with GeoIP
- Created the packet filter:

```
- filter {
      host 31.151.1.21 and
      tcp[13]=2 and
      tcp[12]&0xf0=0x50
  }
```

- Things got back to normal

case01

- Bad performance on Gbit WAN connection
- 5,8 ms network RTT
- File transfers of 15-25 MB/s
- Installed a sniffer to capture file transfer
- used iperf for testing



Max bandwidth in one TCP session

Print Area

		TCP window size							
		64 KB	128 KB	256 KB	512 KB	1 MB	2 MB	4 MB	8 MB
RTT	100 µs	5,5 Gbit/s	11 Gbit/s	22 Gbit/s	44 Gbit/s	87 Gbit/s	174 Gbit/s	349 Gbit/s	698 Gbit/s
	200 µs	2,7 Gbit/s	5,5 Gbit/s	11 Gbit/s	22 Gbit/s	44 Gbit/s	87 Gbit/s	174 Gbit/s	349 Gbit/s
	500 µs	1,1 Gbit/s	2,2 Gbit/s	4,4 Gbit/s	8,7 Gbit/s	17 Gbit/s	35 Gbit/s	70 Gbit/s	140 Gbit/s
	1 ms	545 Mbit/s	1,1 Gbit/s	2,2 Gbit/s	4,4 Gbit/s	8,7 Gbit/s	17 Gbit/s	35 Gbit/s	70 Gbit/s
	2 ms	273 Mbit/s	545 Mbit/s	1,1 Gbit/s	2,2 Gbit/s	4,4 Gbit/s	8,7 Gbit/s	17 Gbit/s	35 Gbit/s
	5 ms	109 Mbit/s	218 Mbit/s	436 Mbit/s	872 Mbit/s	1,7 Gbit/s	3,5 Gbit/s	7,0 Gbit/s	14 Gbit/s
	10 ms	55 Mbit/s	109 Mbit/s	218 Mbit/s	436 Mbit/s	872 Mbit/s	1,7 Gbit/s	3,5 Gbit/s	7,0 Gbit/s
	20 ms	27 Mbit/s	55 Mbit/s	109 Mbit/s	218 Mbit/s	436 Mbit/s	872 Mbit/s	1,7 Gbit/s	3,5 Gbit/s
	50 ms	11 Mbit/s	22 Mbit/s	44 Mbit/s	87 Mbit/s	174 Mbit/s	349 Mbit/s	698 Mbit/s	1,4 Gbit/s
	100 ms	5,5 Mbit/s	11 Mbit/s	22 Mbit/s	44 Mbit/s	87 Mbit/s	174 Mbit/s	349 Mbit/s	698 Mbit/s

case01 (iperf -w64K)

```
$ iperf3 -c 192.168.0.2 -w64K
Connecting to host 192.168.0.2, port 5201
[  4] local 192.168.0.1 port 41390 connected to 192.168.0.2 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  7.30 MBytes  61.2 Mbits/sec    5   45.2 KBytes
[  4]   1.00-2.00   sec  9.09 MBytes  76.3 Mbits/sec    1   66.5 KBytes
[  4]   2.00-3.00   sec  7.70 MBytes  64.6 Mbits/sec    6   39.6 KBytes
[  4]   3.00-4.00   sec  8.61 MBytes  72.2 Mbits/sec    2   66.5 KBytes
[  4]   4.00-5.00   sec  9.19 MBytes  77.1 Mbits/sec    2   66.5 KBytes
[  4]   5.00-6.00   sec  9.83 MBytes  82.4 Mbits/sec    0   66.5 KBytes
[  4]   6.00-7.00   sec  8.96 MBytes  75.2 Mbits/sec    3   52.3 KBytes
[  4]   7.00-8.00   sec  9.28 MBytes  77.8 Mbits/sec    2   67.9 KBytes
[  4]   8.00-9.00   sec  8.06 MBytes  67.6 Mbits/sec    5   67.9 KBytes
[  4]   9.00-10.00  sec  9.97 MBytes  83.6 Mbits/sec    0   67.9 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  88.0 MBytes  73.8 Mbits/sec   26             sender
[  4]   0.00-10.00  sec  87.9 MBytes  73.7 Mbits/sec                  receiver

iperf Done.
$
```

case01 (iperf -w128K)

```
$ iperf3 -c 192.168.0.2 -w128K
Connecting to host 192.168.0.2, port 5201
[  4] local 192.168.0.1 port 41468 connected to 192.168.0.2 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  15.7 MBytes   132 Mbits/sec    1    106 KBytes
[  4]   1.00-2.00   sec  18.0 MBytes   151 Mbits/sec    1    130 KBytes
[  4]   2.00-3.00   sec  16.1 MBytes   135 Mbits/sec    3    102 KBytes
[  4]   3.00-4.00   sec  16.2 MBytes   136 Mbits/sec    2    130 KBytes
[  4]   4.00-5.00   sec  16.8 MBytes   141 Mbits/sec    2    130 KBytes
[  4]   5.00-6.00   sec  17.7 MBytes   148 Mbits/sec    2    106 KBytes
[  4]   6.00-7.00   sec  15.2 MBytes   128 Mbits/sec    3   96.2 KBytes
[  4]   7.00-8.00   sec  14.3 MBytes   120 Mbits/sec    4    103 KBytes
[  4]   8.00-9.00   sec  12.3 MBytes   103 Mbits/sec    5   70.7 KBytes
[  4]   9.00-10.00  sec  16.5 MBytes   138 Mbits/sec    1    130 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   159 MBytes   133 Mbits/sec   24             sender
[  4]   0.00-10.00  sec   159 MBytes   133 Mbits/sec                  receiver

iperf Done.
$
```

case01 (iperf -w256K)

```
$ iperf3 -c 192.168.0.2 -w256K
Connecting to host 192.168.0.2, port 5201
[  4] local 192.168.0.1 port 41476 connected to 192.168.0.2 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  15.1 MBytes   127 Mbits/sec    4   94.7 KBytes
[  4]   1.00-2.00   sec  17.5 MBytes   147 Mbits/sec    2    123 KBytes
[  4]   2.00-3.00   sec  21.6 MBytes   181 Mbits/sec    2    137 KBytes
[  4]   3.00-4.00   sec  21.5 MBytes   180 Mbits/sec    4   97.6 KBytes
[  4]   4.00-5.00   sec  16.0 MBytes   134 Mbits/sec    3   97.6 KBytes
[  4]   5.00-6.00   sec  17.1 MBytes   143 Mbits/sec    3   87.7 KBytes
[  4]   6.00-7.00   sec  11.2 MBytes  94.4 Mbits/sec    4    102 KBytes
[  4]   7.00-8.00   sec  17.7 MBytes   149 Mbits/sec    3    126 KBytes
[  4]   8.00-9.00   sec  19.8 MBytes   166 Mbits/sec    2    147 KBytes
[  4]   9.00-10.00  sec  19.5 MBytes   163 Mbits/sec    3    115 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   177 MBytes   148 Mbits/sec   30             sender
[  4]   0.00-10.00  sec   177 MBytes   148 Mbits/sec                  receiver

iperf Done.
$
```

case01 (iperf -w1024K)

```
$ iperf3 -c 192.168.0.2 -w1024K
Connecting to host 192.168.0.2, port 5201
[  4] local 192.168.0.1 port 41472 connected to 192.168.0.2 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  8.57 MBytes  71.9 Mbits/sec    3   72.1 KBytes
[  4]   1.00-2.00   sec  18.8 MBytes   158 Mbits/sec    1    140 KBytes
[  4]   2.00-3.00   sec  17.8 MBytes   150 Mbits/sec    4   76.4 KBytes
[  4]   3.00-4.00   sec  20.0 MBytes   167 Mbits/sec    1    136 KBytes
[  4]   4.00-5.00   sec  24.1 MBytes   202 Mbits/sec    1    189 KBytes
[  4]   5.00-6.00   sec  16.5 MBytes   138 Mbits/sec    4    136 KBytes
[  4]   6.00-7.00   sec  18.5 MBytes   155 Mbits/sec    2    153 KBytes
[  4]   7.00-8.00   sec  20.8 MBytes   174 Mbits/sec    2    165 KBytes
[  4]   8.00-9.00   sec  24.2 MBytes   203 Mbits/sec    3    106 KBytes
[  4]   9.00-10.00  sec  15.2 MBytes   127 Mbits/sec    3    119 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec   184 MBytes   155 Mbits/sec   24             sender
[  4]   0.00-10.00  sec   184 MBytes   154 Mbits/sec                  receiver

iperf Done.
$
```

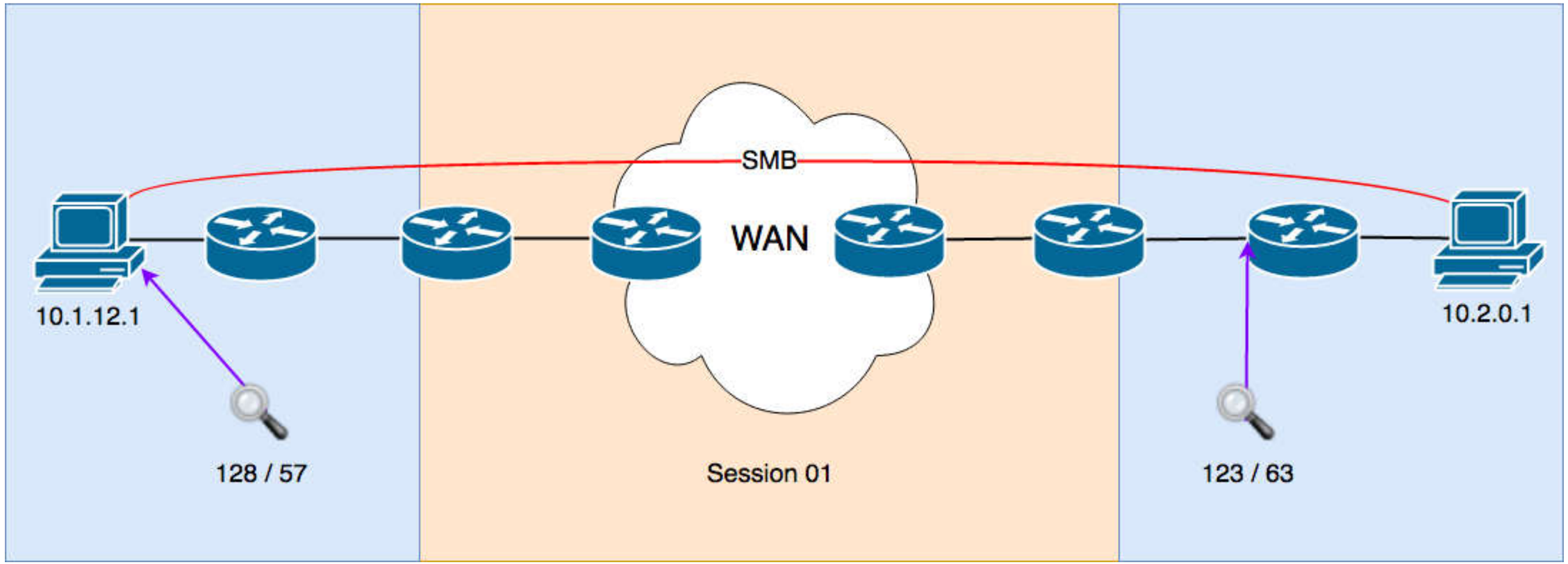
case01 (solved)

- RTT alone not the problem. Window scaling solved this
- Packet loss (CWND limiting) alone not the problem
- Random packet loss is a performance killer, TCP congestion avoidance assumes packet loss to be due to congestion

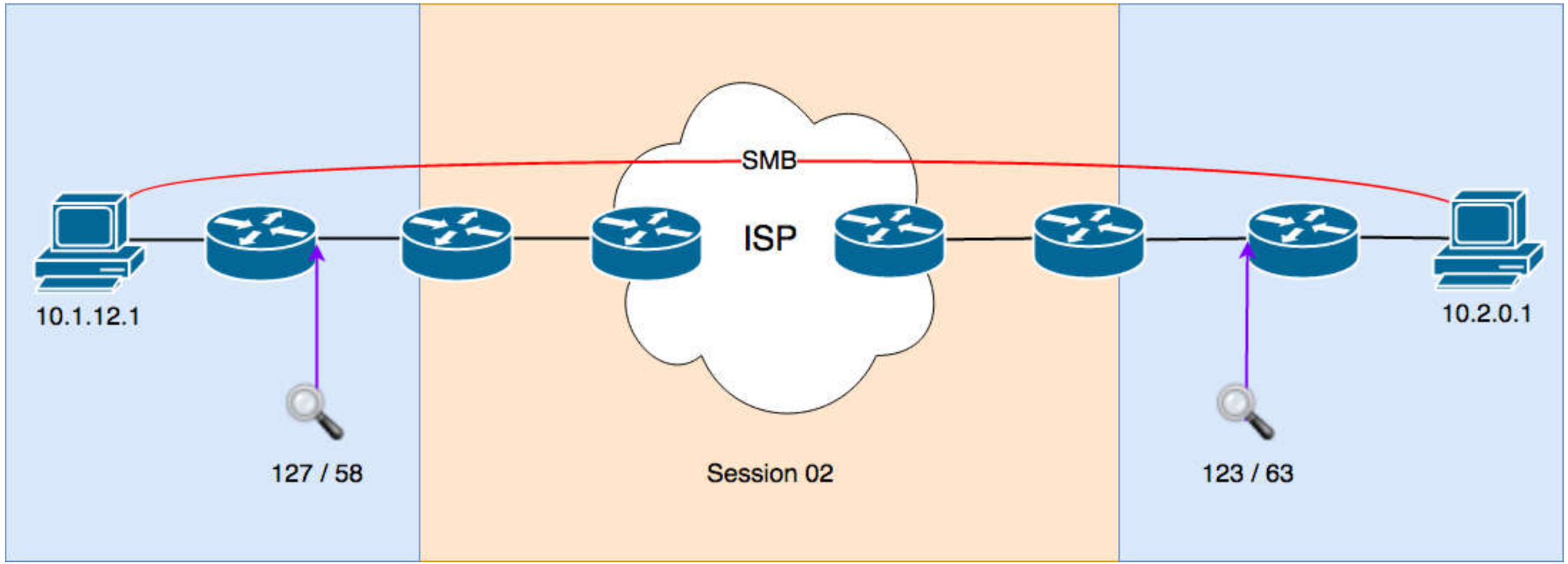
case02

- Customer wants to migrate to another WAN provider
- Customer tried migrating for 2 months, no success and suspects kerberos issue.
- Installed two sniffers and captured the problem

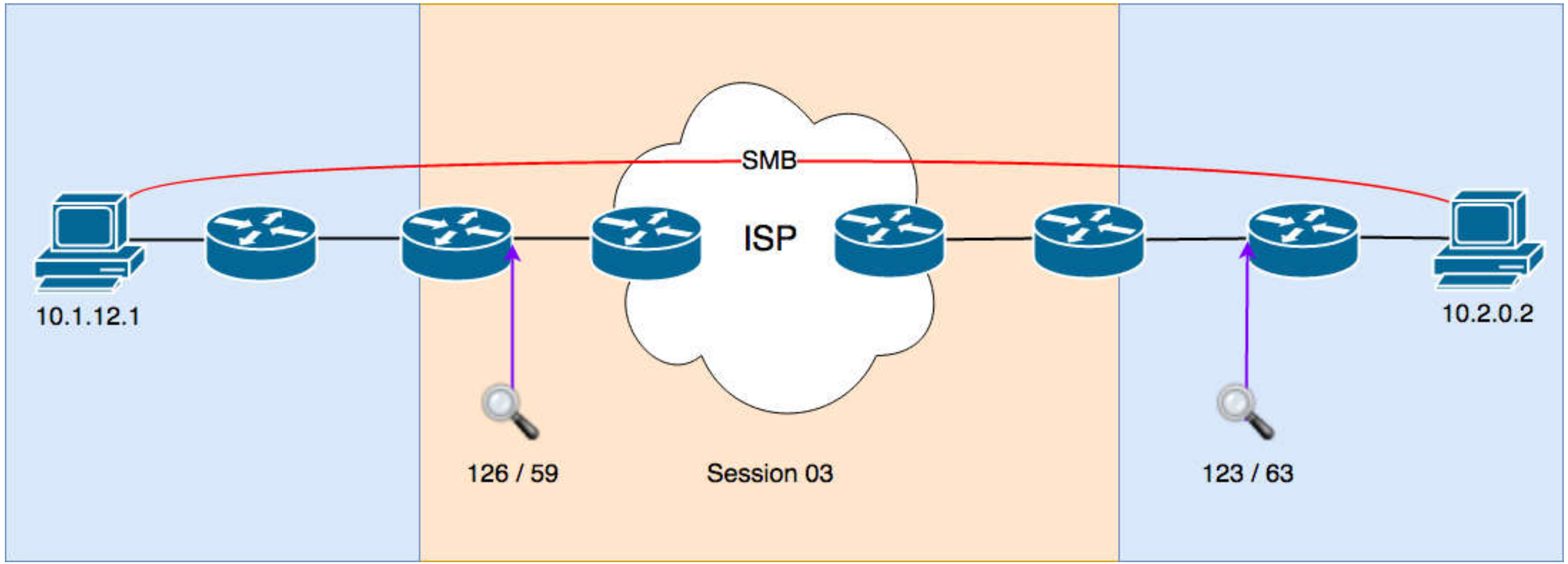
case02 (session 01)



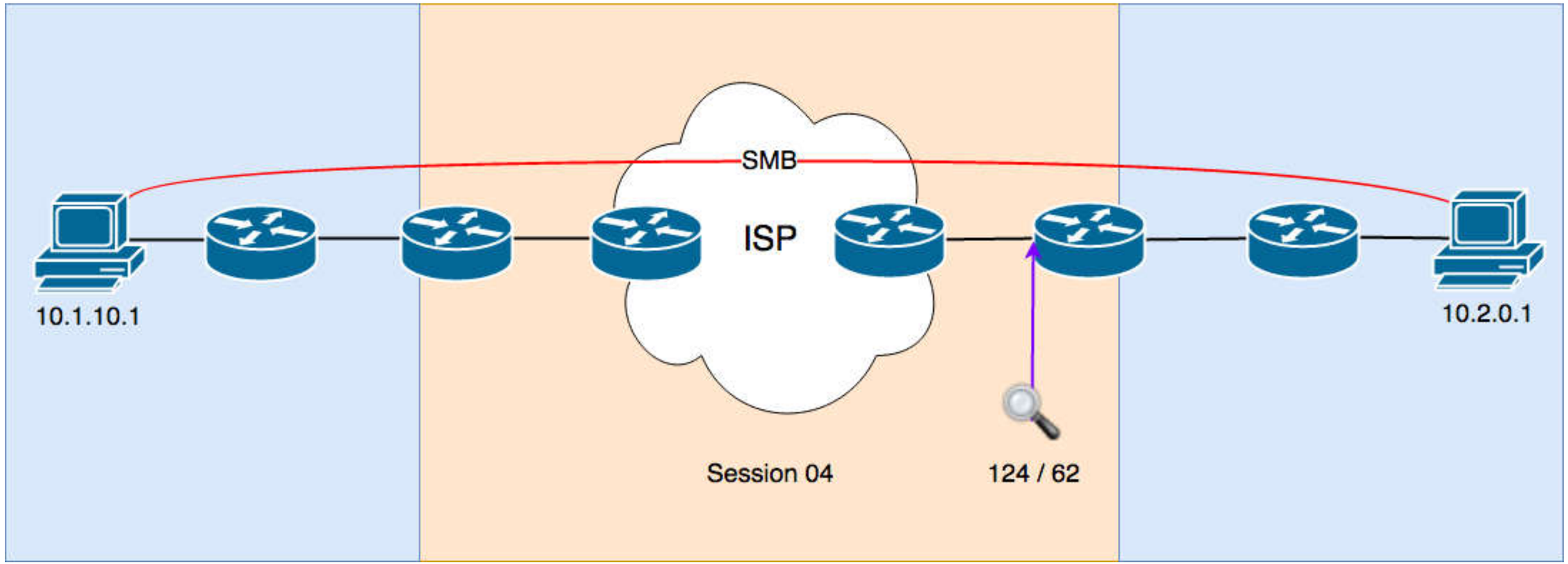
case02 (session 02)



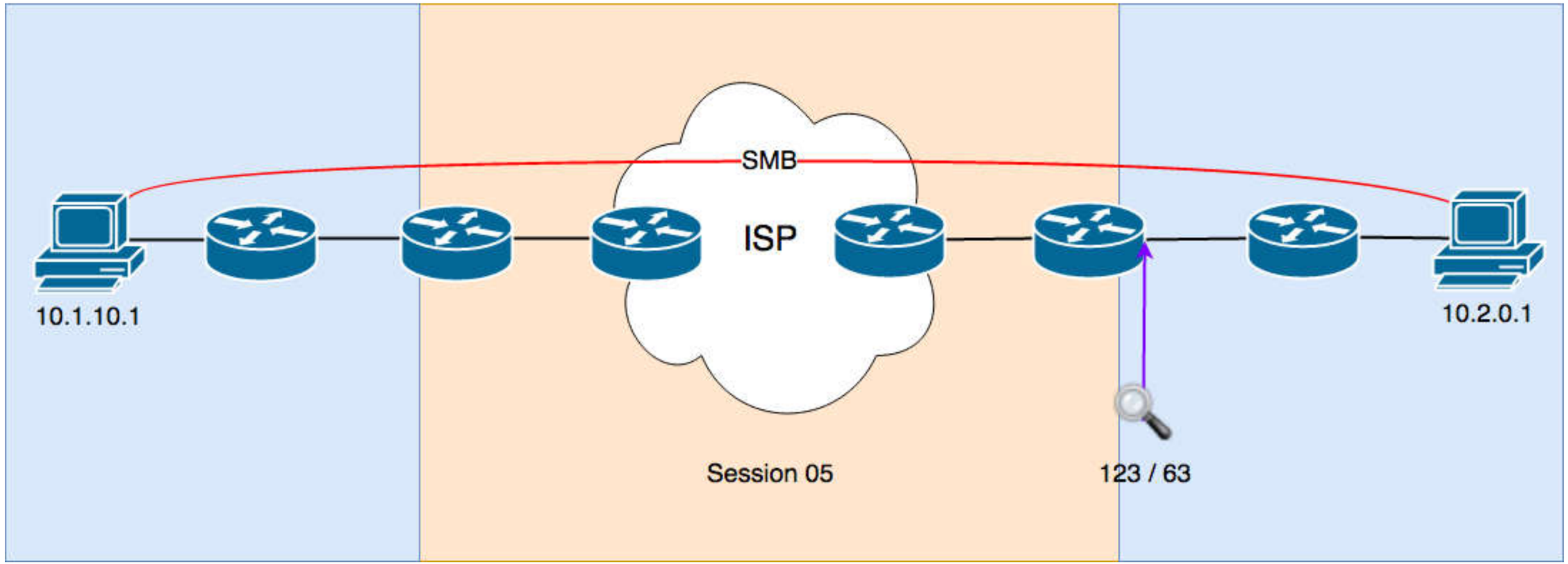
case02 (session 03)



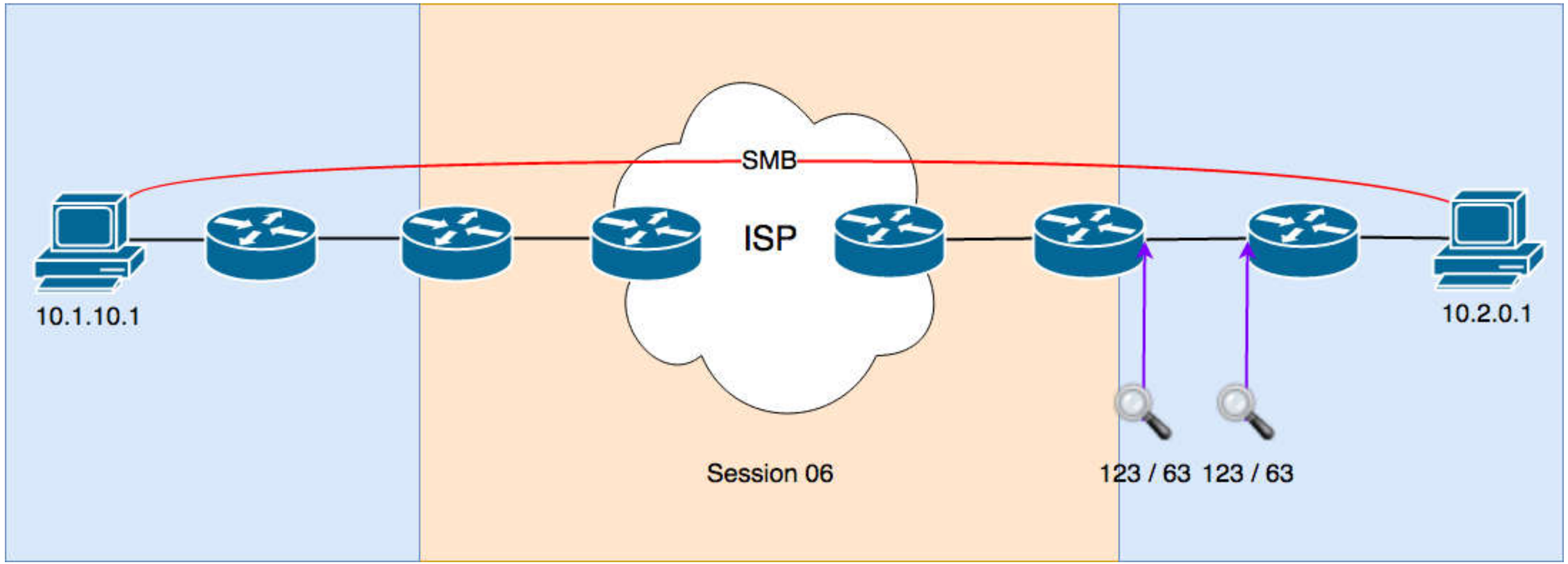
case02 (session 04)



case02 (session 05)



case02 (session 06)



case02 (solved)

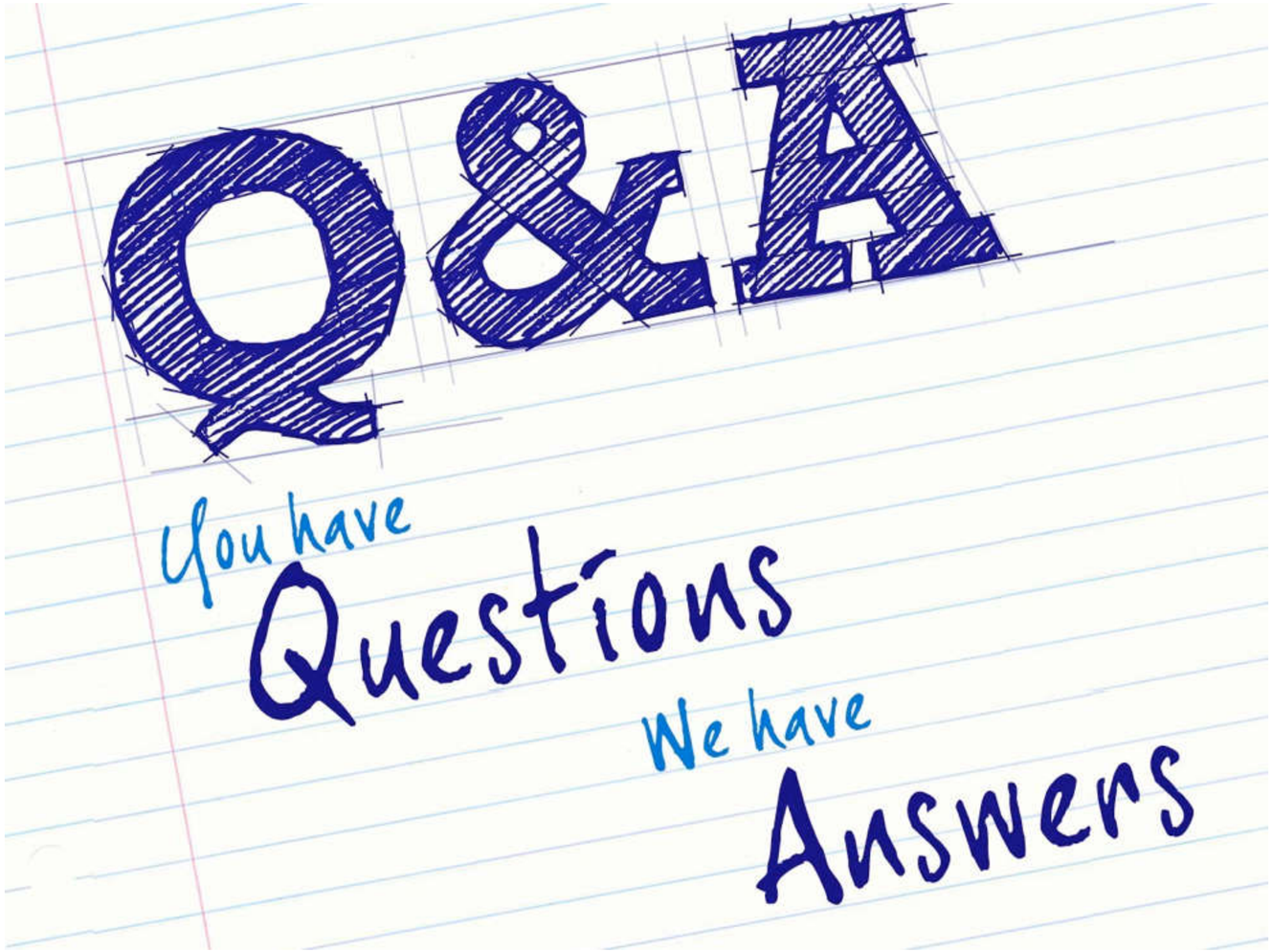
- Only traffic that entered the core-switch vlan tagged had problems
- The switch software had a bug that it did not add padding after stripping the vlan tag
- workaround, moved ISP link to other switch (while waiting for bugfix)

case03

- Customer complains about very slow http responses (~20 sec)
- Multiple remote parties connected over VPN, only one with problems
- Received trace file from the customer

case03 (solved)

- Not random packet loss
- Looking at the ACK numbers will reveal things about the missing packets
- Adjusting MSS values before tunneling solves the issue


Q&A
You have
Questions
We have
Answers

FIN/ACK, ACK, FIN/ACK, ACK

Thank You!

sake.blok@SYN-bit.nl

SYN-bit

deep traffic analysis