

ECN bleaching detection with Pietrasanta traceroute

NOG.HR 5

Luca Sani

Senior R&D Software Engineer @Catchpoint



catchpoint.

Pietrasanta Traceroute

- Based on Dmitry Butskoy [Linux traceroute](#)
- Several enhancements
 - Speedup
 - QUIC traceroute
 - ECN bleaching detection
 - Work in Azure environment
 - TCP “In Session”
 - ... and many more



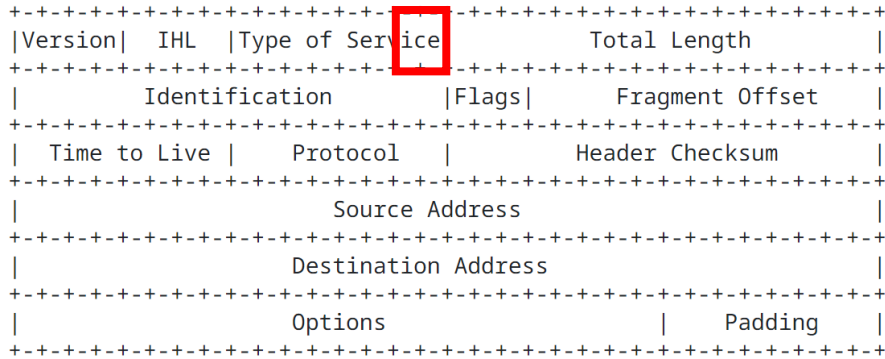
*A noble town since 1841 and a city of art”
(and where our Italian office is located!)*

<https://github.com/catchpoint/Pietrasanta-traceroute>

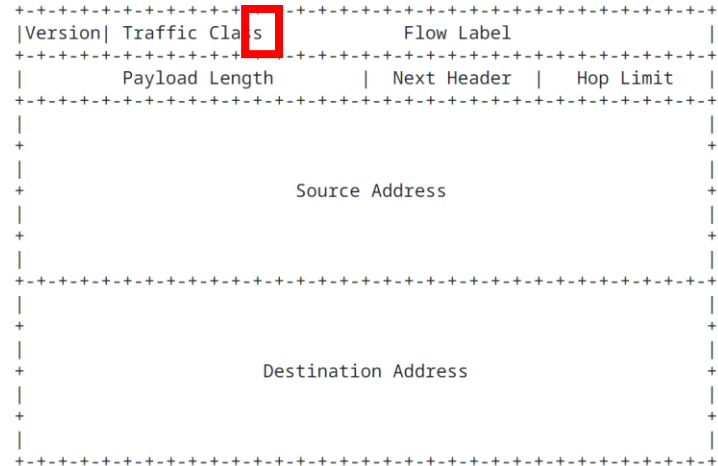
ECN bleaching detection

ECN mechanism

- *The Addition of Explicit Congestion Notification to IP*, [rfc3168](https://www.rfc-editor.org/rfc/rfc3168), 2001
 - Two bits in the IP header (in TOS/Traffic Class)
- The source declares that a packet should be treated with ECN by setting the IP-ECN fields either to 01 or 10



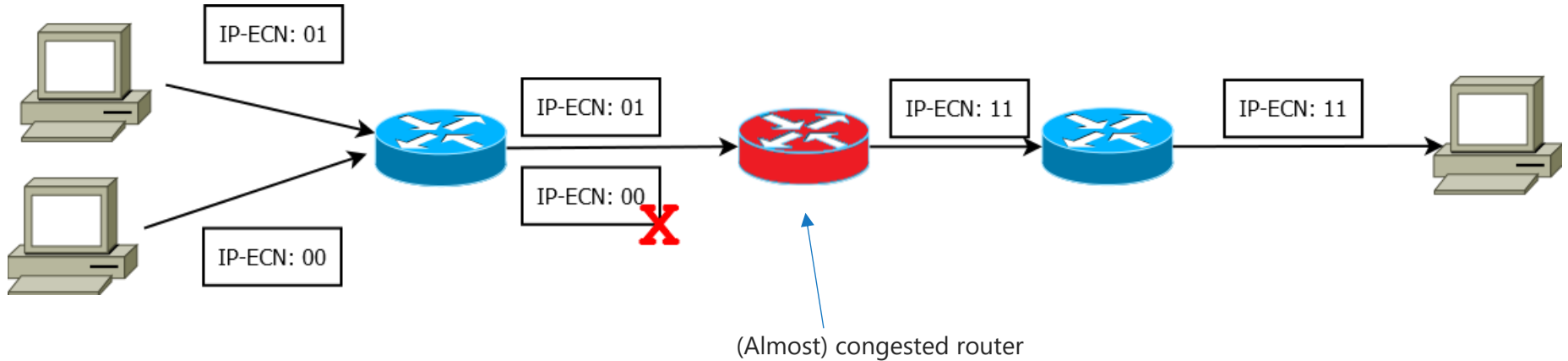
IPv4 header



IPv6 header

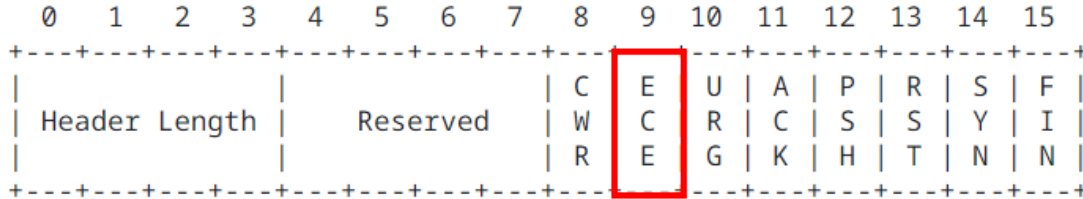
ECN mechanism

- When congestion is (almost) happening, instead of dropping the packet the router sets the IP-ECN fields to 11 (CE - Congestion Experienced)



ECN feedback (transport/application)

- A destination that receives a packet with IP-ECN = CE should report to the source this event
- The source should then adjust the transmission rate
- The ECN feedback is usually implemented at transport/application layer
 - Example: in TCP, the reception of a packet with IP-ECN = CE can be reported using a dedicated TCP flag (ECE – ECN-Echo)



ECN and L4S

- Recently, ECN mechanism got renewed attention due to L4S (Low Latency, Low Loss, and Scalable Throughput – [rfc9330](#), [rfc9331](#), [rfc9332](#) - 2023)
- L4S requires an ECN feedback *more accurate* wrt the “classic” 2001 version
 - “More accurate” = Feedback contains exact counters of IP-ECN values received

L4S

Nokia Bell Labs pioneers L4S, the crucial enabler for large-scale deployments of real-time applications

WWDC23

Reduce network delays with L4S

Shawn Zhang, Internet Technologies

Last-Mile: L4S for Cloud Streaming

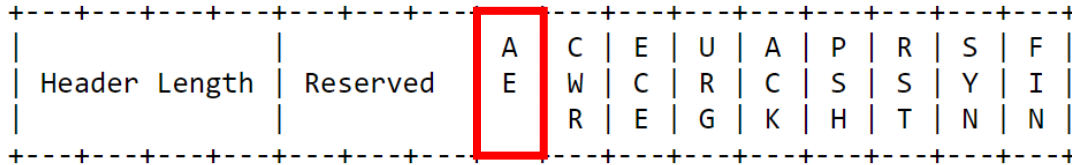
→ Problem 2: Handling impairments in user's network (bufferbloats, packet loss ...)

- L4S [RFC9332] addresses bufferbloats by allowing sender to react faster to queue build-up vs black-box E2E queue build-up estimation
- Use of L4S requires a compliant TX/RX and network (marking, CE feedback, on-path AQM, and new CC)
- CloudXR 4.0 SDK has initial L4S CC support
- PoC L4S support in GeForce NOW in evaluation

The diagram illustrates the L4S mechanism for cloud streaming. It shows a 'Streamer' sending packets to a 'Client' through a 'Congested Network'. The network consists of a 'Downstream AG' and a 'Client' (GIMTS). The Downstream AG has a 'Queue Assignment' and a 'Queue' (E2E). The Client has a 'Queue Assignment' and a 'Queue' (E2E). The network is marked with 'ECT(1)' and 'Other'. The Client sends 'Feedback confirming % CE' back to the Streamer. The network also shows 'Packets Marked ECT(1)' and 'Packets Remarked CE'.

More accurate ECN feedback

- TCP: *More Accurate Explicit Congestion Notification (AccECN) Feedback in TCP* (still a [draft](#))
 - Additional flag into the TCP header (Accurate ECN) + TCP options to carry the counters



- QUIC: Supported natively via [ECN counters](#) in the ACK frame ([rfc9000](#))

```
ECN Counts {  
    ECT0 Count (i),  
    ECT1 Count (i),  
    ECN-CE Count (i),  
}
```


ECN bleaching detection

- Intermediate hops can bleach/alter the value of ECN into the IP header
- See for example: *The Benefits of Using Explicit Congestion Notification (ECN)* – [rfc8087](#), 2017:

Cases have been noted where an endpoint sends a packet with a non-zero ECN mark, but the packet is received by the remote endpoint with a zero ECN codepoint [TR15]. This could be a result of a policy that erases or "bleaches" the ECN codepoint values at a network edge (resetting the codepoint to zero). Bleaching may occur for various reasons (including normalizing packets to hide which equipment supports ECN). This policy prevents use of ECN by applications.

ECN bleaching detection with Pietrasanta Traceroute

- With Pietrasanta traceroute we can send probes with IP-ECN values different from zero and check hop by hop what was the IP-ECN value of the probe ***when it expired***
 - Detect bleaching, but also congestion and any kind of alteration
- We can also check whether the transport (TCP/QUIC) layer at **destination** supports more accurate ECN feedbacks. ECN may not be supported at transport level because:
 - TCP stack need to be patched to support AccECN
 - Not all QUIC implementations report ECN counters into the ACK frame

Report ECN hop by hop - Example

Probe sent

```
Internet Protocol Version 6, Src: 2a12:d8c1:350:a:80db:6ed2:9b67:a9c4, Dst: 2001:67c:27e4::52
  0110 .... = Version: 6
  > .... 0000 0001 .... = Traffic Class: 0x01 (DSCP: CS0, ECN: ECT(1))
  .... 0110 1100 1000 1110 1101 = Flow Label: 0xc68ed
  Payload Length: 40
  Next Header: UDP (17)
  Hop Limit: 3
  Source Address: 2a12:d8c1:350:a:80db:6ed2:9b67:a9c4
  Destination Address: 2001:67c:27e4::52
User Datagram Protocol, Src Port: 56736, Dst Port: 33434
  Source Port: 56736
  Destination Port: 33434
  Length: 40
  Checksum: 0x20a3 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 14307]
  > [Timestamps]
  UDP payload (32 bytes)
Data (32 bytes)
```

ICMP TTL Exceeded

```
Internet Protocol Version 6, Src: 2a12:d8c0:101f:2::1, Dst: 2a12:d8c1:350:a:80db:6ed2:9b67:a9c4
  0110 .... = Version: 6
  > .... 0000 0000 .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
  .... 0000 1100 0110 0011 1001 = Flow Label: 0x0c639
  Payload Length: 88
  Next Header: ICMPv6 (58)
  Hop Limit: 62
  Source Address: 2a12:d8c0:101f:2::1
  Destination Address: 2a12:d8c1:350:a:80db:6ed2:9b67:a9c4
Internet Control Message Protocol v6
  Type: Time Exceeded (3)
  Code: 0 (hop limit exceeded in transit)
  Checksum: 0x747c [correct]
  [Checksum Status: Good]
  Reserved: 00000000
Internet Protocol Version 6, Src: 2a12:d8c1:350:a:80db:6ed2:9b67:a9c4, Dst: 2001:67c:27e4::52
  0110 .... = Version: 6
  > .... 0000 0001 .... = Traffic Class: 0x01 (DSCP: CS0, ECN: ECT(1))
  .... 0110 1100 1000 1110 1101 = Flow Label: 0xc68ed
  Payload Length: 40
  Next Header: UDP (17)
  Hop Limit: 1
  Source Address: 2a12:d8c1:350:a:80db:6ed2:9b67:a9c4
  Destination Address: 2001:67c:27e4::52
User Datagram Protocol, Src Port: 56736, Dst Port: 33434
  Source Port: 56736
  Destination Port: 33434
  Length: 40
  Checksum: 0x20a3 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 14307]
  UDP payload (32 bytes)
Data (32 bytes)
```

ECN detection: Some examples

```
[ bash ]$ sudo ./traceroute -nT -q 1 --ecn=1 -0 acc-ecn,info 95.228.44.181
traceroute to 95.228.44.181(95.228.44.181), 30 hops max, 60 byte packets, overall
timeout not set
 1 172.21.82.1 <TOS:1,DSCP:0,ECN:1> 0.234 ms
 2 64.79.149.27 <TOS:1,DSCP:0,ECN:1> 1.374 ms
 3 64.79.139.17 <TOS:1,DSCP:0,ECN:1> 1.297 ms
 4 66.209.72.25 <TOS:1,DSCP:0,ECN:1> 1.358 ms
 5 *
 6 *
 7 4.68.39.58 <TOS:1,DSCP:0,ECN:1> 6.609 ms
 8 195.22.195.123 <TOS:1,DSCP:0,ECN:1> 160.604 ms
 9 195.22.205.117 <TOS:1,DSCP:0,ECN:1> 173.535 ms
10 *
11 *
12 *
13 *
14 *
15 95.228.44.181 <TOS:1,DSCP:0,ECN:1> 170.007 ms
16 95.228.44.181 <syn,ack,ece,cwr> 172.391 ms
    Timedout: false
    Duration: 1713.448 ms
    DestinationReached: true
```

No bleaching, destination supports
AccECN over TCP

Bleaching happened

```
[ bash ]$ sudo ./traceroute -nT -q 1 --ecn=1 -0 acc-ecn,info 81.236.63.162
traceroute to 81.236.63.162(81.236.63.162), 30 hops max, 60 byte packets, overall
timeout not set
 1 172.21.82.1 <TOS:1,DSCP:0,ECN:1> 0.233 ms
 2 64.79.149.27 <TOS:1,DSCP:0,ECN:1> 1.270 ms
 3 64.79.139.17 <TOS:1,DSCP:0,ECN:1> 1.254 ms
 4 66.209.72.25 <TOS:1,DSCP:0,ECN:1> 1.271 ms
 5 66.209.64.124 <TOS:1,DSCP:0,ECN:1> 1.115 ms
 6 62.115.32.150 <TOS:1,DSCP:0,ECN:1> 1.052 ms
 7 62.115.132.119 <TOS:1,DSCP:0,ECN:1> 1.875 ms
 8 62.115.135.190 <TOS:1,DSCP:0,ECN:1> 6.789 ms
 9 62.115.137.38 <TOS:1,DSCP:0,ECN:1> 64.044 ms
10 62.115.136.200 <TOS:1,DSCP:0,ECN:1> 69.195 ms
11 80.91.254.90 <TOS:1,DSCP:0,ECN:1> 145.761 ms
12 62.115.139.172 <TOS:1,DSCP:0,ECN:1> 155.524 ms
13 62.115.140.217 <TOS:0,DSCP:0,ECN:0> 150.248 ms
14 62.115.35.117 <TOS:0,DSCP:0,ECN:0> 150.434 ms
15 81.228.89.186 <TOS:0,DSCP:0,ECN:0> 150.790 ms
16 81.228.83.227 <TOS:0,DSCP:0,ECN:0> 150.816 ms
17 90.228.166.164 <TOS:0,DSCP:0,ECN:0> 153.555 ms
18 81.224.167.228 <TOS:0,DSCP:0,ECN:0> 153.135 ms
19 *
20 *
21 81.236.63.162 <syn,ack> 150.907 ms
    Timedout: false
    Duration: 1522.420 ms
    DestinationReached: true
```

IP-ECN bleaching in the wild

- We run Pietrasanta traceroute from Catchpoint nodes deployed around the world to understand how many traceroutes show the effects of ECN bleaching
- Period of reference: August 2024
- Besides research curiosity, this can be useful to understand how much the network is prepared to accommodate L4S
- This is not intended to be a rigorous research work
 - The results presented are obviously biased by the node selection
 - We tried to be as fair and distributed as possible in selecting sources and destinations

Results – IPv4

~403k traceroutes → ~48k (12%) bleedings

15%

60557 RUNS

2%

10016 RUNS

23%

145631 RUNS

6%

4013 RUNS

5%

180788 RUNS

31%

2746 RUNS

Results – IPv6

~253k traceroutes → ~15k (6%) bleedings

15%
88842 RUNS

0%
7180 RUNS

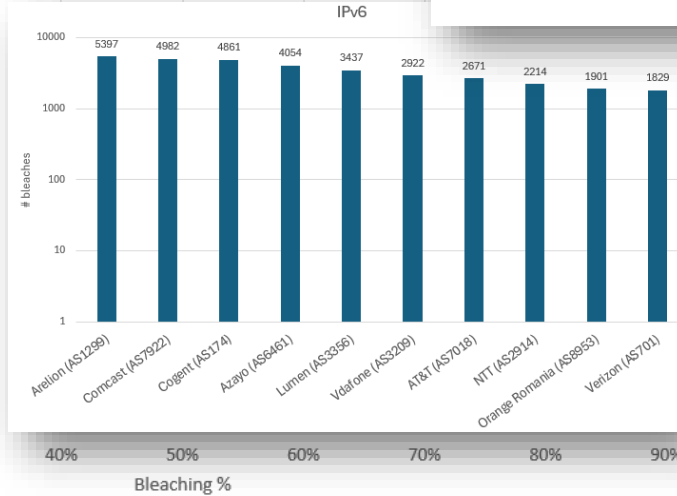
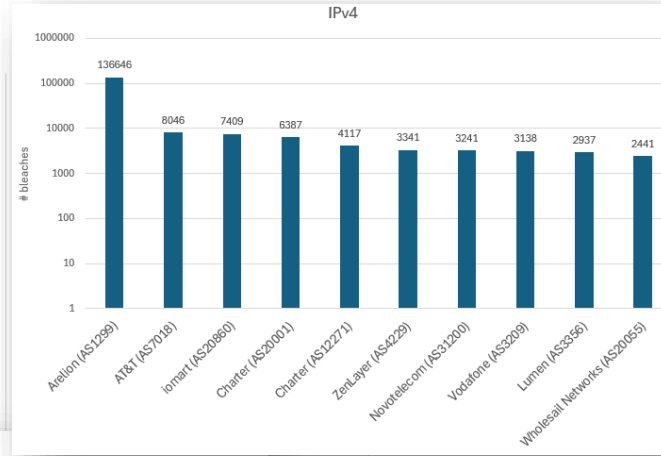
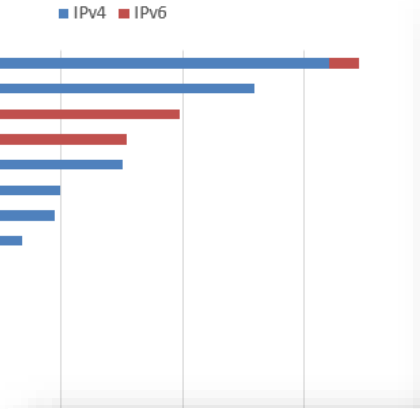
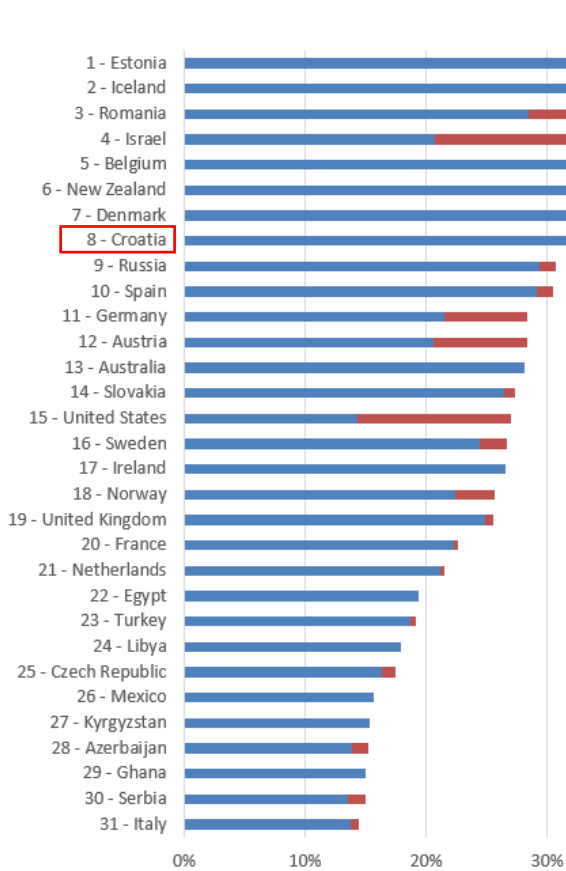
0%
1410 RUNS

3%
100139 RUNS

4%
56124 RUNS

-%
- RUNS

World



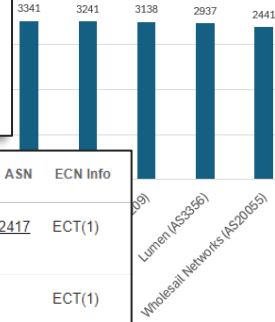
100%

Bleaching %

- 1 - Estonia
- 2 - Iceland
- 3 - Romania
- 4 - Israel
- 5 - Belgium
- 6 - New Zealand
- 7 - Denmark
- 8 - Croatia
- 9 - Ru
- 10 - S
- 11 - Germ
- 12 - Au
- 13 - Aust
- 14 - Slov
- 15 - United St
- 16 - Swe
- 17 - Ire
- 18 - No
- 19 - United King
- 20 - Fr
- 21 - Netherl
- 22 - E
- 23 - Tu
- 24 - L
- 25 - Czech Rep
- 26 - Me
- 27 - Kyrgyz
- 28 - Azerb
- 29 - G
- 30 - Se
- 31 -

- Source is in Zagreb under DHH network (AS12417)
- 364 bleached paths over 991 runs
- The vast majority of bleaching happened due to either Arelion, HE, Cogent or Vodafone

IPv4



IP Address	ASN	ECN Info
178.218.166.129 [gateway]	12417	ECT(1)
10.2.10.2 [10.2.10.2]		ECT(1)
213.186.16.184 [213.186.16.184]	44306	ECT(1)
185.151.132.77 [185.151.132.77]		ECT(1)
184.105.213.250 [100ge0-77.core2.bud1.he.net]	6939	Not-ECT
72.52.92.206 [100ge0-35.core2.bts1.he.net]	6939	Not-ECT
184.105.222.129	6939	Not-ECT
216.66.89.102 [serbian-open-exchange-doo.10gigabitheternet1-1-2.switch1.beg1.he.net]	6939	Not-ECT
10.91.25.92 [10.91.25.92]		Not-ECT
88.218.137.39 [gael-2.sox.rs]	60733	-

IP Address	ASN	ECN Info
178.218.166.129 [gateway]	12417	ECT(1)
10.2.10.2 [10.2.10.2]		ECT(1)
213.186.16.184 [213.186.16.184]	44306	ECT(1)
80.239.132.26 [zgb-b2-link.ip.twelve99.net]	1299	ECT(1)
62.115.122.174 [bpt-b6-link.ip.twelve99.net]	1299	ECT(1)
62.115.138.204 [win-bb2-link.ip.twelve99.net]	1299	ECT(1)
195.2.19.145 [ae40-xcr1.vie.cw.net]	1273	Not-ECT
195.2.19.145 [ae40-xcr1.vie.cw.net]	1273	Not-ECT

982 4861
50%
aching %

Conclusions and future work

- ECN Bleaching is not a tale and still around
- Pietrasanta traceroute may help in identifying where the bleaching is happening
 - You cannot fix what you cannot see!



Thank you!

- Feel free to check/use/ & contribute!
<https://github.com/catchpoint/Pietrasanta-traceroute>
(GPL!)
- And come by to meet us!
 - Pietrasanta is a nice town on Tuscany seaside ...

